AN INFORMED ENVIRONMENT FOR INHABITED CITY SIMULATION

THÈSE N° 2467 (2001)

PRÉSENTÉE AU DÉPARTEMENT D'INFORMATIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nathalie FARENC

DEA en informatique de l'Université Jean Monnet, St-Etienne, France
de nationalité française

acceptée sur proposition du jury:

Prof. D. Thalmann, directeur de thèse
Prof. T. Catarci, rapporteur
Dr S. Donikian, rapporteur
Prof. S. Spaccapietra, rapporteur

Lausanne, EPFL
2001

A mes parents, à Manu, à Isa

Avec tout mon amour

Tous mes remerciements vont au Professeur Thalmann pour m'avoir permis d'effectuer cette thèse au LIG.

Je tiens également à remercier tous les membres de mon jury le Professeur Catarci de l'Université di Roma, le Professeur Spaccapietra de l'EPFL , le Professeur Hersch de l'EPFL, et plus particulièrement Stéphane Donikian, chargé de recherche à l'IRISA/CNRS de Rennes, pour toutes les corrections qu'il a apporté à ce rapport.

Ce travail de thèse a été financé par The Swiss National Research Foundation et par l'Office Fédéral de l'Education et de la Science dans le cadre du projet CROSSES.

Je voudrais également remercier Ronan Boulic pour les discussions que nous avons eu ensemble concernant l'élaboration de ce travail, ainsi que les membres du LIG qui m'ont donné, parfois, un peu de soleil, ils se reconnaîtront c'est certain. Je tiens à remercier les personnes qui ont participé au projet de la ville virtuelle, sans elles les animations n'auraient jamais pu être aussi parlantes et complètes (Fabien, Soraia, Marcello, Elsa). Une pensée particulière aux designers Olivier Aune et Frédéric Sidler qui ont modélisé la ville, cette collaboration à été très agréable et très constructive pour ce travail, merci beaucoup. Je tiens également à envoyer un petit bonjour aux étudiants qui ont participé au développement de ce travail dans le cadre de leurs projets de semestre et diplôme, leurs implications étaient très souvent supérieures à mon attente, un grand merci.

Je voudrais remercier aussi les personnes qui se sont dévouées pour la relecture de ce manuscrit en apportant un peu, beaucoup .... de corrections: Daniel Thalmann, Lorna Herda, Ik Soo Lim, Ronan Boulic, Manu.

Ce travail n'aurait jamais pu être effectué sans le soutien sans faille de ma famille (dans le désordre les mamies, Colette, Christelle et Vincent, Dav et Isa, mes beaux parents, ...), une tendre pensée pour mes parents pour qui la réalisation et l'épanouissement de leurs enfants est la chose la plus importe, et bien ça marche, je vous aime. Tout mon amour et ma gratitude pour mon mari qui a tout laissé en France afin de me suivre dans cette grande aventure et qui a montré beaucoup de patience et de soutien jusqu'à la fin de cette thèse. Une caresse à mon poupon Corentin, un si gentil bébé même quand sa maman n'était plus très disponible. Un petit mot pour Isa, merci pour ton soutien, d'ici peu tu vas rédiger cette même partie, tu verras comme c'est chouette, de gros gros bisous et courage pour la fin. De même, j'envois un très grand merci à tous nos amis qui ont joué un très grand rôle ces dernières années, grâce à vous la vie est beaucoup plus belle, ils se reconnaîtront tous je le sais, bisous.

Résumé

Grâce aux récents progrès informatiques, la construction de grandes scènes virtuelles est devenue un secteur très en vogue en infographie pour les jeux, l'éducation et les films de synthèse. Fréquemment, des personnages de synthèse sont introduits dans des scènes représentant des villes reconstruites, des monuments détruits ou des villes virtuelles afin d'apporter une impression de réalisme. Le but spécifique de cette thèse est de permettre de la planification d'actions (l'action principale que nous avons abordée étant le déplacement) afin de peupler ces scènes avec des humanoïdes. Pour ce faire, nous utilisons une base de données dédiée à la simulation de vie urbaine qui est notre **Environnement Informé**.

Ce rapport présente une méthodologie de conception de scènes tridimensionnelles, et des outils associés afin de créer des animations dans des scènes à grande échelle. Nous proposons une décomposition de l'espace en un ensemble d'entités permettant de faire du calcul de chemin automatiquement, indépendamment de la dimension de la scène. Nous avons défini une décomposition de l'environnement 3D permettant la création d'une base de données, afin de simuler le comportement humain en zone urbaine. En premier lieu nous avons défini comment construire une base de données simulant des connaissances urbaines (reconnaissance de lieux et de leurs fonctionnalités) et des perceptions virtuelles (contact avec le sol, perception des objets, etc). Ces perceptions virtuelles, nécessaires pour la simulation d'autonomie, sont très coûteuses à simuler par une approche de reconnaissance. Elles sont simulées en accédant aux données contenues dans notre Environnement Informé. Nous avons décidé de définir/modéliser un modèle tridimensionnel idéal (notre ville virtuelle) pour la mise en place de nos concepts et outils, intégrant à la fois les données géométriques et sémantiques. Nous avons également défini des règles de conception et de stockage des objets 3D ainsi que des options permettant divers types de rendu de la scène. La connaissance urbaine, que l'on désire retrouver lors d'une simulation, est décrite par un modèle hiérarchique générique de décomposition de l'espace qui va fournir des informations sémantiques aux surfaces géométriques trouvées dans la scène. Ce modèle de décomposition de la scène est donc rattaché à un type de simulation et il est appliqué à la modélisation de la scène 3D. La base de données étant créée, des outils accèdent à son contenu pour calculer des planifications de chemins/actions, des informations concernant un endroit, des localisations d'objets ou de lieux ou encore des informations pour l'optimisation d'affichage.

Nous avons défini des outils pour la visualisation, vérification ou modification de la base de données. Ces outils ont été conçus afin d'offrir assistance à l'utilisateur lors de la manipulation de données tridimensionnelles car la visibilité et la vérification de données 3D avec des informations sémantiques attachées sont difficiles à observer.

Nous avons appliqué cette recherche sur des modèles de secteurs urbains existants dans le cadre du projet européen CROSSES pour permettre des simulations de panique en cas d'accident. Notre recherche s'est aussi orientée vers le domaine de l'aide à l'handicap afin de trouver des chemins adaptés à une mobilité réduite. Notre objectif a été de définir des outils et une méthodologie de modélisation 3D peu contraignante génerant automatiquement l'information nécessaire pour simuler un comportement humain autonome dans un environnement urbain.

Abstract

In combination with the rapid technical improvements of computers, building large virtual scenes has become a popular field in computer graphics for education, films or games. Often, virtual humans populate scenes such as real reconstructed city, disappeared building or virtual town, and they are expected to provide a real life feeling. The specific aim of this thesis is to perform action planning in order to populate these scenes with virtual humans. To achieve this goal, we use a database dedicated to urban life simulation called an **Informed Environment**.

This report presents a design methodology and associated tools for animation purposes in large scale scenes. We propose a space decomposition of the scene into a set of entities allowing automatic path computing regardless of scene size. In order to achieve human behavioural simulation, we decide to focus on the definition of scene decomposition resulting in the creation of a database.

The first stage of this work consisted of building a database simulating "urban knowledge" (knowledge about location and associated functionalities) and virtual perceptions (contact with the ground, object perception etc.). These virtual perceptions, necessary for autonomy simulation, but highly costly in time processor consuming, are simulated through database access corresponding to our Informed Environment. We decided to define an ideal 3D model as a test-bed for concepts and tools, integrating both geometrical and semantical data. We also define rules for the design, storage and display options. The "urban knowledge" attached to a simulation is the base of a generic hierarchical model of space decomposition. This model of space decomposition of the 3D scene is related to a specific type of simulation and directly applied to the modelling of the tridimensional scene. Once the database is created, some tools access the database content for path/action computation, knowledge concerning a place, display optimization, places and objects.

As 3D data visibility and verification with associated semantic information are difficult to observe, we have defined some tools for database visualisation, verification and modification (the "applicity" software). These tools have been developed in order to offer assistance to the user during manipulation of three-dimensional data.

This system is applied to real reconstructed scenes in the framework of the European project CROSSES, for panic simulation during accident. Our field of research has also been extended to the domain of disability help in order to compute path for people with reduced mobility. Our goal was to define some tools and a few constraint methodology that automatically generate all information needed for simulating autonomous behaviour in an urban environment.

I - Introduction

I.I - Urban human behaviour simulation

In combination with the rapid technical improvements of computers, building large virtual scenes has become a popular field in computer graphics. Often, within real reconstructed city, disappeared building or virtual town, virtual humans populating these scenes are expected to provide the real life feeling. Our specific aim is to populate these scenes with virtual humans in order to offer assistance in decision making concerning urban infrastructures. In this view, we have to integrate the problem of planning human actions and behaviour for urban life simulation into a virtual town.

In order to handle huge quantities of data attached to the virtual scene, we use for simulation a simple database. This database corresponds to our Informed Environment and contains different kinds of information. Thus, the behaviour of humans or other mobile entities must be coherent with respect to their location in the city. Populating a big-scale virtual scene such as a town is difficult, due to the need of large quantities of geometrical data originating from the model (altitude of the ground, location of objects, agents etc.) and including additional information that we shall call “urban knowledge". In order to yield more realistic simulations, the environment must integrate several semantic notions about specific areas such as “a side walk is a space dedicated to pedestrian motion". The urban knowledge has been defined in correlation with human perception and analysis in the context of urban life. Due to the cost of virtual perception (Bordeux and al., ref. [12]), the Informed Environment is a good way for substituting perceptions and analysis by information obtained through database access.

I.II - Animation of virtual humans in a virtual scene: constraints & complexity

At a first stage, an analysis allows the identification of major constraints for urban life simulation with virtual humans. These constraints concern the information transferred to autonomous agents as a way to link the scene creation to its use during simulation. Concretely, in order to handle agents and actions associated to a place, the managing agent tool needs all the geometrical information associated to this place. This information includes a good definition of the space surrounding the agent, the where abouts of the agent, the list of objects present in this area and the list of behaviours or actions associated with this place for a certain kind of mobile entity. Knowledge about objects is used for dealing with collision avoidance or for interacting with them.

According to the different aspects of the virtual humans animation presented in the next chapter presenting the state of the art, we can summarize our constraints and their inherent complexity.

A complex behaviour modelling in a large scene scale is based on:

*** a set of semantic notions attached with some places corresponding to the final simulation required.**

*** a structure for the data allowing a rapid access to the data.**

*** the simulation of visual and tactile perceptions allowing the knowledge of the spatial location of the ground under the foot of the virtual human and also visible objects for collision avoidance.**

*** a structure and the perceptions simulation for path and action planning.**

Each of these points bring to the light some notions with different level of complexity. The first point to refine is the scene. For realistic simulation of human behaviour, we use 3D representation both for the human body and for the realistic scene surrounding our virtual human. Thus, the displayed scene has a complexity defined by two parameters, the first one is the mesh (numbers of triangles and the precision of the textures) composing the scene and the second one is its surface. The realism of a scene is always depending on the number of triangles displayed and the quality of the textures mapped on. For a realistic large scale scene, we can imagine that the number of triangle can be so large that one of the first goal, even just for the display part, is to try to have at "useable" number of triangles. The notion of "large scale scene" can be simulated by a scene of 1km by 1km. The number of triangles and the size of the scene defining the complexity of the scene, from our point of view and queries, must be limited only by the power of the computer for its display.

As a direct consequence of the time spent by the scene/human display stage, a tool, responsible for knowledge about the scene and path/action planning, has to be efficient and not a time consuming process. It means that the data needed for achieving our goals have to be pre-processed (fast computation for answering queries) not time consuming for the data creation and use. A pre-processed structure can correspond to a constructed database in order to provide data and tools for the different queries. We decide to perform the maximum of computation in pre-processing in order to have a minimum of computation during simulations. The first stage for time computation optimisation is to add/compute the semantic level above the scene and to store this information. The tool responsible for the informed environment has been defined for an utilisation in two stages: the first one correspond to the creation of the semantic level above the scene in a pre-computed phase, the second one corresponds to the use of this semantic level for path/action planning as an informed environment during simulation in real time or in pre-processing phase.

A first approach is to use directly the set of triangles composing the scene. If the large scene has a 3D representation with a correct visual representation, the number of triangles is very big and the only path planning that the user can perform is to select a set of points for the creation of a path inside the scene. More automatic path planning can not be done due to the number of triangles and the impossibility of the creation of a semantic layer above them.

Another approach is to use a fixed grid and to sample the scene diminishing the number of entities carrying information and useable for automatic path planning. In order to animate a virtual human in displacement, keeping in mind that our large scene has $1km^2$ of surface, we can estimate that a correct precision for a scene decomposition above a fixed grid is 0.5*0.5 meter. This sampling could seem very thin but it corresponds to a human foot step, a door dimension and the space around the gravity centre of a human. Then for our large scene (with only two dimensions, no different altitudes such as inside buildings) we have a grid of 4.000.000 cells. These cells must carry information about their location in space (altitude), if their are useable or not by pedestrians (case of a hole in the ground for example), if they correspond to the location of an object (or a part of an object or a set of small objects), if they carry specific semantic (the user has to link this notion to all the cells by hand). The creation of such level of information above the grid can not be manually performed by the user for such quantity of cell and it is difficult to imagine an automatic process creating such information. Even if some cells are grouped and the number is decreased to 1.000.000, the problem of the association of triangles with the cells can not be done in real time and the problem of path planning with so many nodes in a graph can not be solved due to time consuming for computation (i.e. due to the very large number of solution).

In front of such constraints and associated complexity, the solution that we have adopted is to have a spatial decomposition of our scene into entities based on semantic notions. This decomposition is independent of the number of triangles composing the scene, and it must correspond to human cognitive model that a user can easily construct. These entities are grouped inside a structure that can allow path computation via graphs and semantic notions, definition of surfaces as places and simulation of object perception.

We assume, also, that we have to make some choices of problem simplification in order to keep a fast computation time. These simplification concern:

*** the geometrical creation of the entities not taking into account objects such as walls but only surfaces on the ground,**

*** a space decomposition optimising the number of entities,**

*** the definition of the entities with 4 points defining a convex surface, this modelling constraint limits to maximum one connection between two entities and optimise the path planning computation,**

*** to choose some adapted methods for path planning computation, keeping in mind the need to have a reduced number of nodes and edges inside the graph representing the environment, this constraint can bring some estimation for the shortest path computation for example,**

*** to try to have the less possible number of objects for collision avoidance computation and some simplified representation for these objects such as a bounding box,**

*** to have some parts of the scene not informed, the user is responsible for the data creation required for a specific simulation.**

All these choices inherent to the constraints presented in this section, are the result of the analysis of the resources (technical or human) needed for the simulation such as:

* the creation/analyse of the needs made by the user for a specific simulation,
* the design work of the entities and/or the scene,
* the time computation during both the pre-processing stage and the simulation stage (size of the database, size of the graph representing the database for the path planning computation),

taking into account the level of realism of the simulation performed required as a final result.

This Phd work consists in providing geometrical and semantic information for **the simulation, in an urban scene, of a virtual human who wants to go from home to the supermarket, eventually by bus**. In this framework, we have defined an example of a space decomposition model adapted to such constraints and this research model has been associated to the design step of a virtual city adapted to this simulation. The main goals of this work were to create a methodology useable by designers for a fast scene design, database creation and an easy/real-time use of the database for any kind of simulation. A friendly 3D interface, inside the "applicity" software, helps the user to create, visualise, verify, merge and test some database contents. Thus, via 3D manipulator, the user can use the tools manipulating data, such as the path planning tool, view the results computed and eventually store the result for pre-computed data. Others issues related to the database utilisation have been investigated such as display optimisation, infinite scene. The hierarchical generic model of space decomposition has been defined for the simulation an inhabited city. The model defined by the user is generic via a file text, and can be seen through a graphical 3D display in the applicity software.

I.III - Organisation of the work presentation

The layout of this document is composed by four chapters.

The chapter 2 presents different states of the art about various fields. The first field concerns the human body simulation: the human body 3D modelling and its animation comprising the animation and the behavioural aspect. The second overview is a rapid survey about spatial databases for animation and urban management. The third part presents the 3D modelling in general, where it is possible nowadays to find 3D modelling, its use, and more particularly the design of urban scenes. The fourth state of the art is about the path planning overview for robots and for animation, the last field concerns the display optimization.

The third chapter concerns the model of space decomposition and the database presentation with its internal structure.

Chapter 4 presents the various applications that can access this database for various purposes such as cognition of spatial location, path planning for virtual humans or other mobile entities, autonomy for virtual humans using the information coming from the database, scene composition and display optimization, tools for assistance in decision making for people with reduced mobility and a rapid presentation of the CROSSES project.

Chapter 5 is the conclusion and the presentation of future extensions.

II - State of the Art

This chapter presents various states of the art concerning different fields:

* The virtual human simulation with a brief description of the body representation, the simulation of the five senses, the behavioural simulation and the notion of autonomy. Although this work doesn't concern directly the body representation of the human body and its deformations, we briefly presents an overview because the main purpose of this work is to create realistic simulation of inhabited city simulation. This requirement implies both a correct city modelling and realistic human representation. As we want to simulate human behaviour in a city we cover the various aspects of the physically and psychically human modelling. We present also the various techniques of animation with objects that our laboratory used for simulations mixing this work and other works for more complete animation (Farenc and al., ref. [30]).

* The spatial databases in the domain of senses/knowledge simulation and in the larger field of urban management using most of the time some GIS tools (Geographic Information System) for storing,linking and accessing 2D/3D data.

* The 3D modelling with the scientific modelling, the entertainment domain, the therapy with virtual reality and the animation film sector. Finally we make an overview about the historical and architectural modelling, the extracted notions helping for the definitions of methodologies for 3D modelling in large scale scene such as a town.

* The path planning overview in robotic and animation field that can be applied to the path planning for virtual human inside an informed environment corresponding to a virtual city. The path planning is necessary to simulate autonomous behaviour when a virtual human decides to go to the train station whereas it is located inside a park.

* The display optimization field, a possible use of the database, necessary for large scale scene with enough details making simulation realistic.

All these domains and techniques described in this state of the art had an influence on our choice in the development of this work concerning the inhabited city simulation.

II.I - Virtual human simulation

To perform realistic simulation we need to have realistic human body representation and good motors or techniques for their animation. The level above concerns actions combined creating behaviour and at a higher level, behaviour corresponding to the notion of autonomy. All these aspects are presented in the next paragraphs in order to provide an overview about what we use for the simulation of inhabited towns.

After a coherent representation of actions, interaction and body modelling (Appendix A), the feeling of being alive is attached to our input coming from our five senses: vision, touch, hear, smell and taste. Simulation of these senses enhances the feeling of being here for the user and improves the model of behaviour simulation for virtual humans.

II.I.I - Simulation of Perception: the five Senses

In order to create simulations of inhabited towns, we can order the population into three categories: the user with his/her feeling to being here, some avatars corresponding to the representation of other real persons "sharing the moment and the place" and the "autonomous" agents evolving in the scene. If we analyse our behaviour, it is composed of two principal inputs: the data coming from our perception system and the behaviour based both on our knowledge and our personality (how to perceive surrounding events, how to react and internal feelings not fully dependent on perception) and one output corresponding to the faculty to perform actions (simple action like moving an arm or more complex ones such as grasping an object) and to combine actions. The feeling of a good immersion for a user evolving in a virtual world, and more, in a virtual inhabited town, is provided by a good simulation of his/her five senses. We can try, briefly, to provide a rapid overview of the state of the art nowadays via virtual reality peripherals and new haptics. Visual perception is the most developed, based on modelling and display performed for the sensation of being inside the virtual scene, some high technology peripherals are tested and we can mention the head mounted display providing the user with notion of depth inside the scene. This technology is continuously evolving and constructors expect in a near future to be able to provide public users with this technology. The touching perception can also be simulated via force feedback tools integrated inside joysticks or perhaps in the near future via your mouse or sensors applied to the body (M. Hodges, ref. [39]). The sense of hearing is difficult not due to the access to the sensors providing the input to the users (sound surrounding, stereo headphones are common nowadays) but due to the physical simulation of wave propagation. The model is very complex and depends on a lot of factors, such as localisation of sound sources, mix of the waves, material and objects present in the virtual scene (example of work in this field T. Funkhouser and al.,ref. [36]). The last sense where simulations are tested concerns the sense of smell. A new device is arriving on the public market and it creates basic pheronomes via a room scenter (digiScents firm product ref. [110]). Still in its early stage, this type of simulation has incredible side effects due to the gap in our knowledge of the mental analysis performed with pheromone interactions and of its impact on our feelings (making one's feelings good or uncomfortable, in love with another person or in conflict, for example). The last feeling, taste, has, at this moment, no peripherals available for its simulation.

After this rapid overview of the peripheral simulators of sense for real user, we can try to imagine how can we simulate this perception for virtual humans. Modelling of perception can be handled at different levels. At a higher level, the reproduction of the neural connections is a field of research in Artificial Intelligence. Based on medical observations, the models try to understand and imitate the visual and olfactory system present in real animals and humans. D. Osorio and al. work on neural architecture for different insects' sensory signals compared to other categories of animals in order to propose robots simulating their perception and behaviour (ref. [72]). The simulation of assimilation and parsing of olfactory/visual receptors excitation is really complex and doesn't rely on a simple model, the notion of learning and correlations between data make those systems highly complex and too complex for simulations of humans in a city. The work presented in (ref. [72]) gives an overview of this field.

For the five perceptions, their simulation is done most of the time using variations of internal status. The user or the surrounding simulated environment creates an event simulating a perception. This event can have different kinds of effects, randomly modifying or not some internal status creating new behaviour. More detailed research concerns synthetic vision (O. Renault and al., ref. [85]) combined with visual memory analysis, or database access and filters (C. Bordeux and al., ref. [12]), or memory and learning (J. J. Kuffner and al., ref. [45]).

In all these cases, we can note a difference between perception and intellectual simulation. The input coming from senses does not have an immediate link with a direct consequence on the individual behaviour, in the simulation of virtual human we treat in such case, some variations on the virtual human's internal status via numerical variable. In the Artificial Intelligence field, the behaviour depends on the management of these internal statuses, sum of all inputs provided for the simulation. The next section is a rapid overview of behavioural simulations and some definitions from the psychological and robotic domain of the notion of spatial knowledge or spatial representation.

II.I.II - Spatial representation and behavioural simulation

II.I.II.I Spatial representation

We have studied some examples of cognitive modelling of internal spatial representation. John Funge and al. present a cognitive model corresponding to the knowledge of a character, how this knowledge has been acquired and how it can be used to plan actions, all these parameters forming a complex behaviour. A pyramid representing the Computer Graphic modelling puts at the top the cognitive modelling, above the notion of behaviour (J. Funge and al., Figure II.I.1 and ref. [34]).

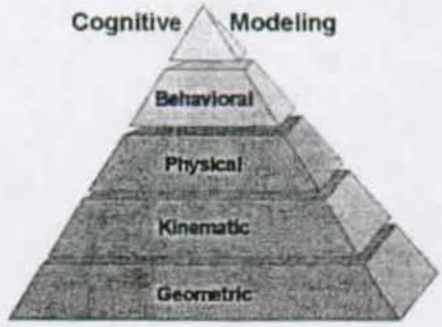


Figure II.I.1 *Cognitive modelling is the new apex of the CG modelling hierarchy(J. Funge and al., ref. [34])*

A specific language CML, is dedicated to assist game developers and animators to afford a character, providing knowledge about its world in term of actions, their preconditions and their effects. For our purpose, it is interesting because it presents the geometrical data set as a basis for this pyramidal knowledge modelling. This geometrical layer can be associated with a spatial semantic hierarchy defined by B. Kuipers that defines spatial knowledge as a "foundational to commonsense knowledge, and hence to most kinds of knowledge that humans possess" (B. Kuipers, ref. [46]). Defining the notion of **large-scale space** as space whose structure is at a much larger scale than the sensory horizon of the agent, he proposes the notion of cognitive map (human knowledge of large-scale space), visual space (immediate surrounding environment) and graphical space (spatial layout and relations among symbols on paper or other displays) as different representations of a large-scale space. B. Kuipers specifies that knowledge of large-scale space consists of several distinct but interacting representations, each with its own ontology, collectively know as the Spatial Semantic Hierarchy (SSH).

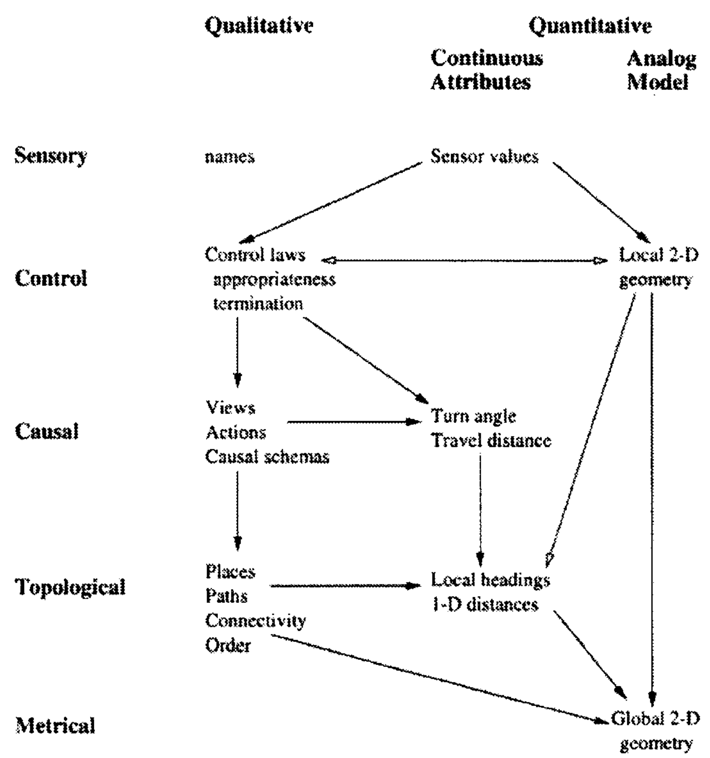


Figure II.I.2 *The distinct representations of the SSH. Closed-headed arrows represent dependencies, open-head arrows represent potential information flow without dependency (B. Kuipers, ref. [46]).*

Figure II.I.2 shows the different lattices according to distinct representations horizontally and different ontology notions vertically. The sensory level is the interface to the agent's sensory system (during motion and exploration by vision e.g.). The control level describes the world in terms of continuous control laws associated with a qualitatively uniform segment of environment. Local geometric maps are considered such as local observers for the law control. The causal level abstracts the continuous world, and the agent's behaviour within it, to a discrete model described in term of sensory views, actions, and the causal relations among them. The topological level introduces the ontology of places, paths and regions, and their connectivity, order and containment relations: features of an external environment. An interesting remark in this work concerns the topological network map that is augmented with a hierarchical region structure. The author emphasises that this structure is more effective for planning than the flat causal action model. The last lattice is the metrical level and represents a global geometric map of the environment in a single frame of reference. This work has been applied to robot evolution inside an unknown large-scale space, the robot applied and linked all the level data inside a global framework in order to discover all of its environment.

In this presented work, the information is hierarchically structured from geometry data to control laws, in order to tidy up, access and connect information. Another way of representing spatial perception and learning has been developed by T. J. Prescott. In (ref. [78]), he declares "In contrast with Kuipers' hierarchical approach in which global topological and metric models are constructed, it proposes a "heterarchy" of local models in which the geometric distinction is only one among a number of characteristics identifying complimentary representational forms, Michael Arbib (ref. [5]), calls this a 'multiple schemata views'". Based on approximate and partial space representation (due to some sample of data coming from may be inaccurate and unreliable sensors input) this schemata modelling allows the merging of different source and richness (topological-metric) of geometric data with semantic and behaviour directly associable. This method can be linked with various work on animal internal space representation and is directly linked with the own agent's discovery of its surrounding environment.

Another way to define a spatial cognitive model for a large-scale scene has been defined by Mitchell (ref. [59]): "Long ago, the urban theorist Kevin Lynch pointed out the fundamental relationship between human cognition and urban form - the importance of the learned mental maps that knowledgeable locals carry about inside their skulls. These mental maps, together with the landmarks and edges that provide orientation within the urban fabric, are what make a city seem familiar and comprehensible -".

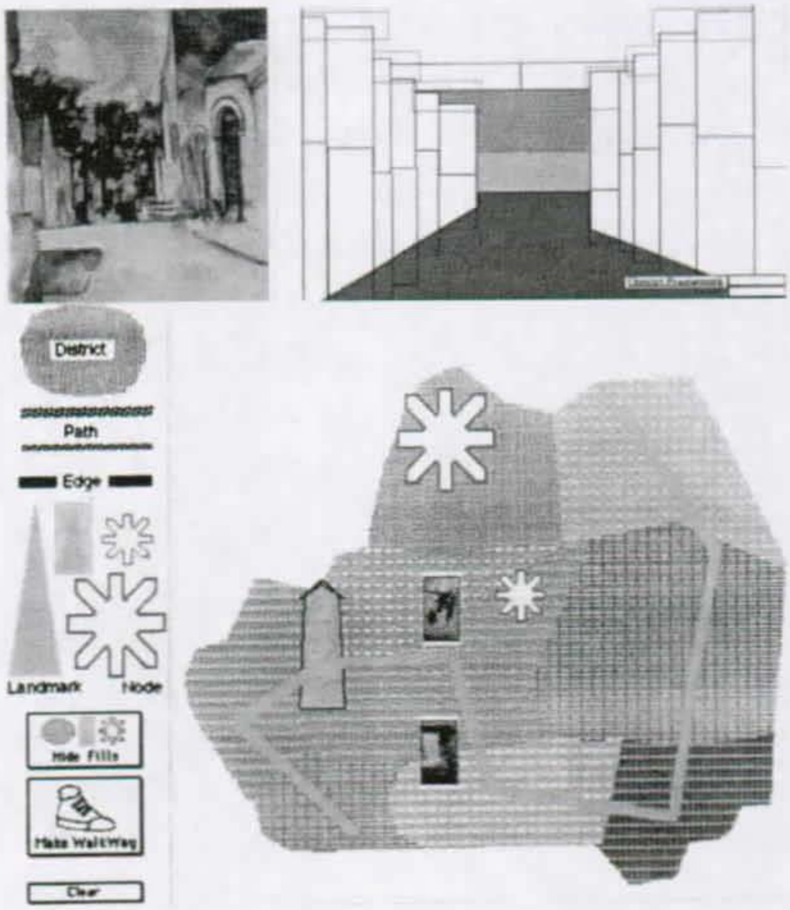


Figure II.I.3 *Cezanne's Ferme a Montgerould painting (1898), with its representation as a set of variable coordinates to be filled with an appropriate excerpt from the painting. Environment composed from different paintings analysed such as the above one, with landmarks and path for a user walk-through (D. Sorid and al. , ref. [101]).*

This definition was the basis of a city land composition from static images in order to navigate in 2D scenes decomposed and reconstructed in a speudo-three-dimensional model composed by some parts of mapped planes distributed in space (C. Strohecker and al., ref. [102] and Figure II.I.3).

The notion of behaviour and autonomy can be added to these examples of representation in order to create autonomous agents in an urban context.

II.I.II.II Behaviour definition

We can denote three components of behaviour in response to various environment structures, namely action, sensation and memory. According to P.M. Todd and al. (ref. [105]), action is necessary for behaviour and indeed adaptive behaviour. That is, creatures need to be able to sense and remember anything about their world (or themselves) in order to adaptively behave in it, provided the world is generous and benign enough.

As we want to simulate urban behaviour this means that, our virtual human must be at least able to react to perception of collision with objects (simulation of visual and possibly touching perception) with an upper layer corresponding to a knowledge of the environment (places and objects) with behaviour adapted to the place (common action associated with a place or more complex behaviour such as a respectful attitude inside a cemetery and playtime in a park. Other simulations can be handled as events perceived by the virtual human and this could be input for its own behaviour.

With a correct simulation of sense perception handled by a coherent and adapted behaviour, we can imagine the next step as corresponding to the notion of autonomy for an agent.

II.I.III - Autonomy Simulation for an agent

Definition coming from the Autonomous agent Conference'98 introduction:

"Autonomous agents are computer systems that are capable of independent action in dynamic, unpredictable environments." The fields where applications use autonomous agents are several and can be ordered in different categories such as:

* Representation for a personal artefact during absence (telephone, portfolio, web application inside a personal web page, document avatar for a survey or a more detailed explanation inside a document...)

* Interconnection between a user and autonomous agents such as Gandalf (K. R. Thorisson, ref. [106]) for advanced interfaces,

* Decision making for shortest path planning for robots, or space exploration by robots.

* Decision-making research about the most adapted reaction according to the user action such as an expert surgical assistant (M. Billinghurst and al., ref. [11]).

* Another field is concurrence or communication/cooperation for multi-agent simulation. We can also find "autonomous agent" for browsing and searching documents on the network according to the user internal schema.

* Simulation of theatre characters using avatars or autonomous agents responding to high-level scenarios or user commands and internal personal traits and social-psychological models.

The aim of our work is to simulate the most used perception: the visual perception. As this perception is costly in term of image analysis, we can substitute this processing by an access to a database providing during the simulation all that can be seen by an agent. Furthermore, as shown in the previous sections, autonomous behaviour are dependent on notions of semantic and behaviour attached to places. Then we associate to the geometrical definition of the database new layers to enhance the possibility of spatial location knowledge and behaviour adapted to the places. The next section concerns a rapid overview of databases providing information about a scene, allowing the simulation of knowledge and urban management.

II.II - Spatial DataBase

II.II.I - Database for senses/knowledge simulations

The domains of spatial representation and databases for sense simulation overlap very frequently. The access to a database providing information simulating perceptions and knowledge can be a good solution for a real-time urban life simulation. Various work in this field already exists and C. Claramunt (ref. [19]) proposes an overview concerning this domain. Defining the notion of Navigation knowledge principles based on a proposal of a spatial view, he presents a way to use a database for path planning in space using the notions of graph and network. This model allows a displacement action to be situated within its geographical context through complementary abstraction levels that accept partial knowledge. The spatial view gives a representation framework for navigation knowledge. It associates the visualisation of a route with multidimensional spaces that allow it to be situated, including significant visual landmarks and textual descriptions.

Some annotations can be added to a non-graphical model in order to simulate complex behaviour. The notion of "annotated world" is an example. P. Doyle and B. Hayes-Roth (ref. [25] & ref. [26]) defined the notion of annotated world dedicated to human-like agents intelligently performing actions relevant to the domain or the place where they are. The annotations correspond to the information creating intelligent agents dedicated to answer queries, to provide guidance, or to participate in the activities of human users. These annotations are textual information attached to places, and have been tested on multi-user worlds within child educational environments. The space is annotated so that it can be inserted into a variety of characters with different personalities and goals that will populate the world as friends, guides, competitors and passer-by encountered by the children. The agents initially know nothing about the content of the world, then are only aware of its basic dynamics; they rely upon the annotations for the rest.

II.II.II - Database for urban management

A lot of other models exist within Geographic Information System (GIS). The domain of application covers a very large panel such as urban infrastructures, ground exploitation etc. and one actual problem is to link data and spatial representations accumulated by governments agencies on different subjects inside a common framework (M. E. Orlowska and al.,ref. [71]).We have selected three example for database applications directly related to virtual reality and three-dimensional modelling of virtual cities. Another way to construct a 3-D model with an associated database has been developed by A. Gruen and al. (ref. [37]). The basis of this work is to analyse a set of points coming from photogrammetry and to automatically generate a topology via a CyberCity Modeler linked to a spatial information system (CC-SIS, that is specially designed for the handling of a 3-D city data) integrated with a hybrid GIS (raster images and vector data). This situation led them to develop an appropriate data model, that should not only represent the geometrical information, but also implicitly or explicitly describe the topological relationship between various data. New public applications of GIS had made their appearance in cars, for dynamic route navigation. Y. Banjou and al. (ref. [10]) present an integration of the Global Positioning System (GPS), with a GIS, and a graphical interface, using spatial databases through landscape images. Built for informing and accessing information adapted to disaster events, this work merges GPS' data, digital camera input, three-dimensional geographic objects, annotations in images and digital elevation model.

Some building creation applications were designed for city planning purposes (T. Fuji and al., ref. [33]) or to handle fire safety procedure. In order to be more efficient, some basic knowledge is based on the data accessible by authorities during an exercise. The data can belong to various fields and various people, R. W. Bubkowski (ref. [15]) mentions the factors implied in fire safety requirements such as:

* Knowledge of the plans of the building with technical data (gas, water pipes, electricity cables...)

* Localisation of the doors and exits influencing the fire progress knowledge of normal occupancy of the place (number of persons)

* Knowledge of the material used for building construction (toxicity and time consummation influencing fire propagation)

* Simulation of fire development and smoke movement...

This mass of information can help for path/evacuation/fire-fight finding solutions for an easy access to data and a good visualisation of all the parameters. Three-dimensional modelling seems to be a good way to group all this and simplify fire safety. The authors propose an international standard for performance codes, such modelling could be a part of such process. The same author has created a training fire simulator using a 3D representation of a part of a building real-time visualisation (ref. [16]). It appears that "the use of 3D computer graphics techniques employing suitable symbolic visualisation permits scientists to perceive several variable values such as temperature, smoke levels, or air toxicity in parallel and quantitative manner. On the other hand, using more natural rendering techniques showing flickering flames and drifting smoke clouds, gives even lay-persons an intuitive understanding of the environmental phenomena being simulated. Such realistic-looking virtual worlds offer the promise of practising fire-fighting and rescue strategies without any physical danger to the trainee."

Virtual construction with associated databases can be at a town level and we have found three examples of such modelling: Los Angeles (presented previously), Lisbon for the World Exposition commemorative tower, the Philadelphia 2000 project.

The Los Angeles project (ref. [42] & ref. [43])., an early 90's application, integrated visual simulation and GIS (Geographic Information System). Ligget and al. (ref. [49]) have created a model of part of Los Angeles using a Computer Aided

Design tool (CAD) and an associated Geographic Information System within mind to aid in the development of the Pico Union neighbourhood in Los Angeles.

Figure II.II.1 *Street view of Pico Union neighborhood model (Liggett and al. ref [49])*

This quarter is considered to be the most in difficulties community of LA. Via this set of tools the goal was to visualise both the reconstructed scene and a set of data stored inside a GIS system linked to the 3D selected object. The model was created using a certain number of rules in order to be able to add links to the database and optimise the display (Figure II.II.1). The goal was to combine aerial photographs with street level video to efficiently create a realistic computer model of a distressed urban neighbourhood that can then be used for interactive flying and walk-through demonstration. The purpose of this model is to visualise potential modifications to the urban fabric and experience these changes in their actual context, facilitating the work of city planners and designers. Furthermore, this paper proposes a methodology to link data coming from a GIS database (all requests and analysis possibilities linked to these kinds of tool) with a visual representation, and also a way to segment the reconstructed scene in order to have correct visualisation during walk-through.

Figure II.II.2 *For the past few years, a CAD/GIS team in Portugal has been simultaneously planning two new "cities" for the same 800-acre site, a world's fair exhibition area and a modern "city within a city". LisbonViews of the World Expo construction project with theVasco da Gama tower and the expo hall, a permanent structure placed in the master overall scheme with temporary and future ones (K. Moltenbrey, ref. [62]).*

The World Expo held in Lisbon (Portugal), leaved a great legacy to the city: a new town-quarter that will be completed in 2010, called "Parques das Nacoes" and will boast about 12.5 million square feet of residential development that will provide living space to about 25,000 residents; 5 million square feet of office space where about 18,000 employees will work; 1.5 million square feet of retail shopping; about 250 acres of parks and open spaces; new schools and a hospital. This planning is constrained not only on a short-time (almost 15 years) but also on the World Expo constructions taking care of general infrastructure, temporally (with notions of recyclability) and permanent infrastructures. For that purpose, the planners requested that any architectural plan submitted to the World Expo, be accompanied by a digital file, preferably in AutoCAD format, since the model base was built using that software. Surprisingly, 90% of the participating countries provided plans in digital format, that were incorporated into the 3D AutoCAD model of the entire Expo site and into the 3D virtual model. Using various representations of a same building, the models integrated more and more details both for the urban site and the exhibition site. According to the responsible team for the project, the first deadline of May 1998 was a success (regarding the criteria of schedule, money and result). The second stage of the project has its deadline in 2010, it has started and seems following the same path. The evolution is the accessibility of all AutoCAD and 3D Studio Max (ref. [122]) design information on the group's intranet sites, enabling architectural firms, for example, to obtain all the relevant information about infrastructure (K. Moltenbrey, ref. [62]).

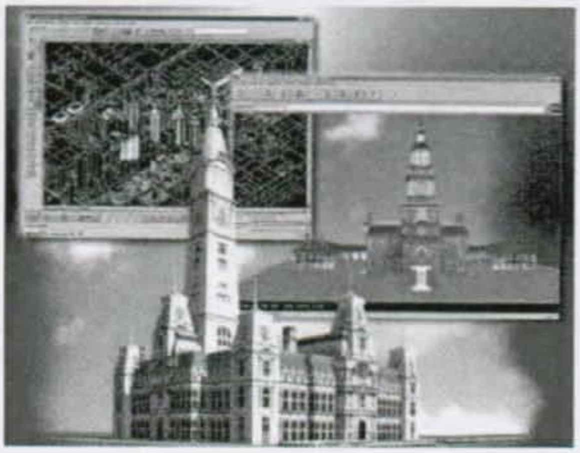


Figure II.II.3 *Early pictures of the work on the Philadelphia 2000 project, the models were created in MicroStation and rendered in MasterPiece- both products of Bentley system (ref. [52]).*

Figure II.II.4 *Picture from the Philadelphia 2000 project (D. P. Mahoney, ref. [52]).*

The Philadelphia 2000 project (D. P. Mahoney, Figure II.II.3 & Figure II.II.4 & ref. [52]) is a 3D model of the town handled by a consortium of organisations and companies and led by the American Institute of Architects (AIA). The model is composed not only of the physical structures that occupy the streets, but also of the underlying infrastructures such as utilities, transportation, phone and cable lines. Extended to an Internet version accessible for tourism, the model is called to be extended, creating a business around it, where people can design their own buildings and integrate them into the model. The interest is very high because of the marketing benefits. Such modelling is a real challenge and architects have to deal with very large databases, different levels of details for the same building, but also the need to be able to "travel" from one point to another one such as from the city centre to the airport, assuming visualisation during the entire simulation of the transportation.

In many various examples seen previously, we have observed applications for terrain exploitation such as city building/industry construction, or decision-making for organisations using 3D modelling. The modelling of virtual towns is dependent on the user's desired usage but can try to benefit from previous studies done with respect to the impact of the urban design and social phenomenon (R. Ingram and al., ref. [42]). The last interesting field we want to mention concerns the study of traffic flow management, with 2D or 3D representations of the circulation direction and the junctions management for vehicles (K. R. Howard, ref. [41] and P. L Mokhtarian ref. [66]).

Modelling of virtual towns or buildings is also done with other methods, using 2D maps with aerial photographs and image analysis (T. A. Russ and al., ref. [94]). The reconstruction can provide access to a database concerning a road network. The French National Institut of Geography (IGN ref. [116]), for its topographical database, uses during a first step aerial photography analysis in order to extract segments of roads and simple junctions, and in a second step, human interpretation and graph corrections complete the network.

Figure II.II.5 *"Tokyo down-town" with partially mapped textures and Zürich, ETH campus Hoenggerberg with mapped textures and planned building (upper right) (A. Gruen and al., ref. [37])*

There are other modellers for 3-D city construction, the CC-Modeller is a nice example (A. Gruen and al., ref. [37]). This tool reconstructs buildings or others objects that can be approximated by polyhedron surfaces (roads, rivers, parking lots, ships etc.) using a Digital Terrain Model (DTM) with a Triangular Irregular Network (TIN) structure more adapted to the complexity of the terrain structure in most city areas than a regular grid model. Moreover the system manages vector data and raster images based on the relational database technology. Above the CC-Modeller, Armin Gruen and Xinhua Wang implement a CC-SIS (CiberCity Spatial Information System) able to generate information and answer most of the questions about data storage, manipulation and geometrical/topological queries or natural/artificial texture mapping onto the 3D model objects (A. Gruen and al., Figure II.II.5, ref. [37]).

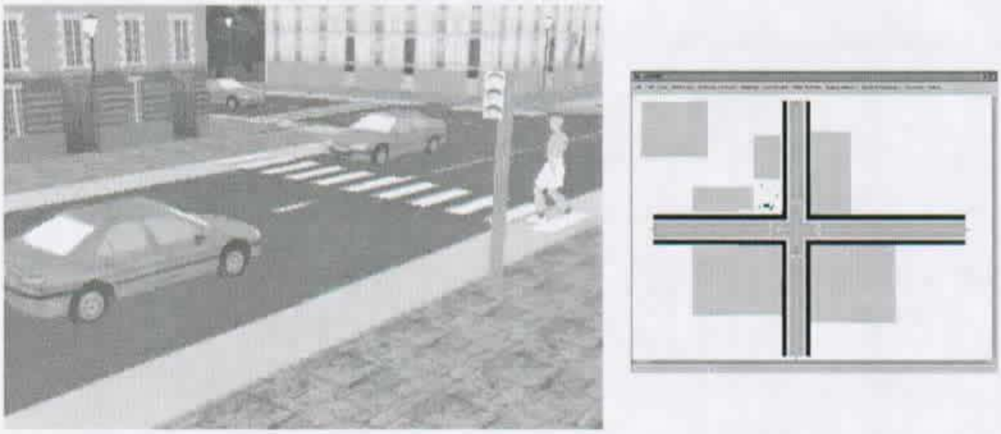


Figure II.II.6 *View of car simulation in the Rennes town form the VUEMS project (on the left) and view of the VUEMS interface for the scene creation (S. Donikian, ref. [24]).*

An interesting project, called Praxitele, arose at the end of the 90's and concerned the design of a novel transportation system based on a fleet of small electrical public cars, under the supervision of a central computer. These public cars are driven by their users, but their operation can be automated at specific instances. Before making tests in real life structure, all the partners (car constructors, persons responsible for transportation technology, automation e.g.) needed a common simulation platform. This platform is in the city of Rennes in France and S. Donikian (ref. [24], Figure II.II.6) defines a modelling system (VUEMS following the Praxitele project) building both a 3D model and an associated multi level database. VUEMS (Virtual Urban Environment Modelling System) is dedicated to the construction of virtual cities for the animation of vehicles' behaviours. For this purpose, the centre of the city of Rennes (France) has been reconstructed using various kinds of information supplied by the Technical Services of the city of Rennes, such as:

* a cartographic database containing the description of linear objects such as networks for gas, electricity, parcels,

* scanned maps of road-ways,

* traffic lights descriptions (location, organisation and synchronisation),

* pictures of the streets for building mapping.

In output, the 3D scene is reconstructed with the buildings' geometry, with mapped texture and a topological graph with a last level of information concerning some semantic data (direction and type of circulation ways etc.). The main objective of this work was to describe the behaviour of dynamical entities evolving in virtual environments and particularly the cars. The decomposition of the scene was based on streets and junctions for car simulation. The next step of decomposition corresponding to a modelling integrating data for the human behaviour simulation outdoor has been done by G. Thomas (ref.[104] and ref.[103]) and the decomposition model relies on our model described in the next chapter and in (ref. [31]). Integrating a modelling system, a set of various databases including a GIS system, a symbolical information structure dedicated to animation and some automata for behavioural simulation, this work seems to be the only one corresponding to the simulation and animation of inhabited city for the state of the art.

At a higher scale of modelling, the TerraVision II project (M. Reddy and al., ref. [83] & ref. [84]), aims at representing the Earth, with various levels of detail from a global view of the globe to a building such as the Taj Mahal in Agra (India). All geographical data is specified in latitude/longitude coordinates and integrated into a three-dimensional computer graphics scene graph, in GeoVRML1 format accessible on the Web. GeoVRML1 is an extension of the VRML97 syntax to provide support for large-scale geographical applications.

II.II.III Conclusion

With all these examples of database utilisation, spatial knowledge definitions and simulations of perception, we can try to define the need for urban virtual human behaviour simulation inside a graphical context needing a fast access to information concerning the simulation of perceptions. The next section presents the 3D modelling domain.

II.III - 3D Modelling

II.III.I Introduction

With the progress of computers over the past decade, more and more three-dimensional modelling is appearing, in disparate domains. Allowing the visualisation of the infinitesimally small such as molecule modelling, infinitely great such as a galaxy or more common-scale objects, the 3D modelling is entering gradually in our current life. A three-dimensional model lets the user manipulate it, offering various points of view, modification and visualisation as it can never be achieved in a classical drawing. We have tried to succinctly enumerate the various domains where we can find this technological explosion with a few examples often corresponding to advertisings found in the press or brief presentations of existing applications using commercial software. Arbitrarily, according to the use of the models, we define four groups: architecture, scientific modelling, entertainment and industrial use. Some modelling examples belong to all four groups. After this brief description, the three themes concerning archeology reconstruction, terrain exploitation and management of world database modelling will be explored in more detail.

II.III.II Scientific modelling

Scientific modelling can represent very large objects such as galaxies or planets (D. P. Mahoney, ref. [55], Figure II.III.1), or very small ones such as molecules, for example.

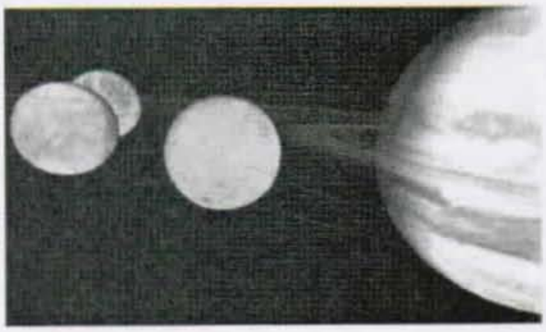


Figure II.III.1 - *Jupiter's volcanic moon Io moves on a circular orbit in an interactive digital solar system created by Superscape for National Geographic 's web site D. P. Mahoney, ref. [55])*

Scientific journals very often present three-dimensional modelling in various fields. We can note for example 3D modelling in astronomic research concerning modelling of solar system asteroids and the computation of their behaviour during impact with other asteroids (E. Asphaug, Figure II.III.2 and ref. [7]) or in the bio-informatics fields relying on DNA molecules or other complex molecules regarding how unit elements compose genes (C. Ezzel and al., Figure II.III.3 in ref. [28]).

Figure II.III.2 - *Collision scenario between a smaller asteroid at high speed and a rubble-pile asteroid (E. Asphaug, ref. [7]).*

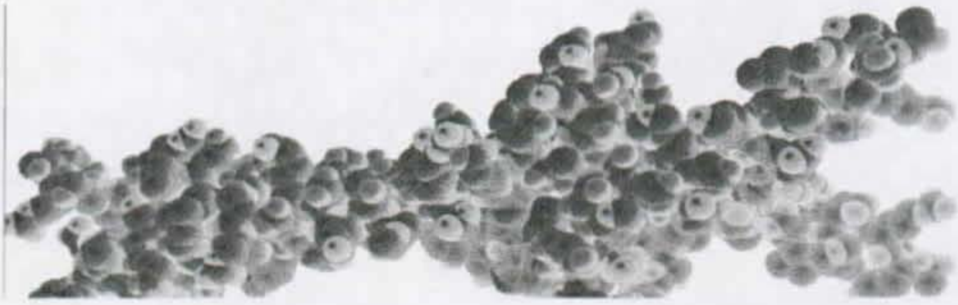


Figure II.III.3 - *Three-dimensional modelling of a complex molecule (C. Ezzel and al., ref. [28]).*

Other data not visible but corresponding to time evolution can be observed with 3D modelling. It can concern invisible phenomena such as force fields, magnetic fields, or the evolution of electricity in the body such as heart fibrillation (C. Ezzel and al., Figure II.III.4, ref. [28]).

Figure II.III.4 - *Computer model of the heart in the fibrillation state shows waves of uncoordinated electrical activity sweeping over the organ (C. Ezzel and al., ref. [28]).*

Another example is brain electrical movements. Figure II.III.5 (D. P. Mahoney, ref. [56]) contains ellipsoids representing the anatomical variation between brain regions of 20 normal subjects, based on MRI data. Such visualisation helps determine whether a brain region is abnormally situated, or whether the variation is normal. This helps in the diagnosis of certain neurological conditions and the evalua- tion of treatment approaches.

Figure II.III.5 *Brain modelling with ellipsoids representing the anatomical variation between the brain regions of 20 "normal" subjects, based on MRI data (D. P. Maloney, ref. [56]).*

In geography, some research teams work on sea water evolution or the state of the ground in a desert region. In such application, GIS (geographic information sys- tem) are mostly directly linked to the modelling in order to provide or the data and/or its evolution. 3D modelling allows the visualisation of numerical data that is difficult to analyse without a good visual representation. 3D can sometimes provide a good solution for problem visualisation.A. Gruen, ref. [37] presents the evolution of tradi- tional Geographic information systems towards mapping from data stored in the clas- sical database into 3D imagery. These new 3D GIS tools allow users to generate fly- through high-resolution scenes for presentations and collaborative work, offering views from virtually any location. One reason why urban GIS are so often used by the collectivities and private citizen is its ability to highlight cause-effect relationships that are slow and hard to recognise. Figure II.III.6 is an example of a project queried by American Forest concerning the analyse of satellite images to illustrate the decrease in tree coverage resulting from urban sprawl in several US metropolitan areas (M. Hodges, ref. [38]).The images to the right show the gradual decrease of the tree coverage in the Seattle area in 1972, 1986 and 1996. One consequence is that cit- ies and countries had to build new storm-water management facilities to replace the water storage previously provided by trees.

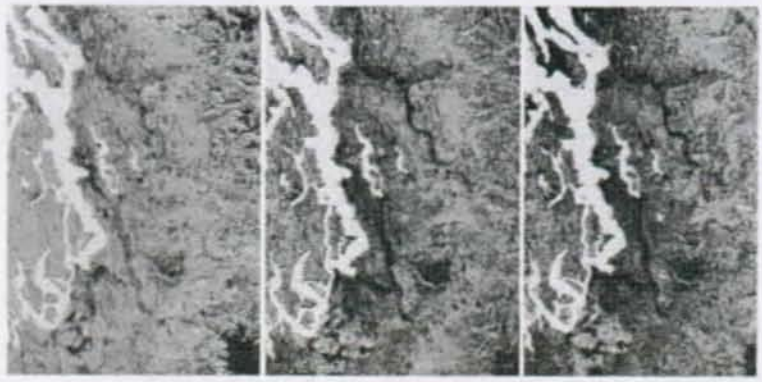


Figure II.III.6 *Using CITYgreen, an application of ESRI's ArcView 3D Analyst, American Forest can quantify the impact of diminishing tree coverage in local municipalities other 1972, 1986 and 1996 (M. Hodges, ref. [38]).*

Another type of data belongs to disappeared or damaged historical structures, here we enter the domain of architectural design tools dedicated to archeology. Another field concerns archaeology objects but not architectural ones, and there are several uses in this domain. We can note that a damaged object such as the statue of the "Basel Bronze Head" was scanned and reconstructed (G. Accardo, ref. [1]), as were the faces of Egyptian mummies using Computer Tomography, anthropological and Egyptology analyse. The bones inside the sarcophagus are virtually reconstructed without opening and destroying the mummy's envelop. Using these bones, the tissue is added on top of the skeleton, reconstructing the face of the deceased (K. Moltenbrey, Figure II.III.7 & ref. [65]).

Figure II.III.7 *Mummy representations, from left to right, a photo, a scanned cover and the reconstructed skul (K. Moltenbrey, ref. [65]).*

Scanned figures can be also animated making them "alive". The Terra Cota project is a good example of such work (N.Magnenat-Thalmann and al., Figure II.III.8 & ref. [51]). Another interesting field of work concerns old books or devices, too delicate to be accessible to the public (B. Eberhardt an all, ref. [27]). Virtually reconstructed, they permit visualisation and interaction allowing visitors to read them and perform experiments by themselves. Users can directly access explanations using online comments attached to the objects.

Figure II.III.8 *Picture of a Xian soldier extracted of the "Xian Soldiers" film (N.Magnenat-Thalmann and al., ref. [51]).*

II.III.III Entertainment, therapy and animation films

In this group, we find games, with recreational or educational goals. Nowadays, it is common to adults and children to play games such as fighting games. In this field 3D, 2D&half and 2D are very well mixed for time rendering, triangle optimization and the entire game industry thrives for increasing inter activity and realistic rendering of 3D worlds. The article by R. Rouse (ref. [93]) resumes quite well the problem of modelling or not the objects present in a game scenario. Modelling or not games objects in 3D depends on the goal of the game and 3D isn't an improvement in all cases. Another type of application concerns virtual therapy (D. P. Mahoney, ref. [53]). Other interesting facet, from the 3D creator point of view is the evolution of game production especially with the arrival of 3D engines. In specialised journals dealing with innovations in visual computing, we actually found a lot of advertising concerning 3D object modelling tools but also to create animation and interaction, such as in games. A lot of games are available everywhere, even in your local supermarket, but also engines to create your own game with all the technology required for manipulation and visualisation of 3D scenes. Computer games other the past two decades have exploded and originated a revolution in the toy market both for children and adults. With the evolution of technology they passed from 2D to 3D. At the same time, like commercial animation software ten years ago, commercial 3D game-engine are starting to be a marketable product. According to (B. Cramblitt, ref. [22]), these engines offer tools that can be divided into two categories: real-time renderers and graphic software development kits (SDKs) designed for games and others interactive applications and the general-purpose 3D game engines. The general-purpose 3D game engines go beyond rendering to offer a range of tools designed specifically for game development, including high-level object databases with culling, collision detection, animation, 3D sound, scene management, and interfaces to 3D modelling and animation programs such as 3D Studio Max. As technologies are moving very fast, more and more game developers create their games using a panel of licences adapted to their needs and covering at least the most sensitive part, which is the rendering step. More details can be found in (B. Cramblitt, ref. [22]).

Figure II.III.9 *Examples of a winter game (Supreme Snowbording) and a flying car game taking place in a futuristic city (Rocket Police) developed using game engines (B. Cramblitt, ref. [22]).*

Advanced gaming tools are often coming from film production research. Special effects tools are created for a film and later they appear in the 3D game market. The film "Toys Story", was one of the first films fully created with 3D graphics, it has certainly left its marks on the third art. The level of technology reached for these productions is incredible and Walt Disney Pictures and Pixar animation provide painters and animator with very powerful software and computers in order to obtain such realistic animation. The most recent technology useable for modelling, lighting, rendering, movie making, texturing and so on is integrated in this kind of production (B. Robertson, Figure II.III.10 & ref. [88]).

Figure II.III.10 *Woody in "Toys Story 2", note the lights, shadows and facial animation (B. Robertson, ref. [88]).*

A particular case of graphics art is the field of art pictures, snapshots of 3D models completed with nice textures and adapted lights, this subject is starting to be a complete part of the artistic domain. A lot of artists and architects aim to produce imaginative scenes, architectural designs and visualisation (most of the time very far from real-time display... a few days or months of computation). International design contests are organised each year, often by owner or vendors of commercial software, such as the archiCAD program. An example of such contest can be found at this address (ref. [109]) where more than 400 entries have been recorded for the 2000'edition. An interesting game, at the confluence between game and artistic field is the 3D CD-ROM version of Puzz-3D for three various models, *Notre Dame Cathedral, Victorian Mansion and Neuschwantstein Bavarian castle* (K. Moltenbrey, Figure II.III.11 & ref. [63]). After a complete construction of the scene, the picture is segmented into puzzle pieces. Users have to reconstruct the puzzle to build sections (floors, wall, roof) and then, they can move these parts into the 3D construction area for further assembly. Once the structure is complete, the real adventure begins, as the player navigates the highly detailed interior spaces in virtual reality and partakes in interactive games.

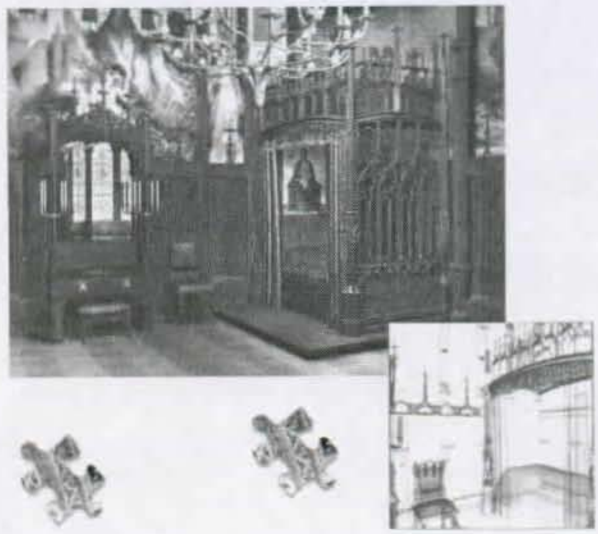


Figure II.III.11 *Because of its size, the Neuschwantstein Bavarian Castle was built as eight separate models, each ranging in size from 1 million to 2.5 million polygons.To achieve the richness of the imagery, such as in this scene and in wire frame format, the artists created more than 400 textures, with some source images as large as 200MB (K. Moltenbrey, ref. [63]).*

That before tackling industrial modelling using CAD for example, a nice field has opened through three-dimension modelling, is therapy using virtual scenes and educational aid. The realistic rendering of 3D worlds plus immersive tools such as head mounted displays put patients suffering of vertigo for example, in situations where the virtual aspect helps them overcome and gradually subdue their phobias. Results seem to show that patients can decondition themselves of their fear and this is a nice way for therapy (D. P. Mahoney, ref. [53]). Still in the therapy field we can find applications for surgery allowing doctors to train themselves for difficult operations. The body or part of the patient is modelled and the virtual model reacts to contact with the scalpel, for example. The surgeon can have various immersive display and force feed-back tools in order to feel as much as possible the resistance of the tissue (D. Sorid, ref. [101]). Once again modelling contains information about the "behaviour" of the tissue. Furthermore, the 3D models and modelling techniques applied are really dependent on the application using it.

MIT Media Lab provides an example of software dedicated to musical education, called Undescore (M. S. Dahir, Figure II.III.12 & ref.[23]). Through its visual display, Underscore lets users dissect even the most complex orchestral scores to gain a better understanding of the music. Accessing a sub-part of the music, the user can observe and hear a particular instrument or set of instruments, play several times the same movement and attach textual annotations to particular spots. 3D modelling allows the user to put into various 3D virtual plans notes or groups of notes and to have a better view of the partition. As for old books, this virtual representation permits new interactions and annotations of perhaps not available documentation.

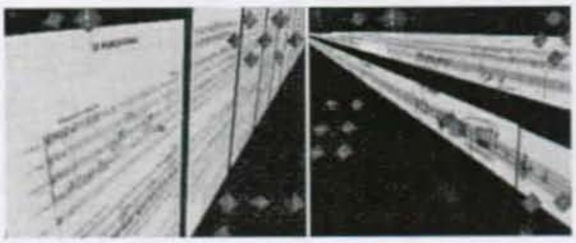


Figure II.III.12 *Visual display of partitions with various types of representations showing distinct parts of the music to gain a better understanding of the music (M. S. Dahir, ref. [23])*

Three-dimensional modelling in the industrial field has growing since the start of the 80s. Using various categories of Computer Aided Design software, more known as CAD tools, or Computer Aided Manufacturing tools (CAM), manufacturers starts with 2D modelling, then they start 3D modelling using Bezier curves for examples. This address (ref. [111]) presents a complete list of the products, companies, information and news concerning the CAD community.

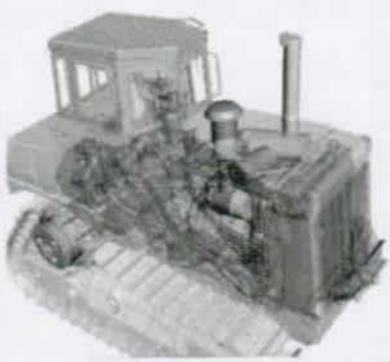


Figure II.III.13 *Virtual prototyping of a tractor built by Caterpillar. This enables engineers to explore numerous redesign alternatives before building physical prototypes. The final solution allowed Caterpillar to fit all new emission-control components into the existing tractor body, whose size and shape had been optimised over decades of engineering (K. Moltenbrey, ref. [64]).*

Figure II.III.14 *Cadcentre created this view of a veterinary products operation for a leading pharmaceutical company (C. D. Potter, ref. [76]).*

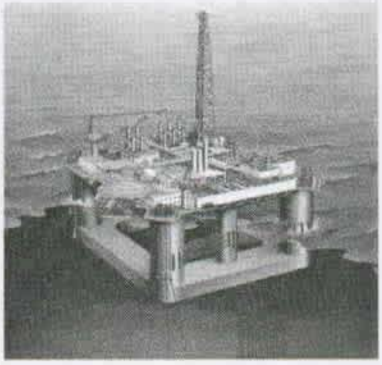


Figure II.III.15 *Norway's Statoil says it will cut 100% of lifetime costs of its North Sea natural gas platform using Intergraph's Data Warehouse System with Adaptative Media'3D model viewer (C. D. Potter, ref. [76]).*

Used first at object modelling ranging from tractor (K. Moltenbrey, Figure II.III.13, ref. [64]) to industrial site such as a gas platform (C. D. Potter, Figure II.III.15 & ref. [76]) or chemical firm plant construction (C. D. Potter, Figure II.III.14, ref. [76]), the evolution tended towards modelling of ergonomic problems including the design of humans with various characteristics, robot movements and interactions between the work place and themselves (C. D. Potter, ref. [77] & C. W. Reynolds, ref. [86]). Teams of researchers are working on the mobility analysis of robots in CAD models of industrials environments (i.e. C. Van Geem and al., ref. [108]). One goal is the feasibility study of maintenance interventions on industrial sites, these tools being integrated into CAD commercial software (ref. [111]).

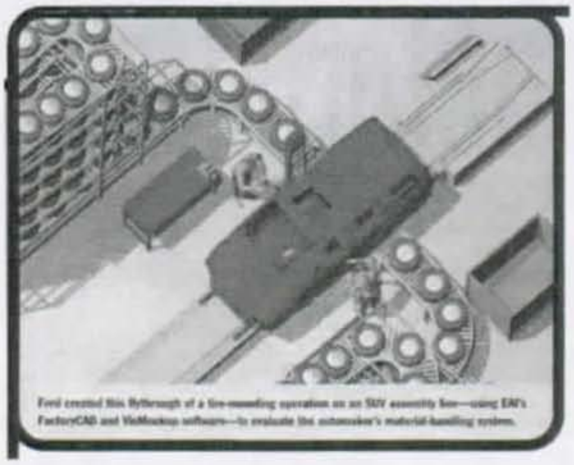


Figure II.III.16 *Ford created this fly-through of a tire-mounting operation on an SUV assembly line-using EAI's FactoryCAD and VisMockip software- to evaluate the automaker's material-handling system C. D. Potter, ref. [77].*

An example of a software is SimFactory2000 that allows simulation of a factory for its human interaction and/or robot motion. Optimization concerns several domains, from material flow, to manufacturing process or spatial location of units of production (C. D. Potter, Figure II.III.16 & ref. [77]).

Nowadays, public software is integrating both modelling and manufacturing strategy. After a step of inter-operability between various strategies and CAD software (A. Rowell, ref. [92]), the actual deal concerns the federation of the various trades and project conceptions.Various examples exist such as the management of the manufacture of 50 tilting trains in four years with a new methodology of work, based on sophisticated software such as Metaphas (webpages ref. [112] & ref. [113]). All the data is accessible by various teams involved in the project and the issue is to reduce both the time delay for the project but also the cost (product life cycle, employment, optimum organisation) and the documentation conception, among other issues, for maintenance and staff interactions. The arrival on the scene of Internet brings more and more accessibility to huge data directly from the office where an architect, i.e. can observe pictures of a specific problem without going all the times on the builder's yards (J. B. Novitski, Figure II.III.17 & ref. [70]).

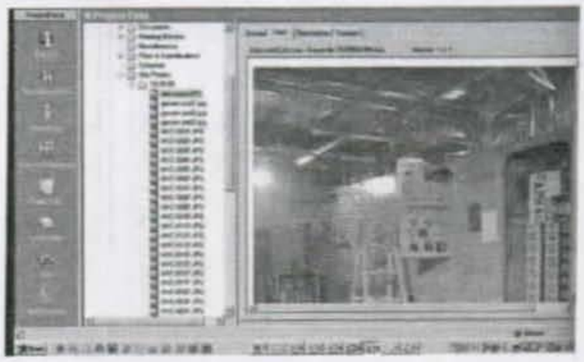


Figure II.III.17 *Web page coming from applications known as project extranets, project collaboration. This collection of technology is beginning to fulfil some of CAD's unkept productivity promise, while also pushing computer graphics into new application areas (J. B. Novitski, ref. [70]).*

CAD's activity area is impressive and 3D plant models are used from car design to factory simulation. Here are some examples of application in the industrial field. We have to note that CAD are used also for all the various domains concerning 3D modelling, from industry to art pictures.

II.III.IV Historical reconstruction and architecture

The task of producing urban models rapidly with a good degree of realism for various types of simulations is becoming increasingly important. Virtual architectural modelling can provide various kinds of services, from simple visits of "far away places" or disappeared ones, the model can also be used for simulations such as fire emergency exits or organisation of improvement such as in the industry, as above. First, an observation will be made about historical reconstruction with various examples, then models promoting future construction, a kind of 3D advertisement and finally, we will look at works more axed on modelling techniques and simulation of fire emergency e.i.

A new 3D model can have various origins. The danteum project, one of the basis model in the architectural field, concerns an ideal building that has never been built. The creator described, during time of war, an ideal building model, highly symbolic, and this mythical model took shape in virtual modelling after decades of imaginary representation (T. Nagakura, ref. [68]). The model may be constructed, damaged or destroyed (reconstruction of medieval architecture M. Masuch, ref. [57] i.e.), this concerns the field of virtual archeology. According to B. Roehl (ref. [91]) "First and foremost, virtual archeology makes it possible for researchers to share their discover-ies and insights with their fellow archeologists. Rather than trying to describe an excavated site using words or rough sketches, the entire scene can be reconstructed in three dimensions and explored interactively....

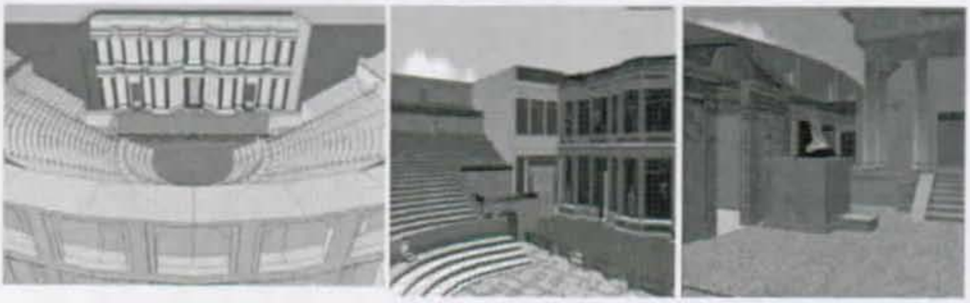


Figure II.III.18 *Views of Pompei reconstructed. The following is a series of screen shots of the Pompei Project. It was shown in 1995 at the DeYoung Museum in San Francisco, where it was a big attraction. A room full of people could see it on a projection screen, while one person with an HMD navigated through it. It ran on a Silicon Graphics Onyx-4, with 4400 chips and an the (then) new Infinite Reality graphics accelerator (A. Antonicelli and al., ref. [3] & ref. [114]).*

Archeologists can work like never before, working together to build increasingly detailed models of these ancients worlds. Not only can archeologists share their knowledge with each other, they can also display much of it to the general public. Up until now, the only way for most people to see historic artefacts has been in a museum." Avoiding museum's problems of space and damages (exposure to air, sunlight, vandalism, displacement...) these models have moreover the possibility of being accessed via Internet. For very old books, their representation via electronic copies give them new life (B. Eberhardt and al., ref. [27]). Figure II.III.18 is an example of a reconstructed scene using old descriptions and last remnants of disappeared civilisations such as the city of Pompei (A. Antonicelli and al., ref. [3] & ref. [115]). Some virtual scenes are freely accessible via Internet for educational purposes such as for teaching Egyptian history e.i (ref. [114] & Figure II.III.19).

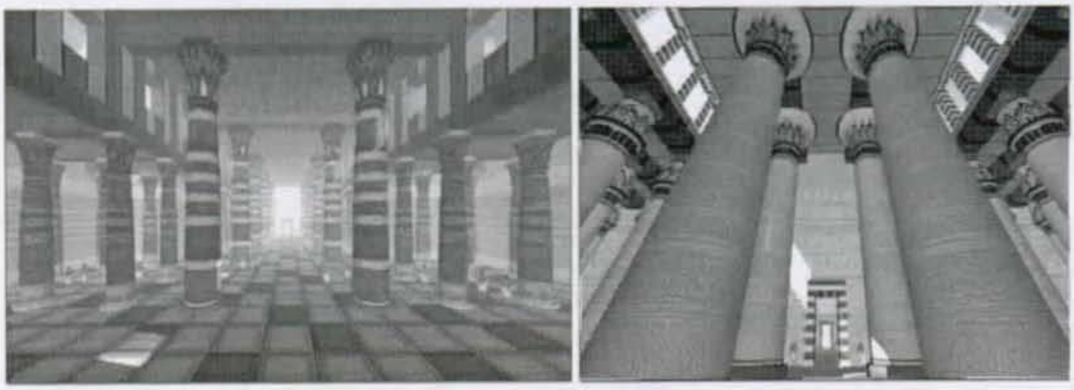


Figure II.III.19 *various views of a reconstructed Egyptian temple (ref. [113]).*

Other archeological applications present the evolution of a site, for example, the Vatican's evolution from the earlier settlements to the present. It provides a chronologically-ordered sequence of digital models representing the Vatican Hill, its plain to the South, and the North-Eastern slope of the Janiculum, as a topographical site developed from about 30 A.D. through 1940 (J. Silvetti and al., Figure II.III.20 & ref. [99]).

Figure II.III.20 *Views of the evolution of the city and buildings of theVatican (J. Silvetti and al., ref. [99]).*

Another use of virtual historical places is the example of the town centre of Derry for the Bloody Sunday lawsuit (M. Michal-Maguire, ref. [58]). Around thirty years later, the town-quarter was reconstructed and al. participants had to locate themselves spatially for the day of the 30th January 1972, to enricher their testimonies during the process. Using these technologies, all belligerent persons can precisely try to remember what occurred this day and determine the amount of responsibility, among others, of the army.

Present architecture is completely computerised. Although the main objective is the design of 2D or 2D and half planes, 3D modelling is often intended for overall views of new constructions or of modifications. Here are some examples of scene design in 3D for the presentation of projects or advertisement. The interior of the building in 3D, is often proposed to the customer to provide a view of what the project could look like. The next figure shows a kitchen design (D. P. Mahoney, Figure

II.III.21 ref. [54]). The 3D modelling can also permit the evaluation of a hotel's decoration, to carry out some tests simulating outdoor light and the general effect of the design (D. P. Mahoney, ref. [54], K. Moltenbrey, ref. [61]). Figure II.III.22 and Figure II.III.23 (K. Moltenbrey, ref. [61]) are pictures taken from the reconstruction of the Bellagio Hotel. While Atlandia architects began sketching 2D CAD and 2D rendered images of the hotel-casino (36-story places and 3025-rooms: the most expensive resort), its virtual design department started constructing the project as a 3D computer model to augment the physical model used by the architects. This enabled project planners to quickly and easily evaluate design options at the same time that it allowed Mirage Resorts executives to acquire a life-like perspective on all aspects of the project.

Figure II.III.21 *The architects at ReelGraphics (Berkley,CA) rely on 3d Studio Max to render models, such as this kitchen, that was built in ArchiCAD or AutoCAD (D. P. Mahoney, ref. [54]).*

Figure II.III.22 *By creating a real-time 3D simulation of the new $1.6 billion Bellagio hotel and resort, Atlandia architects and key executives could evaluate design options throughout all phases of construction (K. Moltenbrey, ref. [61]).*

Figure II.III.23 *The design group of Atlandia re-created the exterior and key interior areas of the hotel, which provided a realistic perspective of the project (K. Moltenbrey, ref. [61]).*

According to D.P. Mahoney (Figure II.III.24 & ref. [54]), the architecture domain is evolving towards visualisation. As clients become even more attuned to the technology's potential in other areas, they begin to expect no less for their own projects. This 3D modelling expectations imply additional work and difficulties for architect teams. Most architects don't have time, infrastructure and training for adding the visualisation level to their projects. Such an evolution implies that for common projects such as house building, 3D modelling tools must be very simple (parametrics objects with various levels), the geometrical model directly imported from 2.5D CAD software, lights and rendering phase completely optimised.

Figure II.III.24 *Image modelled and rendered by 4D Solution of Regina, Saskatchewan, Canada using Imagine software from Impulse based on AutoCAD drawings from Genesis Architecture & Engineering (ref. [54]).*

An application in building can also be test visualisation for the modification of existing ones, such as the project on Notre Dame's bell towers, an investigation for temporary adding modern spires to the top of the Parisian cathedral (K. Tyrka, Figure II.III.25 & ref. [107]). Studio Naco modelled the spires in 3D to use a visualisation tool for presentations to the government and church authorities. Furthermore, the virtual model was used to analyse the constraint forces on the cathedral due to the new parts of the towers and to verify material constraints inside the spiral straircase.

Figure II.III.25 *To help win public and political support for the Spires of Time project, Naco architects produced a 3D contextual model of the structure for use during presentations (K. Tyrka, ref. [105]).*

II.IV - The path planning

A lot of works have been done in the field of path planning. Major research subject in robotic, the path planning with image analysis coming from robot's sensor has a very large application domain. J. C. Latombe made a very complete overview of the state of the art (ref. [47]): "During the last three decades motion planning has emerged as a crucial and productive research area in robotics. In the mid-80's the most advanced planners were barely able to compute collision-free paths for objects crawling in planar work-spaces. Today, planners efficiently deal with robots with many degrees of freedom in complex environments. Techniques also exist to generate quasi-optimal trajectories, coordinate multiple robots, deal with dynamic and kinematic constraints, and handle dynamic environments. Motion planning have also found applications outside the traditional robotics field, such as design for manufacturing, graphic animation and video game software, minimal-invasive surgical planning, and molecule binding and folding. Problems arising from these domains are shaping motion planning research at least as much as robotics problems". In this reference the author presents the early work and the new applications such as Graphic Animation. It appears that when someone mentions the fields of path planning the first step is to evaluate the needs, several concepts could be useable at this moment. Our constraints can be defined as follow:

* We want as output not a list of points but a list of surfaces in order to let the behaviour control define the adapted behaviour and adapt the mobile entity's displacement with an optimal space occupation.
* These surfaces must be adapted to the specific mobile entity we want to route (notion of semantic associated to places),
* The scenes are very large (town level) and three-dimensional, as a result space cell decomposition like S. Bandi's work (ref. [9]) is impossible (too many cells and problem of resolution for the grids due to the disparity between small objects such as mail boxes and very large surfaces such as rolling ways).
* All the data must be in 3D due to the non-sensing of ground repulsion (data just in 2D like in most of the GIS systems).
* Each mobile entity as a constraint surface and this can be specific for each type of mobile entity (minimal surface for displacement, these constraints can be augmented for handicapped people).
* The time computation for path planning must be reduced as least as possible because it is just a "tool" for animation, most of the time is dedicated to time display both for virtual body and realistic environment.
* The virtual humans do not have the feeling of collision with virtual objects even with the floor. Furthermore the computation of intersection of all virtual humans with the scene mesh is impossible due to time computation
* The data acquisition corresponding to synthetic vision associated with image analyse must be simulated still due to time cost. Thus the location of objects and notions of ground must be provided by a database access.

Moreover, we can mention another three types of path planning research. First, the work of J.P. Laumond andT. Simeon (ref. [48]) with the notion of probabilistic road map visibility. Probabilistic path planning algorithms introduced at the beginning of the 90's, this method consist in making an optimal pavement of the space with notions of connectivity between these pavements. In the particular domain of 2D pictures, we can note the work of C. Strohecker and B. Barros (ref. [102]), creating path inside images of cities. Based on the mental images of large-scale environment, they propose some notions of districts, paths, edges, nodes and landmarks allowing structuring of virtual places with notions of orientation and navigation.

Figure II.IV.1 *Picture from Antz film production (B. Robertson, ref. [88]).*

In the animation domain, the film production of ANTZ *(B. Robertson,* ref. [90] and Figure II.IV.1) uses various techniques with path planning. As a productor of ANTZ said about the path planning: "avoiding obstacles, avoiding each other, and having a goal the ants try to get in the environment. The simulator can work with ants placed on a place with eight fields, this plane is then mapped onto the geometry of the environment. To place the ants on the plane in random way, animators paint a density map that directs the flow and the distribution. To place the ants more uniformly, animators can use a grid. The simulator can also cause the ants to follow a one-dimensional path to, for example, create a line of the ants walking along the edge of the wall. To create this path, an animator simply draws a curve in space." The article [89] presents a complete overview of the existing techniques for animation of crowd via various public software.

II.V - Display optimization

There are a lot of methods to refine a mesh model during or before its display. We can categorise these methods in two sets: re-meshing method (the mesh of the object is modified during the display) and the choice/substitution of parts of various models without modifying the mesh of an object. A progressive mesh method presented by H. Hoppe (ref. [40]), or simplification of envelopes by J. Cohen and al (ref. [20] and ref. [21]) are some examples of such technology. Applied on terrain surface with quadtree techniques the project of TerraVision (M. Reddy and al., ref. [83]) adds a new layer above mesh representation attaching various types of information.

Another technique mixes three-dimensional models with simplified representation for distant part of a scene using images in background (D.G. Aliaga and al., ref. [2]). Dedicated to interactive large architectural model walk-through, T. Funkhouser and al. (ref. [35]) present an algorithm of display optimization based on cell-to-object visibility. The building model is a set of objects, each of which can be described at multiple levels of details, and contains an index of spatial cells with pre-computed cell-to-cell and cell-to-object visibility information. As the observer moves through the model interactively, a real-time visibility algorithm traces sightline beams through transparent cell boundaries to determine a small set of objects potentially visible to observer.

II.VI - Conclusion

The purpose of these rapid and incomplete overview made in this chapter on the five human senses, body and cognitive modelling, actions and behavioural simulations concerns the presentation of a database simulating such knowledge and perceptions. This database describes the surrounding environment of the virtual human providing mobiles entities with various types of inputs. This input is directly linked to the simulations wished by the users. Regarding the quantity of input knowledge and behaviour rules needed to perform a coherent complex behaviour for a virtual human, the database provides a tool for input creation, storage and tidying up. The database can store various types of data simulating various levels of perception/knowledge:

* The visual simulation (from the geometrical knowledge of the scene);
* The tactile simulation (obstacle avoidance or exact object location);
* The simulation of "urban knowledge" (for autonomous evolution inside an urban scene) needing at least visual simulation. Some actions or more complex behaviour can be linked to a place;
* The hearing and smelling perception simulations corresponding to direct modifications of the internal status according to a place. In this case the decomposition of the scene is done according to the localisation of the inputs modifying this internal status.

In order to create the scene decomposition adapted to creation of the information composing the database we rely on notions already used in previous virtual city building. Once the database is complete it can be used for different purposes such as path planning using graph and tools for a optimal display optimization.

The next chapter presents an example of scene decomposition adapted to the simlation of a inhabited city and a more generic definition of this model adaptable for other simulation. Furthermore, we have a look on rules and modelling techniques for the design stage and we try to complete these rules in order to diminish all the inherent constraints of our model.

III - The Informed Environment

III.I - Definitions and how to perform behaviour simulation.

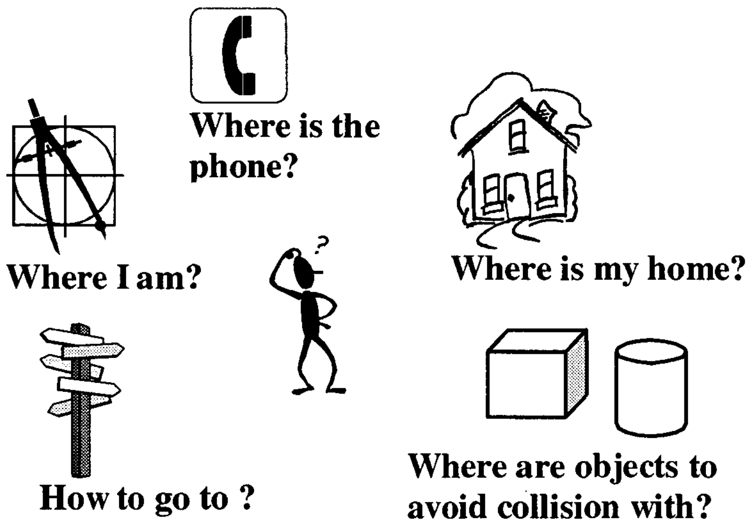


Figure III.I.1 *An autonomous agent in a virtual world*

We have defined **mobile entities** as objects with mobility such as pedestrians, cars, buses and bicycles. These mobile entities use certain surfaces (the **Environment Entities or ENV**) for displacement. An ENV represents a surface or volume and has associated semantic information. A single ENV can be composed of different kinds of objects such as objects perceived as obstacles (trees or walls for example) and objects used for specific interactions (moveable objects, doors or escalators).

Various methods can be applied for carrying out a simulation of an inhabited city. A set of elementary rules can be used with respect to various location characteristics, to define human behaviour during interactions with objects or with other humans. This method has the drawback of handling everything and the rules must cover all these topics. Thus, efficient rules for an urban context are too numerous and complex. Another possible method is to distribute information or "knowledge" to specific applications (N. Farenc and al., ref. [30]). In this manner, one application deals with internal crowd management (S. R. Musse and al., ref. [67], another one is dedicated to object interactions (hand position, movement of the body and the objects) using smart objects [15], a rule-based-behaviour gives high-level orders to humans (E. Schweiss and al., ref. [96] and A. Caicedo and al., ref. [17]) and finally, an application deals with all data coming from the environment. The latter application is our Informed Environment.

On the basis of this concept, as defined in [30], to a complex environment, we want to add information representing urban knowledge. A **complex environment** is characterised as a place where information (semantic and geometrical) is dense, and can be structured and organised using rules. The notion of **urban knowledge** encloses urban structural information and objects (complex environment) useable according to a set of conventions, and more particularly an association between places (geometrical area) and semantic information. The geometrical information originates directly from the three-dimensional model: our scene. One possibility for dealing with this kind of problem is to create the scene and during this process to associate information via an interface between the designer and a database constructor. The main idea is to add a semantic layer onto a core corresponding to a classical scene (ensemble of graphical objects) modelled using graphical software. The semantic layer associates objects with properties useable during simulation of urban life.

III.II - Hierarchical decomposition

Our model of Informed Environment scene corresponds to a set of Environmental Entities defining a database. In order to perform human simulation, the surfaces that we use must be at human scale, this implying a fine decomposition of the scene. In this fashion, with a huge complex environment such as a city, we have to consider the problem of dealing with a large quantity of data during access or manipulation. Our approach is to define some structured areas. The areas are either subdivided into sub-areas, or grouped, depending on the level of information. Thus, by analogy to a geographical map, we decompose a large area into sub-areas with information inherent to the level of description. At the city level, with the database, we can associate information corresponding to the main axes of the town for entering or exiting. These main axes allow crossing of the city.

At a lower level, these axes will be recognised as streets. In the database, the street level provides information about cross-walks and side-walks. As we use the notion of encapsulation, the same surface can belong first to a side-walk, then to a street, then to a block and at the highest level, to the city. This classification corresponds to a hierarchy, sorting and tidying up all the data. The city is divided into several areas, depending on their geographical and functional properties. Figure III.II.1 shows a graph representative of this structure.

Our model decomposes a city into twenty-nine different Environmental Entities, this decomposition is adapted to our needs for simulating inhabited towns (refer to Figure III.II.1). In the hierarchical decomposition tree, there are ENV's corresponding to the same ENV type but with different ascendant ENV, such as, the circulation area type. This type is presented in the parcels or in the building or at a floor level. They are distinct due to their localisation in the hierarchical decomposition and due to the fact that their functionalities can be different. These ENV can either be decomposed into a set of other entities (they are ascendant ENV) or not, in which case they represent leaves in the hierarchy tree.

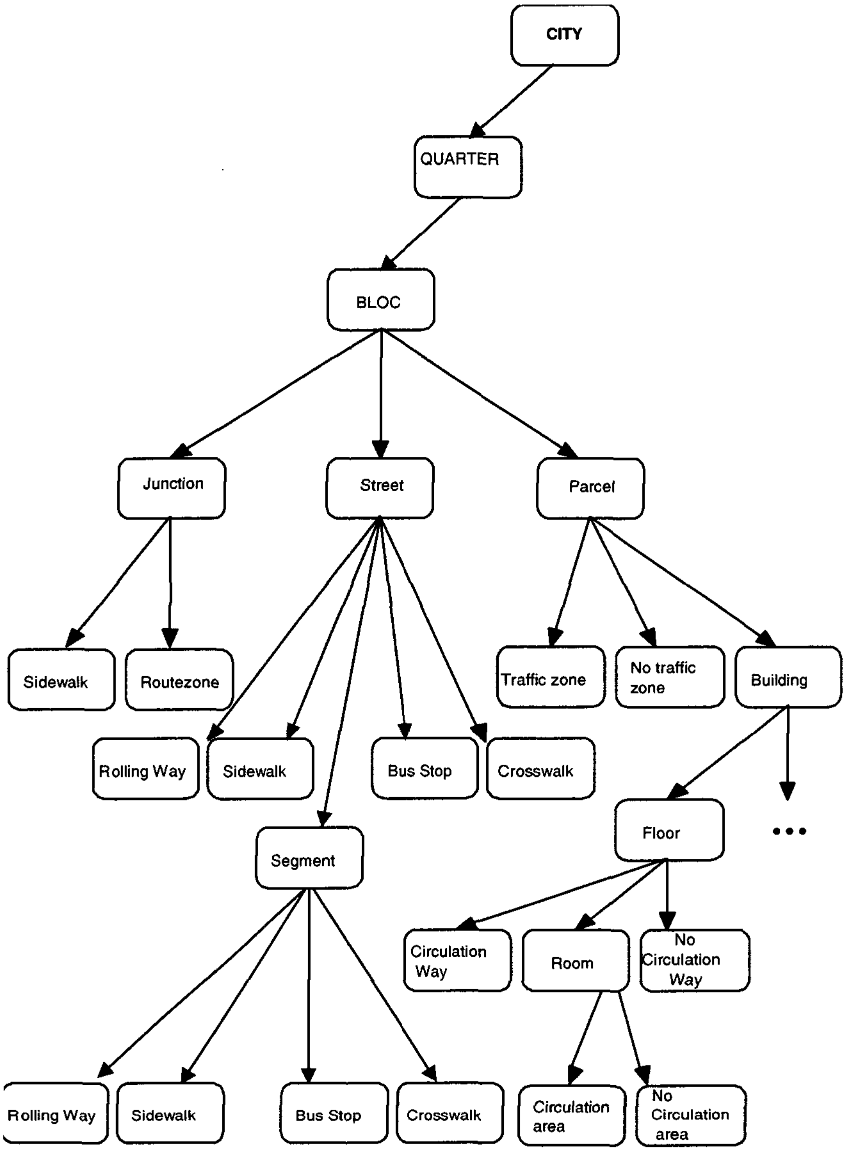


Figure III.II.1 *View of the hierarchical decomposition*

Figure III.II.1 presents the city decomposition into "quarters", the quarters are decomposed into block levels. We define a **block** as a part of a quarter, composed of **streets**, junctions and "parcels". **Junctions** are crossroads that connect streets. We consider as a **parcel** a portion of land that has no streets or junctions inside, like a park or a piece of land with buildings. In this way, all the space covering a block can be marked out and all points on the surface of a block can be associated with an ENV. All these entities are themselves composed of side-walks, pedestrian crossings, rolling ways (traffic) for the streets, travelling or non-travelling areas for parcels.

III.III - A Generic Model: The Data Model

This model of hierarchical decomposition is defined in a generic way before the database creation stage. We chose this decomposition in view of the simulation we had in mind, with autonomous humans in a virtual city. For other types of simulation, other semantics may be more appropriate. For this, a script describes the decomposition model in a file defining a set of prototypes, the ENV prototypes. Thus, it is easy to define a new decomposition hierarchy for the ENV. All the links between entities are described. For the moment, we work only at the block level and only the ENV's under the block are used and known to the system. As we do not use any commercial GIS system, we also define how to store the ENV, the file names, the type of the ascendant ENV, the names of the sub-ENV and the ENV type. This ENV definition (an ENV prototype or the parcel data model) yields, in the case of the parcel:

Parcel	***name of the entity***
type : 1	***type of the entity***
nb_under : 3	***number of entities composing the entity***
parzci	***list of the sub_entity names***
notraffic_zone	
building	
type_eng_up : 0	***type of the up entity, in this case the block***
label : INT	***name found in the scene for the parcel***
name : PAR_#2#_INT	***name after re-naming***
rename_from : 0	***type of information used to rename the ENV, in this case the name file***
PAR_#2#.wrl:	***format of the file name***
name file : 1	***number of file where we can find this ENV***
PAR_#2#.wrl	***name of the previous files***
store_file_name : DEF_FILE_PAR	***name of the file to store the ENV.***
type_mobile : 5	***mobile entity able to use this ENV (pedestrians in this case)***

III.IV - Object Naming and Re-naming

The designers of a scene and the database creator have in common the urban scene decomposition that gives names and types to the objects in an urban model. This decomposition has been defined in collaboration with a designer (O. Aune, ref. [6] and N. Farenc and al., ref. [30]) in order to create a syntax and some constraints useable by both scene and database designers. This kind of interface provides facilities for scene creation. The objects are not fully named due to the size of the name at a low level in the tree; the designer only designates scene parts by abbreviations. In this manner, according to the model presented in the next figure (Figure III.IV.1) we can find the following categories of names:

BLOC_xx_INT
CAR_xx_INT
RUE_xxxx_INT
PAR_xx_INT
VRxx
TRxx
BUSAire_xx
PIETON_xx
ZNCxx
ZCIxx
so_xx
FLOOR_xx
ROOM_xx

The solution we have adopted is to use labels above groups of objects in order to re-name all the entities before database construction. The ENVs are identified through their names, unique keys in the scene and in the database. The labels associated to the current ENV and all ENVs higher in the scene hierarchy compose these names.The next Figures present the names given by the designer and the re-naming using labels or file names. With these labels, an entire urban scene can be modelled, covered and analysed in order to be stored in a database. Another positive aspect of such nomenclature is the possibility to associate to a class of objects, such as a circulation area (ZCI), a basic semantic and to refine and complete this information according to its place in the hierarchy (and then to its final name and type after renaming). A exterior ZCI is an area where pedestrians, cars or other mobile entities can manoeuvre, except for buses. A ZCI inside a room can be only used by pedestrians.

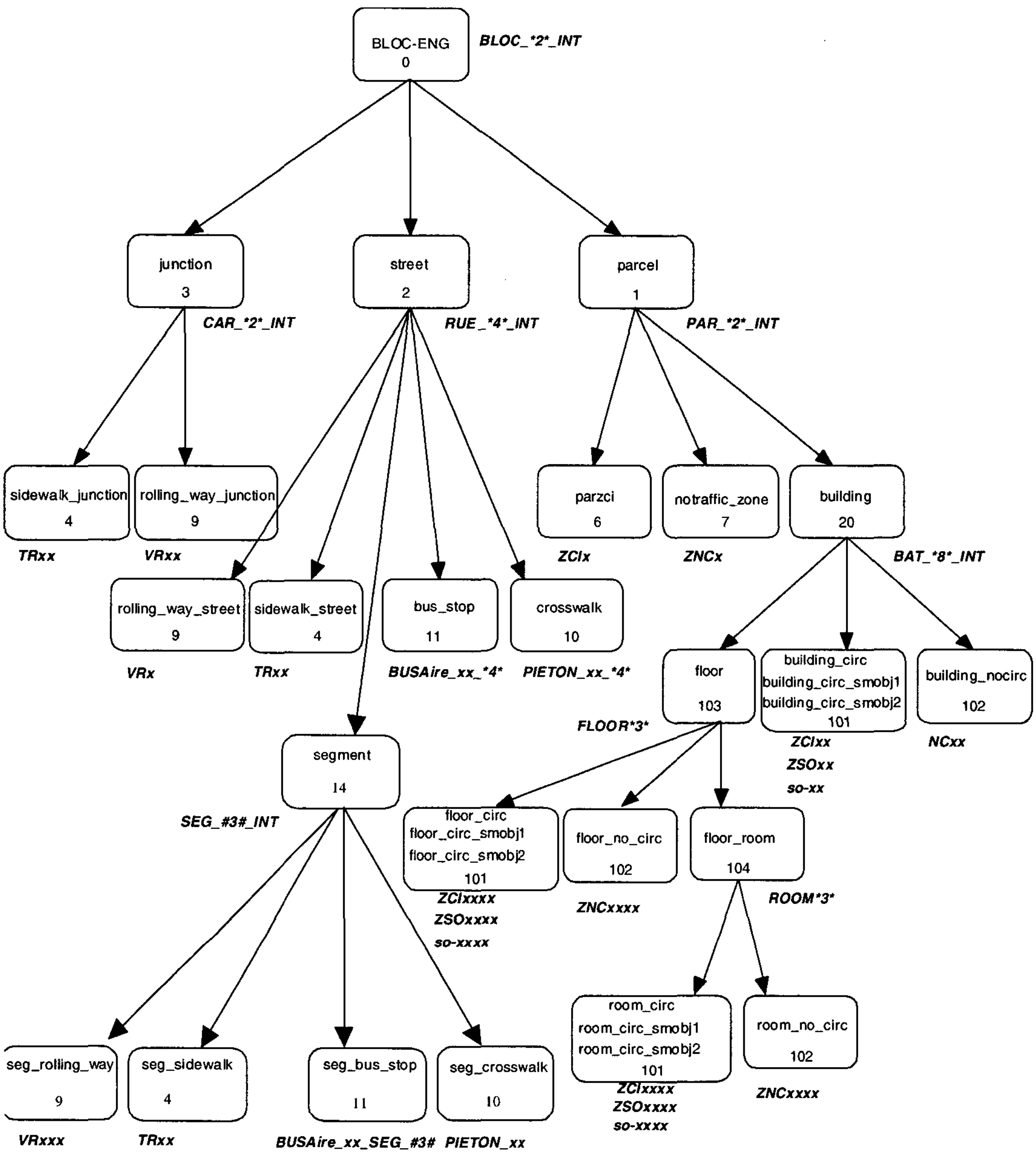


Figure III.IV.1 *Scene graph decomposition used for the design*

An example of this short nomenclature is the case of a side-walk. The designer gives the object the name "TRxx" for a side-walk, this object is under a group of objects in the Inventor scene hierarchy (a set of SoSeparator nodes in Inventor language (ref. [73])) labelled "RUE_VH01_INT" representing a street segment composed of the observed side-walk. This entity represents the entire street segment containing several cross-walks, side-walks, rolling ways etc. according to the model of decomposition used. By using part of the street name we are able to fully name the side-walk "RUE_VH01_TRxx". The link between the side walk entity and the street entity is established at this stage by the new name of the object. All the objects are named by the designer based on the object scheme naming previously defined. Applying the rules of ENV re-naming, the entire scene is read and all ENVs are fully named.

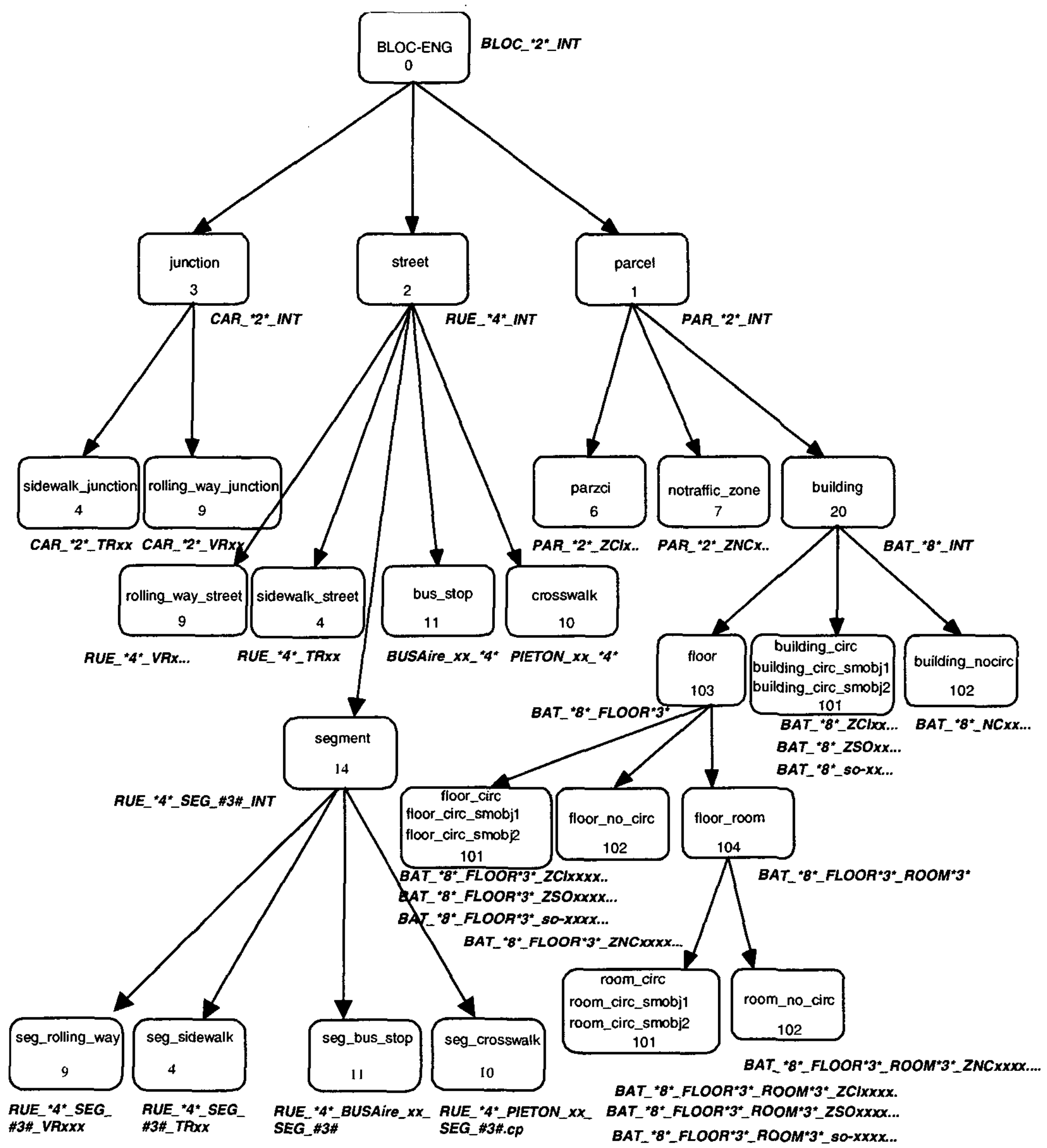


Figure III.IV.2 *Hierarchy name in the scene after the re-naming step*

Figure III.IV.2 shows the names found in the scene parsed for database creation. The next chapters define the database's contents and its use. It can be mentioned that the database format, once it has been created, is a textual set of files Thus, the visual scene used for the simulation can be the scene composed by entities described in the previous Figure, or another scene (the initial VRML2 format (see next chapter) or other format).

To conclude, this methodology has been tested in another work, it is included in the model of Gwenola Thomas (ref. [103] and ref. [104]) that is a modelling tool used to create the Rennes town reconstruction with car and pedestrians animation/ simulation in the city of Rennes. The software is composed by an automata system providing behaviour to vehicles and pedestrians for inhabited city simulations.

III.IV.I Conclusion

The three-dimensional decomposition model presented in this work is dedicated to the simulation of inhabited city environments and is based on visual and urban knowledge coming from vision simulations.

We have defined our model for the spatial decomposition according to all information coming from the different models defined by researches on the spatial representation, simulation of perceptions, knowledge modelling, behavioural simulation. We adapt these notions to create a relevant hierarchical model for the spatial decomposition: our generic model combining access to geometrical data dedicated to action planning for virtual humans and semantic notions in order to provide complementary information for an adapted behaviour to the scene localisation.

III.V - How to design/segment a scene

III.V.I A well segmented scene

Human animation simulation inside a 3D scene requires information about the geometry of the scene. The scene must be segmented according to the data needed such as the plan of the floors, for making a virtual human cross a room, for example. This segmentation is adapted to the required simulation. In any case, objects such as the floor must be recognised at least by their own geometry, through its name if a syntax for all the objects is applied on the scene. Thus a model, in a building, with only a global mesh, for example, will be very difficult to use as the scene is not segmented. The display can take too long time for real-time application if the scene is too huge and not segmented without information about a coherency between the sets of triangles composing the scene. Some previous work talking about big modelling of city mentions layers concerning a same object, these layers take part of a good segmentation but there is a need for organisation between all this data representing a 3D segmented model. The generic model of decomposition presented before segments a scene according to the simulation the user wants to perform. With this generic model the entire scene can be named and segmented. We also adopt a coherent model for storing the three dimensional objects and its links in directory according to the generic model.

III.V.II A model of space description

In order to directly use a scene without adding any three-dimensional information concerning places forming a scene, the designer must use a model of space decomposition corresponding to the needs of the final application. This model segments the scene according to the semantic information linked to the place and has an influence on the simulation the user wants to perform. The semantic information can imply various behaviour in the case of inhabited city simulation. The best way to add information during scene modelling is to attach names to objects, giving them some characteristics without ambiguity. After various tests with designers, we arrive at a set of conclusions:

* Object naming must be coherent with the tool that will analyse these objects, this implying that there is a common rules set describing a fixed object nomenclature.
* The object names are not enough to carry the entire urban scene decomposition in cellular entities, so the designer must group objects and also attach meaningful names on them.
* In the case of several group dependencies, the final name for an object can be too long and too constrained (time and concentration) for the designer. For example, a side-walk inside a street in a block would normally be named: BLOC_01_STREET_VH01_SIDEW_02. The error level during the design process is too high and such a constraint is not useable by the designer.

III.V.III Object and moveable object location knowledge

In order to be able to move an object or part of an object during simulation, its own three-dimensional representation has to be localised in the scene model and for an easy use, to be extracted from the fixed scene. All the objects have their bounding box envelop computed and stored inside the database. For all these processings the objects' representations (the scene part of the scene) must be identified from the rest of the scene and labelling is the easiest method. The objects can be, as the entire scene, modelled by a designer providing names, or identified using more or less automatic shape recognition. As a result, from an initial scene, we can perform extraction of moveable objects creating new files containing the object representation with some links positioning these objects inside the static scene.

III.V.IV File organisation and 3D modelling

For 3D modelling, we are working with 3dStudio Max [11]. This software includes everything we need to produce vrml files (vrml importer/exporter, vrml nodes like Inline (ref. [118] & ref. [73]). We can then convert these files to the Open Inventor format via Casus Presenter (ref. [120]).

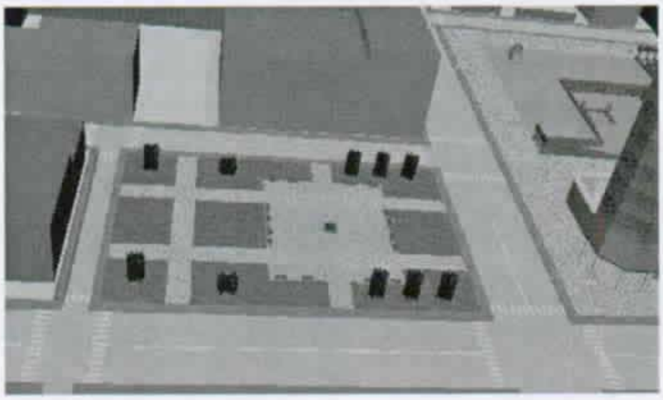


Figure III.V.1 *Side-walk selection*

The Figure III.V.1 represents a side-walk selection inside the city.

Figure III.V.2 *Icon representation for a SoSeparator node (ref. [73]).*

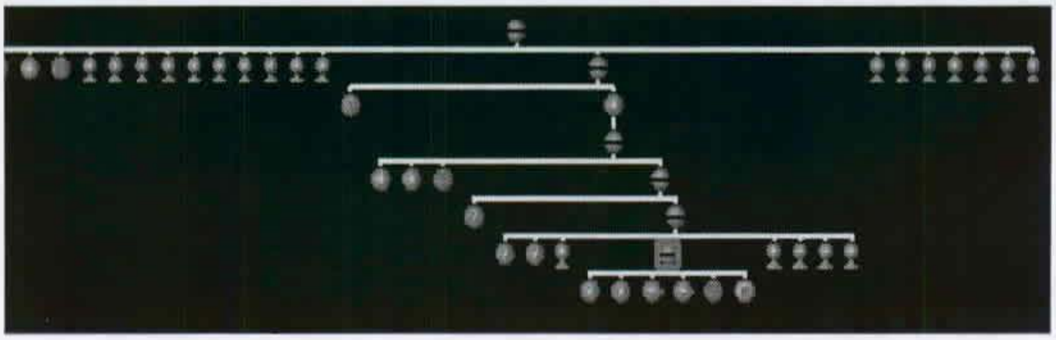


Figure III.V.3 *View of the scene graph representing the scene and in red the node representing the side-walk*

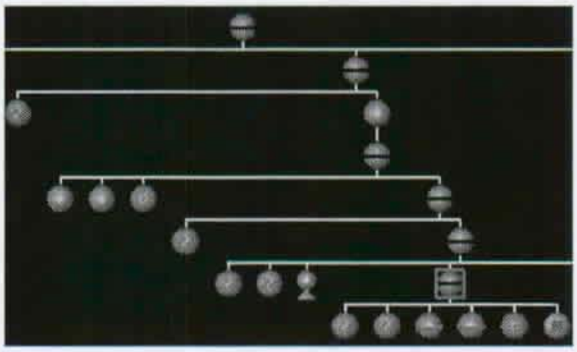


Figure III.V.4 *Zoom of the scene graph*

In order to illustrate the graph decomposition of the scene we can have a visual representation of the inventor tree-structure via icons (Figure III.V.3 and Figure III.V.4). Representing a separator node via the Figure III.V.2, the selection of the side walk corresponds to a path inside the scene graph. The node marked has a label "BLOC_01_STREET_VH01_SIDEW_02", in the SoSeparator nodes above we can find all the labels used for the full name construction such as "I_TRC_VH01", "BLOC_01".

Figure III.V.5 *Icon representation for a SoInline node (ref. [73])*

In Figure III.V.3, the icons representing some SoInline nodes (Figure III.V.5) under the SoSeparator node (Figure III.V.2) of the block (the upper one in the graph) are the various parts of the scene and are links to files corresponding to the parcels, streets and junctions. These inlines correspond to the generic model applied for the 3D design.

III.V.IV.I Hierarchical structure for directories

To save time at the design step, the reuse of objects with same geometry and texture is very important. To be able to import objects previously constructed into a scene, designers use a vrml node called “Inline”. This function is implemented in 3dStudio and allows the embedding of files. An "Inline" corresponds to a link with a pre-designed object file giving its location. This file location can be absolute (directly from the root) or relative (to the location of the created scene). If the designer needs to modify an object, the modification will be done only once and will be passed to all the instances linked via "inline" nodes. In order to handle object management, the designers store the entire scene in a file structure.

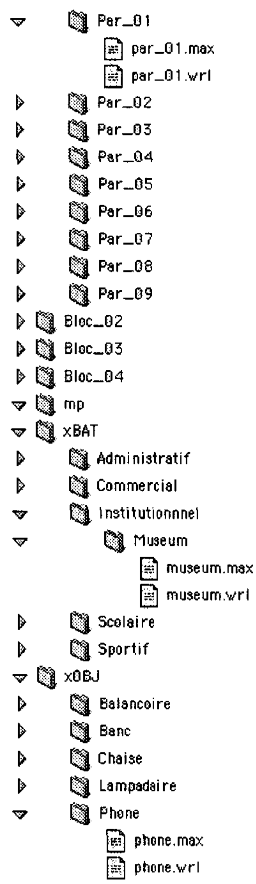


Figure III.V.6 *The directory organisation*

This structure tidies up all the object file representations and is necessary for handling LOD notions. Using the notion of decomposition needed for the database construction, we sort objects in relative directories in order to reuse some models and textures. The highest level is the quarter level that can be decomposed into blocks. This decomposition is the same for directories and at the quarter level, we find the texture directory (/mp) with the blocks, buildings, and objects directories. For texture management, there is a specific directory where all the textures are stored. The designer has to point in this directory in order to reuse the same textures as much as possible. Figure III.V.6 describes this structure.

According to some tests we have performed with various types of node for embed file, the "DEF-USE" tool seems to be efficient. This tool has the advantage of reusing the same object several times in one load operation (with some loaders). We have established that a "DEF-USE" use is efficient only with objects not directly analysed for the database construction. The database is composed of ENVs such as sidewalks that are part of a streets. They are “fixed” for an easier computation of their geometrical characteristics, whereas a tree or a public bench will be described via “DEF-USE” links. In the same manner, the textures are handled with “DEF-USE”. All the rolling ways are mapped with the same asphalt texture that will be loaded only once. As we have structured the object directories, we have structured the fixed and linked objects inside files. The next section describes a part of this organisation.

III.V.IV.II Object structure

To be processed and converted into an informed database of objects, the designer has applied the generic model described previously to object names. Objects with same properties are grouped.

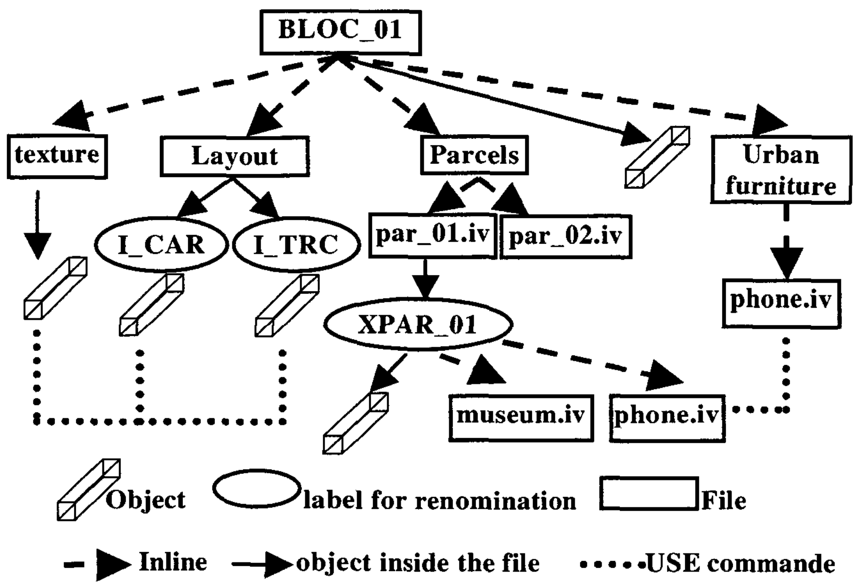


Figure III.V.7 *The scene file structure*

For example, at the block level, a block scene is composed of the entity corresponding to the surface of the block plus four embedded files corresponding to:

* The layout of the block, corresponding to all the street segments and junctions,
* The "parcel file" containing the list of all the parcel files of the block,
* The urban furniture such as public telephones,
* The list of the textures used in the displayed scene.

These four parts correspond to three "inlines", the names associated permit to easily exchange the included part in the scene graph just by modifying the included file. Thus for the layout file, we can exchange a heavy representation of streets with lots of details and precise texture mapping by simple coloured cubes. This notion corresponds to the notion of level of detail (LOD) and the naming of the associated parts is directly linked to the database. It is possible using labels to construct the scene adding a building inside a parcel, for example, but also to associate various representations according to parameters such as location of the relative point of view. Figure III.V.7 shows this structure of scene decomposition. Dot lines represent some "DEF-USE" links, hatched lines for "Inlines" and full line for included objects. For time rendering optimization, we use the notion of LOD in order to have various representations of a same object. The database contains all the data in order to handle in various ways this notion of LOD for display optimization (see section V.III).

III.VI - Conclusion

The main goal of this doctoral work concerns action planning in a large scale scene such as a town. In order to achieve human behavioural simulation we decide to focus on the definition of a scene decomposition allowing the creation of a database. This section presents an overview of the first step of this work concerning the 3D modelling of a virtual city integrating both geometrical and semantic data. For this purpose we decide to construct an imaginary 3D scene, with rules of storage and display options. These rules also include object decomposition and naming that will compose the database for virtual life simulation. The definition of "urban knowledge" and simulation of perception are the base of the generic decomposition model and are applied to our scene modelling. This generic model has been defined via textual description and corresponds to a common urban life simulation. For more precise simulations the model can be completed or modified. In this case, the designers labelling the entities and the objects have to take into account these new constraints. The next chapter presents the database structure created directly using the 3D model of the scene. We decide not to use a GIS system to store all the data, for speed reasons and ease of access. The imaginary scene was chosen in order to test and perfect the model of space decomposition. The 3D model can be a town reconstructed and segmented into a set of entities with names defined by the generic model such as for the Crosses project (see section V.V). The next pictures are some snapshots of a block. This block will be used for the database presentation in the following sections.

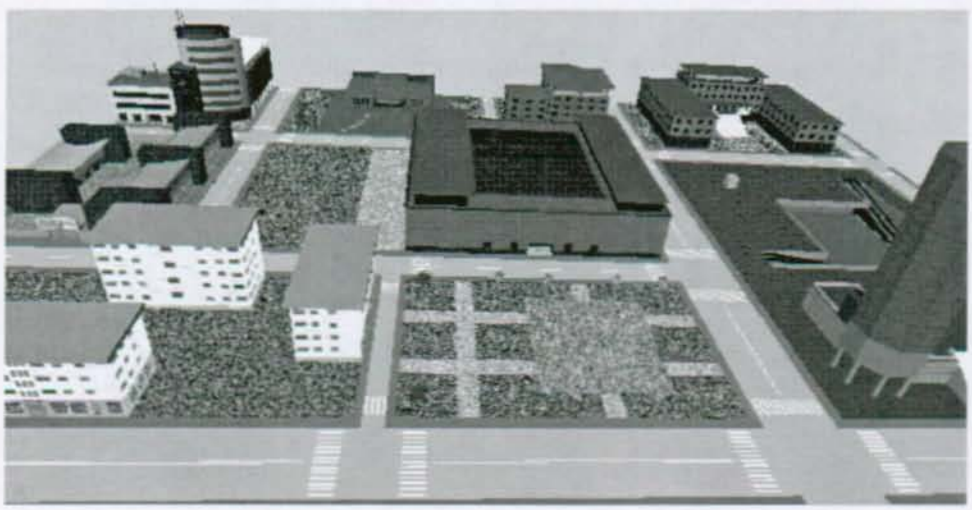


Figure III.VI.1 *Global view of a block, with a park, a museum, a station, a supermarket, a school, various buildings and a factory*

Figure III.VI.2 *View of the garden in front of the museum*

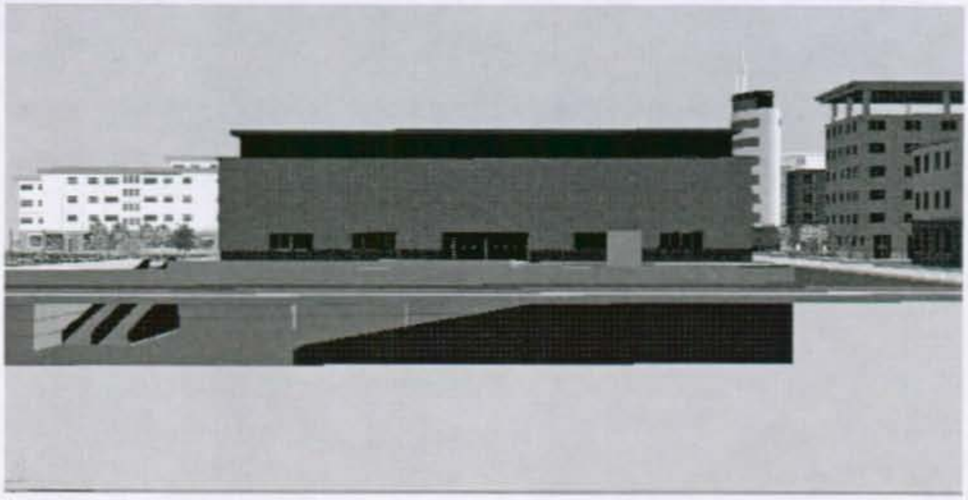


Figure III.VI.3 *View of the station with two stairs*

Figure III.VI.4 *View of the factory and some other buildings*

IV - The Database

This section presents the application handling the database for the creation and test stage, the analyse of the objects located inside the ENVs surfaces, the creation of the database by various methods more or less automatically applied, the possible storage of complementary data, and the visualisation of the data stored in the database for verification and modification purposes.

IV.I - The application "applicity"

The database is built, verified and tested using a graphical and textual interface coming from an application called "applicity". The applicity application lets the user accessing the command to start the automatically creation of the database, and after, the visual interface furnishes the more complete and correct representation possible for the database contents. The visual interface permits to designers to manipulate the scene in 3D and the graphical representation of the database contents. Thus it is possible to "pick" some parts of the scene for a manual creation of the database, to "pick" some points with the mouse in order to performed some path computation inside the scene, or to "pick" an icon representing the database contents in order to have more information about an ENV. Nine students in the scope of their projects (semester or diploma projects) have been integrated inside this development.

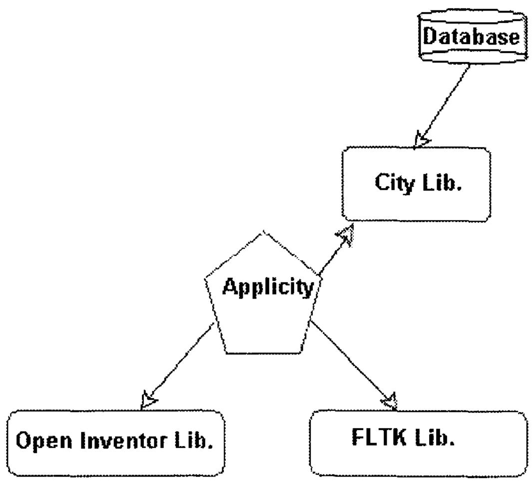


Figure IV.I.1 *schema representing the application and the links with the various libraries*

From a view point of the implementation, the interface is an Open Inventor Viewer for the 3D objects display, linked with an FLTK interface for the interface (buttons, menus, etc.) (see Figure IV.I.1). Applicity accesses to the data stored in the database through a library called "City Lib". City Lib allows to read and load the textual files describing the generic model (the hierarchical model of space decomposition) when the database as not been already created. City Lib is the core of the application and is tested inside this interface, once it is correct, this library is shared for the simulation of inhabited city (as a client inside the framework of the city simulation or for the CROSSES project (see chap. V.I, chap V.II and chap. V.VI)).

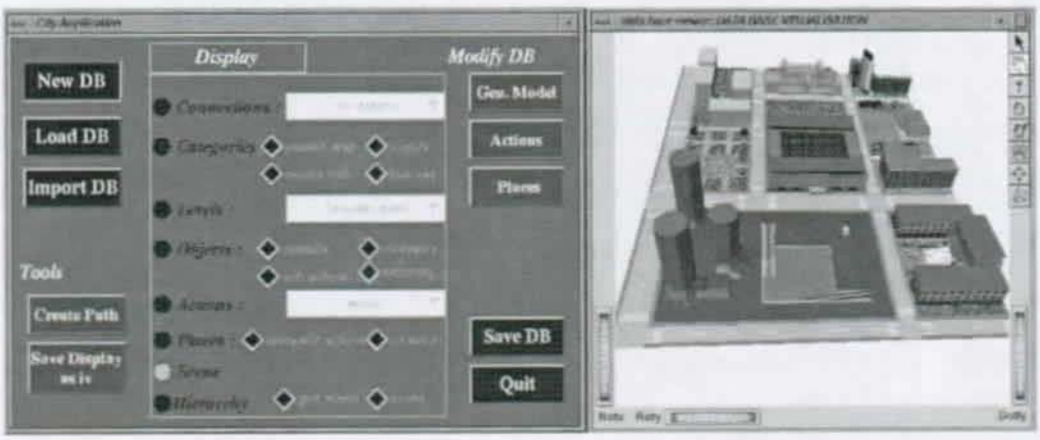


Figure IV.I.2 *Picture of the Graphical Interface with the main panel and the Open Inventor Viewer*

Figure IV.I.2 is a snapshot of the main interface of the Applicity application. Through this main panel, the user can visualise the database contents and call other panels for the database creation or the filling of the database with complement of data. All the next pictures representing the database contents were made using this interface.

The creation of the database is done through this application and the next chapter presents the automatically creation of the database with extraction of the different types of objects not used for the visualisation and the possible database creation by hand. Also, this interface allows to mix various databases, to add textual information and to display the database contents.

IV.II - Database creation

The inventor files composing a scene are parsed and all the Environmental Entities (ENV) are extracted automatically using the objects' names. These names come from the generic model defined previously and are composed after the step of objects re-naming. Still using the labels giving names to objects, ENVs are recognised but also some other types of objects. This labels can inform ENV, but also other objects belonging to the surfaces of the ENVs. These objects characteristics are stored inside the database for object collision avoidance, and if they can be moved or not visualised, their visual representation are extracted from the scene.

IV.II.I The Objects belonging to ENVs' surfaces

We have defined five categories of objects:

* The Environmental entity objects
* The moveable objects
* The fixed objects
* The lure objects
* The Smart objects (M. Kallman, ref. [44]).

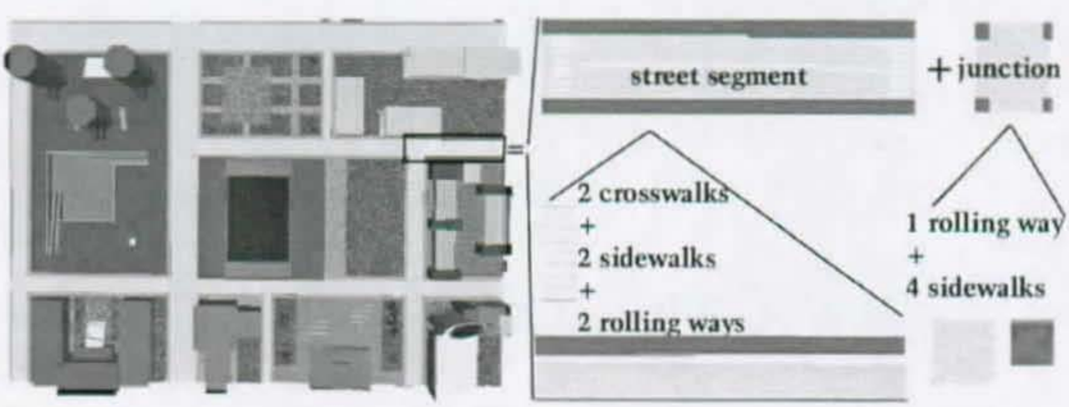


Figure IV.II.1 *View of the various levels of decomposition for a selected street inside a block*

The environmental entities compose the database, covering the space. Their labels correspond to the hierarchical model presented in the previous paragraph. Next figure (Figure IV.II.1) shows an example of our urban model with the underlying scene parts. Here is represented a street decomposition into cross-walks, side-walks and rolling ways. These objects are some ENVs inside the database.

The **moveable objects** are some objects which the geometrical representation is extracted to the scene in order to move them during the simulation. Their labels start with "MOBJ_" string and are above their geometrical representation in the inventor tree. An example of a moveable object can be a glass. The representation (as a part of the database contents) of this type of object is a blue bounding box around the object, whereas fixed object representation is in green (FigureIV.II.2).

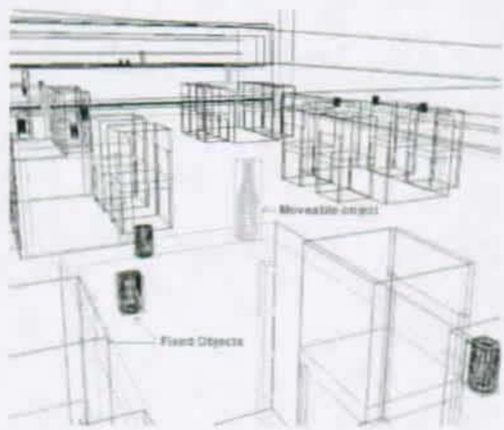


Figure IV.II.2 *Various types of objects inside the museum scene*

The **fixed objects** are not extracted from the scene, their labels start with "OBJ_" string. Their space location, name and bounding boxes are computed and associated with the ENV covering the surface where they are. Some requests to database furnish these information for collision avoidance or to inform on the location of this object in the scene. An example of fixed object can be a table (Figure IV.II.2).

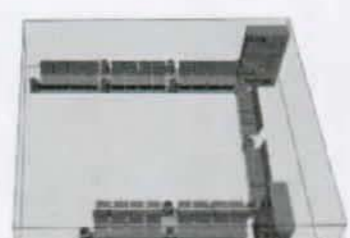


Figure IV.II.3 *Lure representing a global surface for fast computation of collision avoidance*

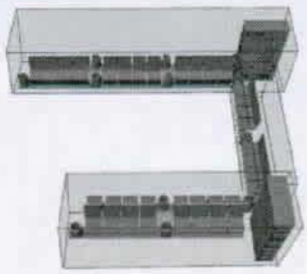


Figure IV.II.4 *Lure representing a set of surfaces around chairs for fast collision avoidance computation, but bringing more information such as a place location*

The **lure objects** are objects just bringing some information, but their graphical presentation is extracted in order not to be viewed during the simulation. The labels are "LOBJ_", and they are also associated with ENV where they are located. An example of lure objects can be a set of chairs inside a train station. A lure object can be a box representing all the chairs for a more easy collision avoidance computation (Figure IV.II.3) or various set of chairs for location knowledge such as "chairs in front the clock inside the train station" (Figure IV.II.4). Once the virtual human is near the lure objects, the real names of all the objects inside the lure objects can be provided by the database.

The geometry of the **smart objects** are also extracted from the scene and these files are given to the application which will store inside them particular behaviour and human interaction to make them smart (M. Kallman, ref. [44], Figure VII.II.2).

All the objects located inside the ENVs' surface are attached to the ENVs' through the database. For example, a room well modelled, has its internal surface as an ENVs surface (ZCI), the walls are unknown, the designer doesn't insert these objects inside the surface. Inside the surface of the room, we can find a chair and a table. If these objects have some recognised labels above their geometrical representation in the inventor file, the labels inform if the objects are moveable or not, just some lure objects or some smart objects.

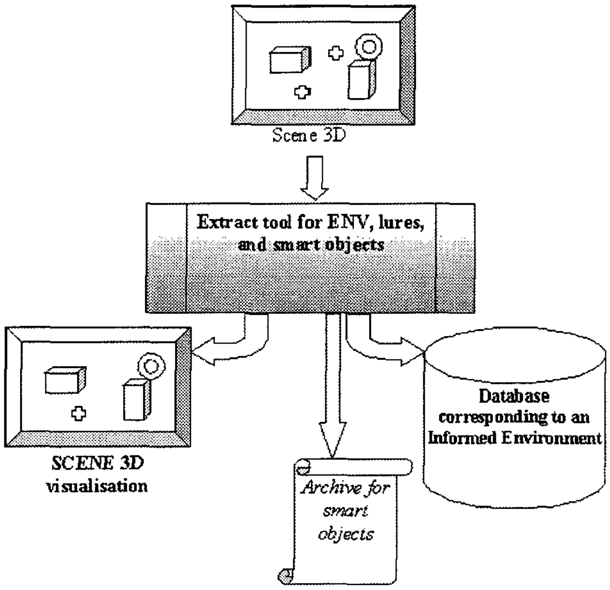


Figure IV.II.5 *Manipulation of the scene during the database construction*

From the 3D scene, the tool creating the database composes the database (a set of textual files), some archives for the smart objects and the movable objects, some complement of information via the lure objects' data and the three-dimensional files for the scene display (Figure IV.II.5).

IV.II.II Automatic creation

The database is automatically built using the textual and graphical interface of the application "applicity". Once all the objects not used for the visualisation are extracted from the scene, the scene is analysed in order to create the database. The database can be created at various levels, regarding the need of the user. For example, if the scene covers a block, the user can only create a database covering just a part of the scene with the contents of one parcel like the station for example. For this purpose the user defines the major entity used for the database creation, he chooses the major type of entities for the database from the set of type entities found inside the scene (Figure IV.II.7).

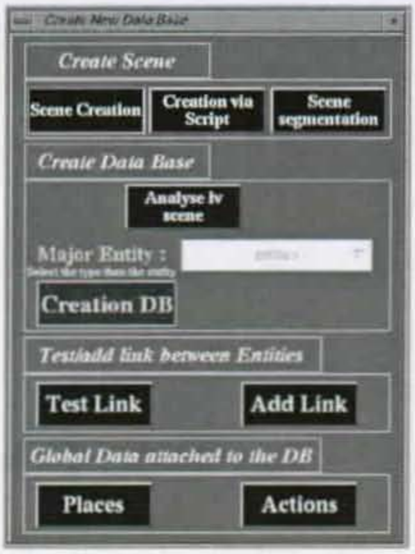


Figure IV.II.6 *Panel for the database creation*

Figure IV.II.7 *Example of all the types of entity present inside the bloc scene*

Once this type is determined, the user chooses an entity name inside the list of all entities with the type PARCEL inside the scene proposed by the application. In our example case, the major entity has the name "BLOC_01" and all the labels analysed and used for the database construction are under this entity inside the hierarchical decomposition.

The three-dimensional scene is parsed and all the labels (all the objects) corresponding to the description of the generic model are found and linked together according to the hierarchical model. The layout of the database is created and the space is parsed according to the modelling performed by the designer. At this step of the database creation, the database is composed of an array of ENVs containing:

* the name of ENV
* the type of the Entity
* the name and the type of the entity above it in the hierarchical decomposition, NULL if the entity is the major one
* the surface of the ENVs described by four points,
* the Entry/Exit point created from the surface for the connectivity link for the path computing,
* eventually the matrix for the placement of this ENV in the scene
* the name of the file containing the 3D model of the ENV, eventually all the files above if there is an overlapping of inline files before accessing the 3d object.
* the number and the list of all the children directly under the ENVs
* the geometrical connectivity between the ENVs under
* the number and the list of the various objects found under the labels corresponding to the ENV.

The objects are also stored in various arrays, and described by:

* the name of the object,
* the name of the ENV where it is located,
* the specification of the object (mobility...),
* the bounding box surrounding the object,
* the name of the file containing the 3D model of the object, eventually all the files above if there is an overlapping of inline files before accessing the 3d object.
* the matrix of placement for the object

The applicity deals with missing data if it concerns ENVs which are not leaves in the hierarchical decomposition. For example, a parcel is known by its labels and all the sub-ENVs, but its geometrical description is missing. In this case the application computes the sum of all the surface of the ENVs composing the parcels and applies the result for the parcel surface. During the database creation, the application insure that the ENVs overlapping corresponds to the hierarchical decomposition even if it can be false in the 3D model. A warning message is sent to the user but the hierarchical model is always respected.

IV.II.III Manual creation above an existing scene

The database can be created above a designed scene without rules for scene design. The objects are not obligatory created as separate objects and a mesh can cover a set of very various objects. The city application, via a GUI helps the user to select some set of triangles of some points in order to create the hierarchical decomposition. The user, through a GUI, selects in the Open Inventor Viewer some triangles present in the Inventor model or some points. A type and a name are associated with this geometrical description. An ENV is created, helping the user with the name composition and also with the links between the new ENV and the ENvs previously created and already stored in the database. The geometrical connectivity is computed or the user can compel a connectivity link, in this case he has to define the connection point for future path computing. The interface can be enhanced in order to be more "smart" and to verify data such as coherency between all the data provided by the user.

IV.III - Database creation from various databases

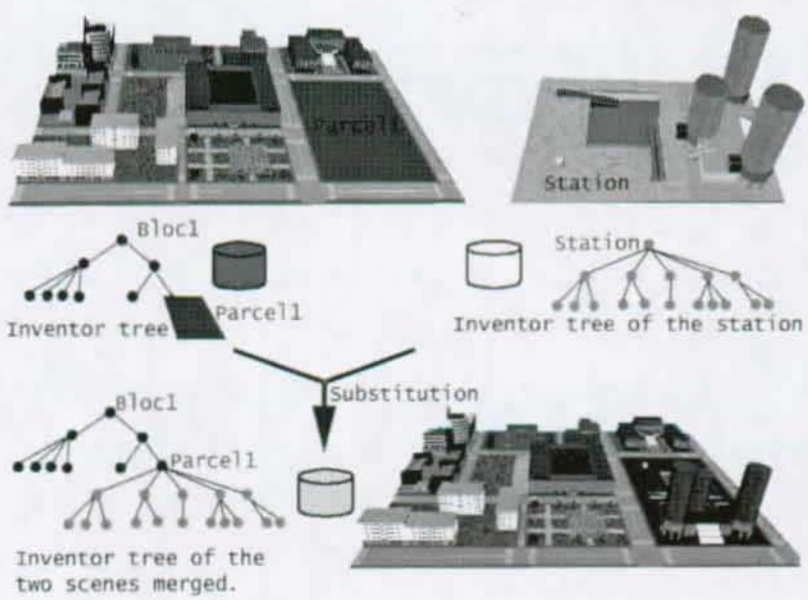


Figure IV.III.1 *Merging of two databases, one representing the station parcel and other representing the streets, junction and other parcels inside a bloc entity*

A database can be created using various database covering various part of a scene. For example, a user has created a database for a parcel, with the interior of a building such as a train station. This database has been tested several times and completed for specific simulation in the train station. The user can also, have two others databases, one for all the streets covering a block, with the cross-walks description and the rolling ways and a simplified description of the parcels, and the other describing more precisely the parcel representing a park. All these databases have been completed with various data not directly extracted from the 3D model. A database merging these three databases is needed for a simulation of a path planning with behavioural simulation from the park to the trains. In this way, the user can also create a composed scene corresponding to the composed database (see section IV.III).

Figure IV.III.1 represents how we have integrated the station used for a simulation (with its visualisation files and the associated tested database) and the rest of the block previously used for other simulation.

The database can also be manipulated, by deleting an entity and all its sub_entities for a more simplified database, or by exchanging a database part by other one. The user is responsible to visualise the 3d scene corresponding to the database used. Some data stored inside the database can inform the user about the files used for the database creation. As the database consists of a set of text files, the loader of 3D scene can use a various type of file, not an Open Inventor file type, for the simulation display. Thus, only if the user wants to load automatically the file where the database has been extracted from, most of the time, the displayed scenes are various from the initial scene used for the database creation.

A last option for the database mixing is to try to minimise the global size of the database stored in the memory during a simulation. A good example is the museum case. The museum can have a specific database attached to its interior, with data corresponding to information directly linked to such place (picture description, area describing historic period with summary associated etc.). In order to avoid a global database too much heavy, the user can decide to have in memory just the exterior data describing the streets, junctions and outside parcels. Only the circular areas used for the entry inside the museum are known. For path planning computation these ENV surfaces are known. Once the virtual human reaches one of these locations, the database can be switched to a new one corresponding to the inside of the museum. By this way the database content is always minimised at one moment, and this schema corresponds to a normal spatial displacement, consisting of reaching the entry of a new big place, and then taking an adapted plan to move inside. The connection of databases corresponding to inside and outside of an entity (in this case a building), is stored in the database by the user, at the level of the higher entity (the parcel or the building). The information concerning the management of some various graphical representations of the scene, in order to optimise the display can also be linked by the user at these levels as additional information.

IV.IV - Additional Information for specific simulations

Some additional information can be created and added above the layout of a database consisting of ENVs, Entry/Exit points and Objects specification. The database layout is a common information for all the simulation corresponding to the generic model used for the database layout creation.

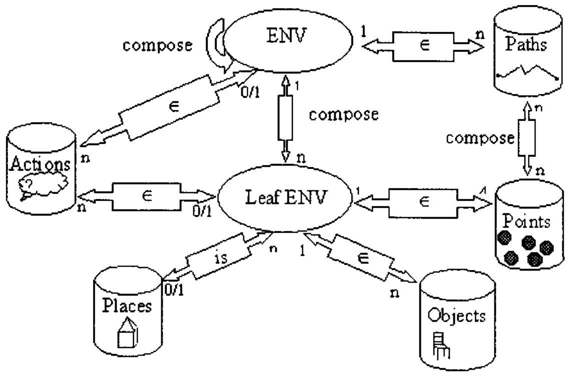


Figure IV.IV.1 *The database model (or schema) composed by the layout of the database (ENVs, Points and Objects present in the initial scene) and by the complement of data such as paths, actions and places.*

Another tables, specialised for a specific simulation can be defined by the user and inserted inside the database, linking more data to the space decomposition in ENVs. This information can concern pre-computed paths (see section V.I.III). A specific path for crossing an ENV such as a park can be linked and stored inside the database and given back during the simulation without path computation. This path can present constraints defined by the user such as specific points to reach during the displacement. Other type of information concerns some actions attached to an ENV. This can be linked to a generic type of ENV or directly to an ENV in the database. The actions are basic actions performed in a place, such as playing with a balloon, reading a book or talking in a park. A list of actions can be associated to an ENV such as a parcel. When a virtual human is present inside a sub_ENV such as a circular area the database furnishes these actions as information about "what to do in this place". The action can be represented through three types of information: a key frame (how to move the skeleton of a virtual body for a specific movement such as playing "hello" with its hand), or it can be some parameters concerning stairs i.e. (size of the stairs, number) that can modify a walk motor, or a rule/plan provided to a higher application such as a rule based behaviour. The information can be more complete, with behaviour and plans stored in the database such as actions. These plans (A. Caicedo and al., ref. [17]), associated to some places or ENVs provide, to applications querying the database, complex behaviour via rules. An example could be a behaviour for taking the train, buying a ticket, looking at the clock etc. Another types of data can be inserted in the database such as the some web links describing a place or objects via textual information. The data stored by the user depends of the simulation to perform. The last type of information concerns the notion of level of details representing various representation of an object, of a part of the scene. These fields in the database informs about the weight of the file in number of points, triangles, the memory needed for the display or the optimal range for the distance for the camera (see section IV.IV).

IV.V - Visualisation of the database contents

In order to verify and complete the database, we have developed a database visualisation tool. This visualisation tool has been divided in two principal data 's display through two windows: the main window for the scene visualisation and the second one for database representation. The main window shows the scene and provides to the users the possibility to select some parts of the scene. The database window displays the database content, or the scene file decomposition via inlines structure, or the data model. Furthermore we add a tool to create the LOD database complement which will be described at the end of this section. As explained in (S. Smith and al., ref. [100]), every description of the interaction techniques has various drawbacks. An informal approach of the description makes the model more inconsistent, and S. Smith and al. show that there is no obvious way to describe it. If an environment is not fully comprehensible, it makes it difficult to interact with.

IV.V.I Visualisation of the Environmental Entities

The application furnishes a tool to visualise the database contents, and the first thing to observe are ENVs present in the database.

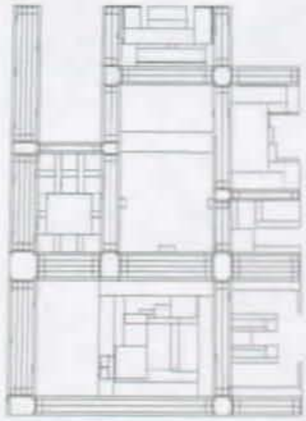


Figure IV.V.3 *Visualisation of ENVs at the level of side-walks, cross-walks, and rolling ways, with some areas of circulation inside the parcels, such as in the station and in the park, the entry for the museum and other buildings.*

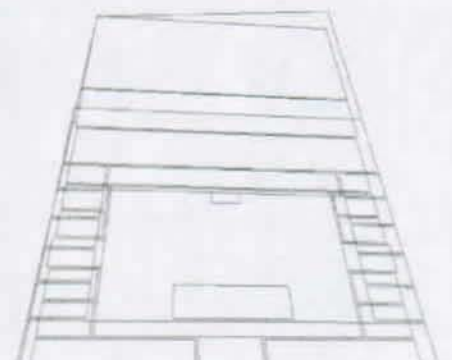


Figure IV.V.4 *Museum view with its specific set of ENVs corresponding to the surfaces of displacement known inside the museum*

Figure IV.V.3 and Figure IV.V.4 present the example of two separated databases, the layout of the block, with the entry of the museum, and a more specific database concerning the interior of the museum. Figure IV.V.4 displays only the ENVs composing the museum and complementary data such as objects and paths are also stored in this database.

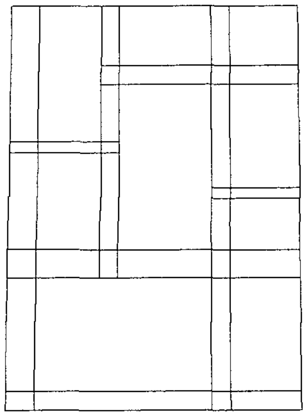


Figure IV.V.1 *Visualisation of ENVs at the level just under the block entity, representation of the layout of the streets, junctions and parcels.*

The user can select the level of ENV he wants to visualise.Figure IV.V.1 is a representation of the Environmental Entities composing the block entities.

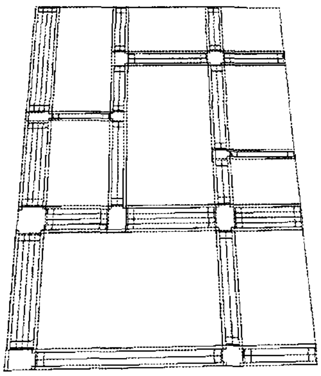


Figure IV.V.2 *Visualisation of ENVs at the level of side-walks, cross-walks, and rolling ways.*

Figure IV.V.2 represents a view of a database with only the description of the streets contents, the parcels are empty.

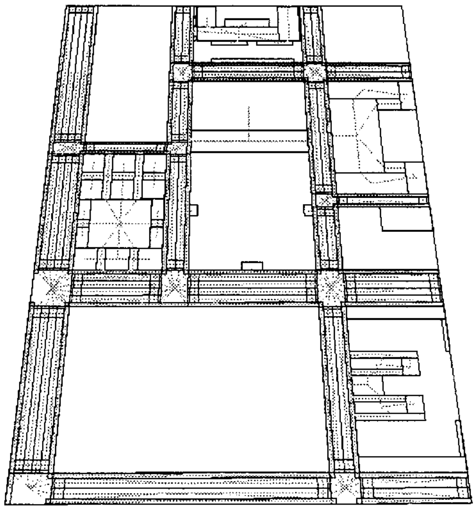


Figure IV.V.5 *Representation of the connectivity links in a block for all mobile entity types.*

Once the user has verified his/her group of ENVs, the geometry can be also observed via the connectivity links existing between ENVs. These connections can have also various levels, depending on whether they link some leaves in the hierarchical model or more higher entities. Figure IV.V.5 represent these links at a block level.

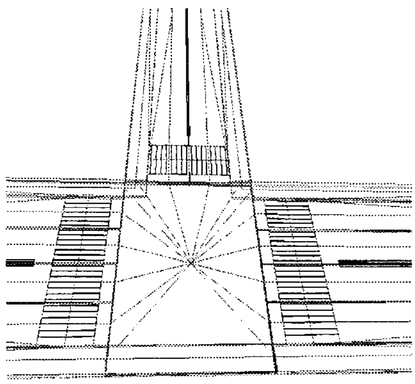


Figure IV.V.6 *Detailed view of the connections existing for a junction entity*

Figure IV.V.6 is a detailed view of the connections existing between the junction and all its sub-entities and the neighbourhood streets entities. Other information concerns the object present inside an entity surface. These objects are represented by their bounding boxes with various colours corresponding to their type: if they are moveable or not, smart objects or lure objects.

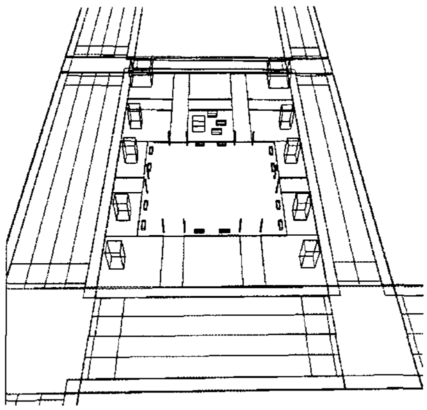


Figure IV.V.7 *Detailed view of the park with the objects inside represented by their bounding boxes*

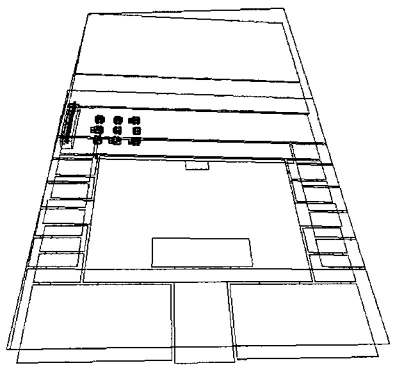


Figure IV.V.8 *View of ENVs composing the database associated to the museum scene with the objects present in this scene*

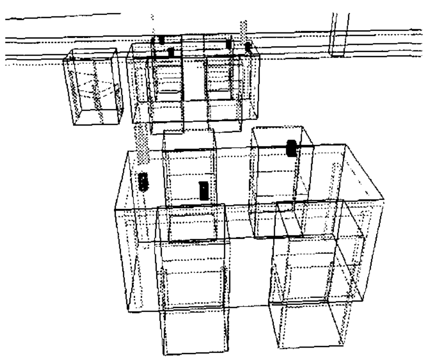


Figure IV.V.9 *Detailed view of the object bounding boxes present in the museum like tables, chairs, glasses and bottles*

Previous images show some examples of objects visualisation in various databases associated with the park of the block or with the museum database.

IV.V.II Visualisation of the inlines composing a scene

The notion of inlines composing a scene has no attribute to associate with, so we illustrate an inline as a sphere on a cube displaying the name of the file. The inlines are sorted in a hierarchical structure, which is displayed in the visualisation window. The main goal of this representation is not only to show the hierarchy but also to bring an easier way to manipulate the LODs characteristics. We have a couple of manners to achieve it.

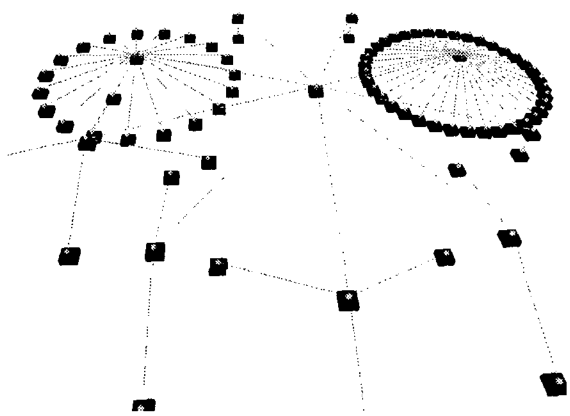


Figure IV.V.10 *Inline representation for the block entity composition*

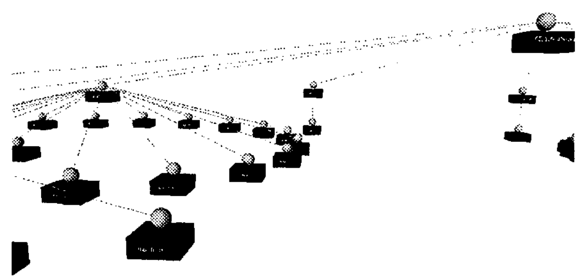


Figure IV.V.11 *Detailed view of the inlines*

The level of details are linked to the inline files, not directly to an object. We have two modes of selection: concrete object selection via the scene window, using a mouse and the "inline" selection. The selection via the scene visualisation can be just a part of the object or just a part of the inline-selected content. The second one allows the user to select directly the inline (a file), without knowing what objects it contains. The database contains the structure of all the files with their LOD and their embedded files. If the user has stored in the database comments about the weight of the various

LODs, this information and textual comments can be displayed in textual windows for user information. Another type of information can be associated to the inline notion in the database, it is the surface covered by the initial representation, the interface can help the user when he/she associates a new LOD providing information if their selected new representation has the same surface as the initial part of the scene.

IV.V.III Visualisation of the generic model by icons

As we have explained earlier, the generic model uses a hierarchical structure; hence, its representation follows the generic model's hierarchy with 3D icons. The main reason to choose 3-dimension icons over 2-dimension icons is to symbolise as near as possible the displayed scene. The user can manipulate the symbolic representation of the generic model and have various views (profile or top) in order to have a better perception of the model structure.

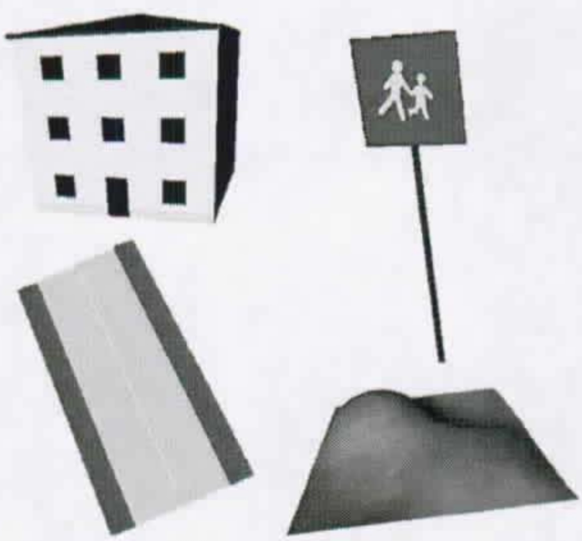


Figure IV.V.12 *Icon representations*

Continuing with the properties of the generic model, the entities overlapping is represented via links with colours depending on the entity depth degree from the root entity in the hierarchical structure. Inherent entity information is reflected through: the icon type associated with, its spatial location (plan location plus depth location) and via its name display. We present the strategy adopted for these attributes meaning/choice. In order to represent a hierarchical decomposition, we have to determine how we represent an ENV and all its sub-ENVs. We have to position these children relatively to the ascendant entity position. To represent the depth notion, we adopt a multi-layer approach, that can be viewed from a profile view. In this configuration all the icons situated at the same layer have the same superior entity. Hence, the layers mean the levels of depth in the hierarchy. Once the position in the hierarchical decomposition model is represented, the icons are simply arranged around a circle. Thus, all the children icons have the same distance from the superior entity. By writing the type of the object on all the faces of a cubic base, the user can perceive this information from all points of view. The colour of the name informs about decomposition possible into sub-ENV for an entity. The icon is attached on the top face. The icon choice is important for the understanding of the generic model. An icon symbolises an entity (a concept or a viewed object). Its drawing must reflect the most possible the semantic attached to the entity it represents. We classify the entities from their attached semantic notion in order to choose the most relevant icon (see Figure IV.V.12).

IV.V.IV Visualisation of the database via icons

The dissimilarities between the generic model visualisation and the concrete scene visualisation are few. The concrete scene visualisation respects the majority of the properties explained in the precedent section. Nevertheless, there are some key differences: the child's spatial position and additional information not coming from the generic model. The database contains notion of spatial position, which doesn't appear in the conceptual generic model. This spatial location is important for the association between the scene and the icon representation. Taking in account spatial location information makes the symbolic representation not only richer but also assures a more coherent representation (T. Catarci and al., ref. [18]). Location of ENVs is important and the visual representation must reflect such data. The Database provides the centre of the entity and instead of using a circle model for child's position, the location in the plane corresponds to the location of the entity in the scene. This approach gives us a clear and understandable scene, with a lot of semantics, making easier to recognise an object within the hierarchy.

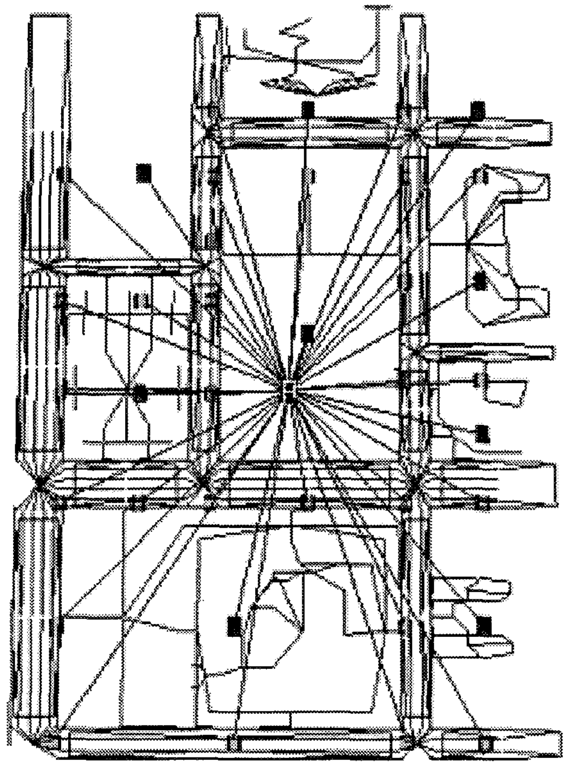


Figure IV.V.13 *View from the top of the ENVs representation through their surfaces and their icons.*

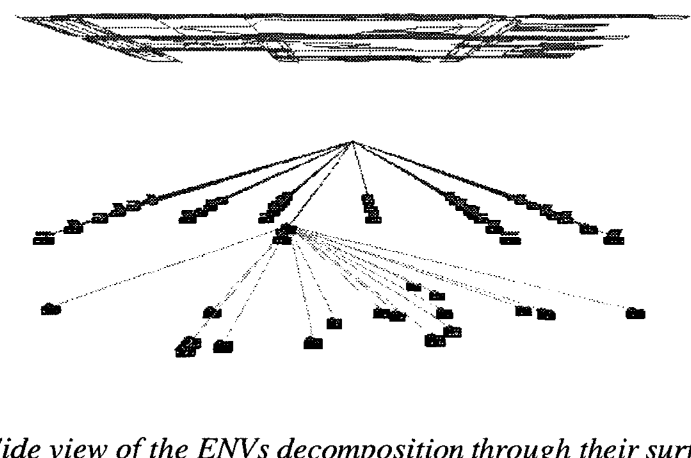


Figure IV.V.14 *Side view of the ENVs decomposition through their surfaces and their icons with two degree of decomposition for a parcel contents.*

Figure IV.V.13 and Figure IV.V.14 show various views of the database display. Figure IV.V.13 represents an up view with the ENVs surface in the front plane and the star corresponds to the block decomposition with the location of each block's sub_ENVs. Figure IV.V.14 corresponds to the multi layer notion, a side view, for parcel decomposition. This parcel has been selected and all the icons corresponding to its sub-ENVs are placed under it, such as in the hierarchical model. Figure IV.V.15 is a top view of the database representation with icons such as building, street or the block icon.

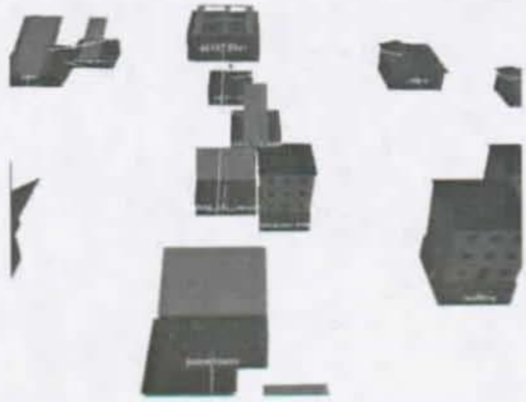


Figure IV.V.15 *Detailed view of the icons representing the dependencies between the ENVs with various levels of decomposition.*

Other important difference is an information window. An information window is showed as soon as we click over an object of the hierarchy. The goal of the information window is to give more data to the user about the selected object. It presents information such as object's name, number of children and other additional information stored in the database such as the list of actions commonly performed by humans in a place linked with an entity. Next figure (Figure IV.V.16) is a view of a parcel visualisation, with icons, spatial location, name display and an information window concerning data attached to this entity.

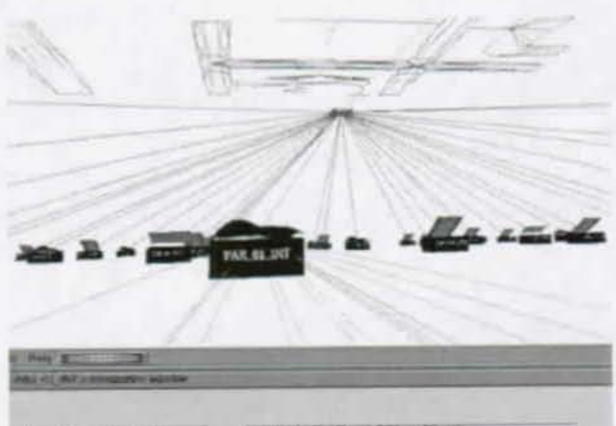


Figure IV.V.16 *Visualisation of the database hierarchy with textual information for a parcel entity.*

V - The DataBase Use

This chapter presents the various applications that can access a database corresponding to an informed environment linked with a 3D model. Once the database is created and archived, the user can load the files containing the data and exploit it independently of the three-dimensional model. The database exploitation responds to various type of request such as:

- Where I am?
- Where are objects?
- Where is a place?
- What are the common actions performed in this place?
- Where can I go?
- How can I go from here to this place?
- What is the faster path between these two places.

The database provide others aids such as help in scene composition via a scenario or during simulation, a set of tools for assistance to handicapped people or for emergency situation training such in the Crosses project described in the last section.

V.I - Simulation of spatial perception

V.I.I Where I am?

The first utilisation of the database is to locate the mobile entities inside the scene. Using the database and the Environmental entity surfaces, some routines determine if the mobile entity is inside the surface of major entity (i.e. the block). Testing the components of the block (street, parcel and junction) the location is recursively computed finer and finer to finally place the mobile entity upon a surface corresponding to a leaf inside the hierarchical decomposition. The hierarchical decomposition is well adapted for such process because it cuts down the computation's time and the number of required tests. Once the mobile entity is located inside a surface corresponding to an ENV, the database can provide plenty of data associated to this entity, to the upon entities, the neighbouring ones, the files composition etc.

V.I.II Places and objects location knowledge

The database contains geometrical and semantic information for mobile entity simulation. An area defined as an Informed Environment provides sub-areas (ENVs), along with the list of objects to be avoid for collision. Peripheral objects such as walls are not included in ENV surfaces (definition in the modelling phase section) in order to minimise the number of objects. The database corresponds to perception for being aware of all the objects inside an ENV or all the surfaces adjoining an ENV. The decomposition hierarchy makes no distinction between a park and a cemetery. Both are parcels in the city. In order to specify such knowledge, we add a label above the ENV definitions, thus allowing specification of a place. This additional characteristic permits declarations such as "go to the park", or specific behaviour or action definitions such as "in a park common actions are playing, reading and walking".

In order to route virtual agents in a city, the user has some difficulties to situate the place using the composed named defined and re-composed in the pre-processing phase. The localisation of the side-walk called "RUE_VH01_TR02" or the park name "PAR_02_ZCI32" is not at all useable during simulation to send order or to locate places. We have defined a **"pseudo environment"** which corresponds to some places such as a park or the school-side-walk, and these names have been provided by the user and attached to ENVs in the database. We can have several parks and various methods can be applied in order to know which one is mentioned. Using a GUI interface the user associates a place to various ENVs. Conceptually this association can link any types of ENV, but in practice the ENVs must be some leaf ENVs in order to be sure that we have a correct surface for its displacement. The associated database can be linked to the main ENVs database or just occasionally associate for a specific simulation. If the user specifies two parks the nearest will be chosen to compute paths for example. This part of the database allows orders such as "go to the supermarket, passing through the park" for example.

When a path is computed for a mobile entity (see next sub-section), the application querying the database provides the list of objects to avoid collision with. These objects are present inside the useable surfaces and they are declared as a global set of bounding boxes once the path is computed, or at each entry inside a surface as local objects. The first solution minimise the number of messages exchanges between the route module and the database but increases the number of tests for the collision avoidance during the simulation. The second solution increases the number of messages but minimise the computation of collision avoidance. The list of objects (mobile or not mobile) can be also delivered by the database simulating more realistically the visual perception according to the computation of the field of view of the mobile entity. Some tests have been done by C. Bordeux (ref. [12]), via a group of filters upon a list of objects provided by the database in order to determine the set of objects the virtual human has in its field of view with the reactions and actions associated with the objects. In other terms, using the direction, angle and distance from the eyes of the virtual human to the list of objects we can determine what are the objects perceived Another option is to segment the surface of a crossed entity according to the objects located inside (Figure V.I.1). With this sub-set of entities, a graph and the path associated are computed, providing to users only surfaces without objects inside.

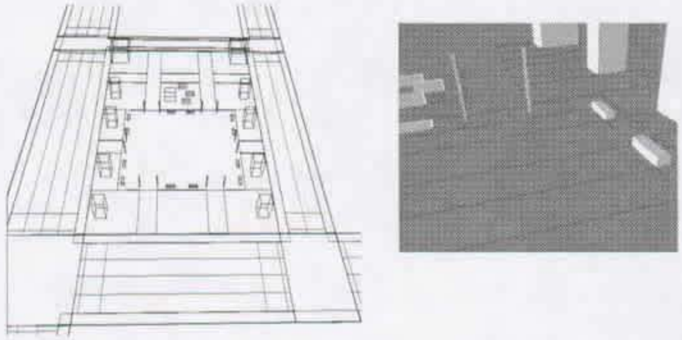
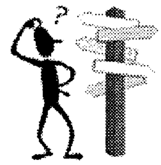


Figure V.I.1 *View of the entities composing the garden on the left side and the right picture shows a view of the segmentation of these entities according to the objects.*

In such configuration the user has to define some constraints concerning the minimum surface useable to go from one entity to another one and to be aware that the computed distance for the path does not take into account the objects collision avoidance.



V.I.III Where can I go? How?

Another way to use the database is to create paths by the means of our Informed Environment. Depending on the type of mobile entity (pedestrian or bus for example), the Informed Environment determines a path using the Entry/Exit points and the type of the ENV currently in use. In our system, we decide, as an urban rule, that a pedestrian cannot use a rolling way for walking. At a street level, a path for this kind of mobile entity can only pass through side-walks, cross-walk and circulation areas.

Taking into account:

* all the research works presented in the state of the art chap. II,
* our needs for path planning and behavioural simulations,
* all the data stored into the database,
* the time dedicated for path planning computation,

we decide to build a graph using all the entities present in the database. The edges of the graph correspond to the environmental entities and the entry/exit points belonging to sides of the ENV are the nodes of this graph. For each type of mobile entity we create a graph with the entities useable by this mobile entity. This representation is compatible with the notion of hierarchical decomposition, a edge in a graph can represent a sub-graph as well as a block is a set of sub-entities composing it. This modelling supports various level of graphs according to the notion of level of decomposition inside the space decomposition. The database can be used for the storage of main paths for crossing a block or the main axes for crossing a city or a country. These paths are associated to the environmental entity they cross.

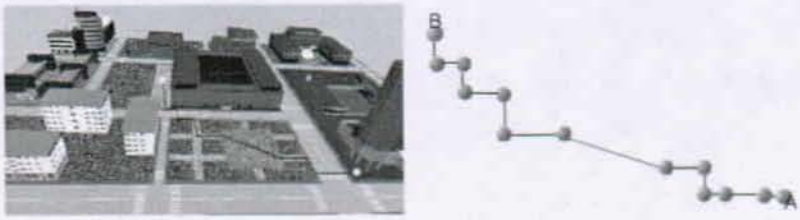


Figure V.I.2 *Path from the train station to the museum, at the level of the leaves of the hierarchical decomposition. On the right the nodes correspond to the connections between the entities represented by the edges.*

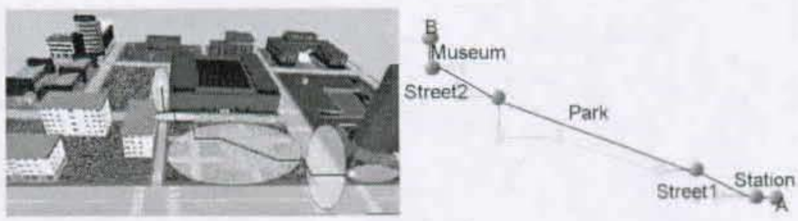


Figure V.I.3 *Path from the train station to the museum, at the level under the block inside the hierarchical decomposition. On the right the nodes correspond to the connections between the entities (streets and parcel in this case) represented by the edges.*

Figure V.I.2 and Figure V.I.3 show an example of path computation and representation for pedestrian at two levels of granularity. The path furnished to the tools user, corresponds to the first one with all the edges corresponding to all the circular areas, side-walks and cross-walks, and all the point linking these entities represented by the nodes inside the draws. There is a direct connection between the hierarchical model of decomposition and the path representation and computation. This notion of path with various level of sharpness is also used for path computation with constraints in the case of pedestrian with reduced mobility (see chapter IV.V).

V.I.III.I Computation of path planning

The first step is to use the database to define all the ENV and the associated usable points by the considered mobile entity. Consequently, for each ENV, according to the type of mobile entity, the database returns all the connex ENV leaves that are reachable by the mobile entities (see Figure V.I.4).

We have adopted a model of four points defining a planar and convex surface for an ENV's surface for three reasons:

* More than 4 points or non convex surface can create multiple possibilities for connection between two ENVs,

* If the surface is not planar, the path through ENV does not contain the variation of the surface. The path is a line between the entry and the exit points used and there is no rapid way to know all the points on the surface of the ENV,

* Finally, four points compared with three points is the best configuration for a smooth travel and space distribution. The centre of the surface is used for the evaluation of the weight of the edges of the graph, and more the surface of the parallelepiped is "rectangular" (minimizing of the difference between all the angles composing the surface) more the computed path is smooth.

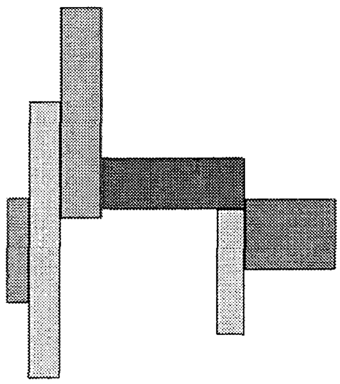


Figure V.I.4 *Example of various ENV connex geometrically*

To illustrate the steps of the path planning computation we starts with a set of useable ENVs by our mobile entity(Figure V.I.4). Each surface is used to extract 4 entry/exit points on the middle edge of the ENVs' borders. In our example we use only the points allowing to pass from one ENV to another as shown in the Figure V.I.5. Thus we obtain a first graph without weights. The weights can correspond to various criteria but for our example we just put the distance between each ENV. In order to find the shortest path in the graph, we have introduced a modified version of the Dijkstra algorithm (ref. [32] and ref. [79]) adding edge dependencies. We have defined some criteria and constraints in order to choose this algorithm: we want to find the best path (shortest, fastest); we can deal with several arcs between two ENVs; the distance inside an ENV depends on the entry and exit points creating a notion of dependency between arcs. The nodes are the ENV and the edges correspond to the points allowing passage from one ENV to another.

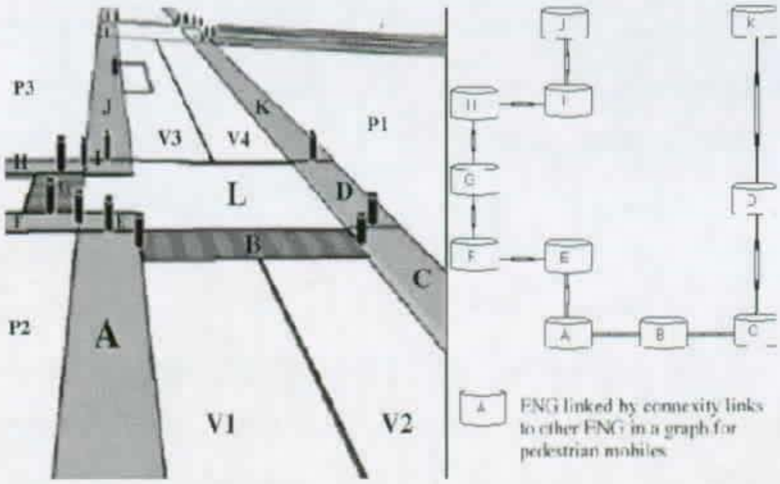


Figure V.I.5 *Example of ENVs inside a street with the entry/exit points and the graph associated*

The edges are associated with weights corresponding to a cost related to the distance, for example, between the entry and exit points of the ENV currently in use. Figure V.I.6 shows an example of two paths from S to F passing through various ENVs. The path Path2 is the shortest path, but if we look at the sub-path going to the ENV4, the shortest path is a part of Path3. The weight depends on the entry and exit points in the ENV4. Thus, we have a system with dependencies on edges.

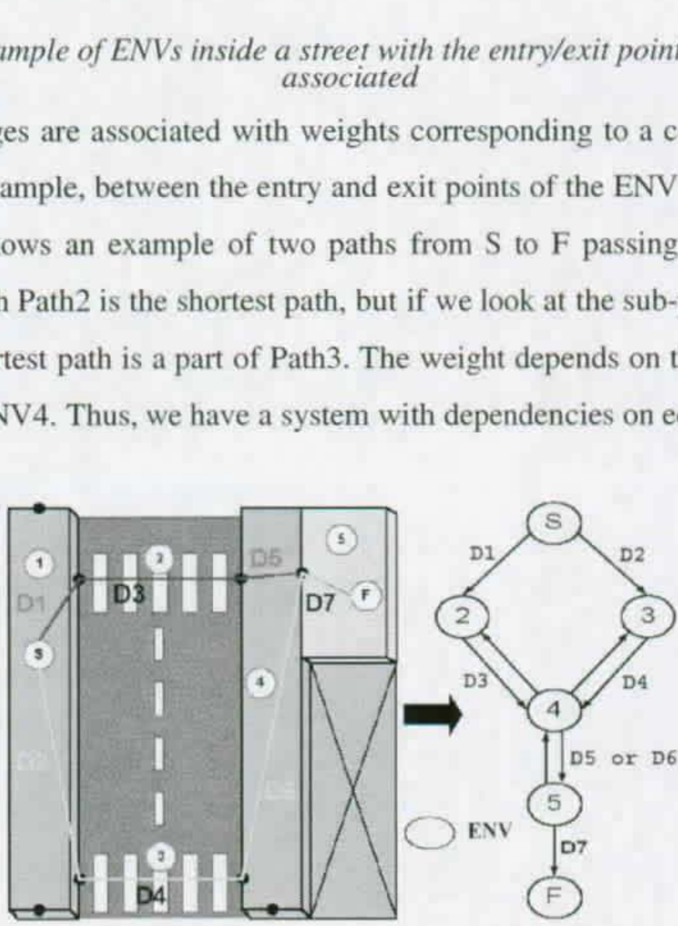


Figure V.I.6 *Path through a street showing edges dependencies*

We have a specific graph for each kind of mobile entities and some ENVs allow a connection between graphs. For example, say we have two graphs, one for pedestrians and one for buses: these graphs were both created in the same way as previously defined. In order to set a path for a virtual human in a park, using the shortest time option, path creation tool indicates taking the bus in order to quickly reach the supermarket. Bus stops are the links between graphs for pedestrians and buses. An interface has been created for selecting points in the city scene in order to define starting and ending points for path. It is also possible to add constraints for guiding the path through a specific place such as the park. These paths are used as pre-computed paths.

The environment paths are available to be used by the application managing humans such as a crowd module (S. R. Musse, ref. [67]). The path is a list of surfaces joined by points. The client using the environment paths creates its own path in this area in a way to avoid obstacles associated with the place and other humans. There are two ways to store a path structure. The first one corresponds to:

* the initial position (name of the point and (x,y,z) position),
* the current ENV where the start point is situated (name of the ENV),
* the exit point to get out of the ENV (its name),
* a list of name of ENV + exit point,
* the final position (name of the point and (x,y,z) position).

This is an example of computed path:

NUM_ENG: 6	**number of ENVs crossed**
3957.912354 40.000000 -12529.305664	**initial location of the mobile entity**
PAR_03_ZCI_Ca	**name of the entity where is located the start point**
point_468_PAR_03_ZCI_Vd	***name of the exit point***
PAR_03_ZCI_Vd	**name of the next entity**
point...	
PAR_03_ZCI_Ca	
point...	
PAR_03_ZCI_Ha	
point...	
RUE_V01b_TRd_013	
point...	
PIETON_A1_V01b	
point...	
RUE_V01b_TRg_09	**final entity of the path containing the arrival location**
77.579094 40.000000 -10676.068359	**final point of the path**

We stored pre-computed path inside the database in this format, and we associate some travel models to some high level entities (such as how to cross a town quarter) in order to easily reuse these paths. This declaration of path is very “light” but the “client” application needs the database in order to get all the complementary information such as the definition of the surface, the location of the point according to their names and the orientation to have at these locations. This full definition is another solution to define a path. It is used when some paths are read without the database as fixed information or during the simulation for guiding mobile entities. A module requiring “I am a pedestrian in the park and I want to go to the supermarket” would be provided with a complete path with all the surfaces to cross, all the objects to avoid collisions with and all the points to reach in term of (x,y,z) information. This is an example of computed path:

NUM_ENG: 6	number of ENVs crossed
POINT: AUTOMATIQUE	first point corresponding to the start point
3957.912354 40.000000 -12529.305664	initial locaation of the mobile entity
ENG: PAR_03_ZCI_Ca	name of the entity where is located the start point
5300.000000 40.000000 -11300.000000	surface of the entity
5300.000000 40.000000 -13900.000000	
2400.000000 40.000000 -11300.000000	
2400.000000 40.000000 -13900.000000	
0.424203 0.000000 -0.905567	orientation at the exit of the first entity
POINT: point_468_PAR_03_ZCI_Vd	name of the exit point
4600.000000 40.000000 -13900.000000	location of this point
ENG: PAR_03_ZCI_Vd	name of the next entity
4800.000000 40.000000 -13900.000000	surface of the entity
4800.000000 40.000000 -16500.000000	
4400.000000 40.000000 -13900.000000	
4400.000000 40.000000 -16500.000000	
...	
ENG: RUE_V01b_TRg_09	final entity of the path containing the arrival location
200.000000 40.000000 -10200.000000	
200.000000 40.000000 -16500.000000	
0.000000 40.000000 -10200.000000	
0.000000 40.000000 -16500.000000	
-1.000000 0.000000 0.000000	
POINT: AUTOMATIQUE	final point of the path
77.579094 40.000000 -10676.068359	

V.I.III.II Drawbacks and adopted solution

Using the points situated on the ENVs' borders as vertices for the graph of the path computing we can observe some jagged forms (see Figure V.I.7 and Figure V.I.8). The points used for the path computation are fixed and correspond to the location of the middle edge of the surfaces belonging to the path.

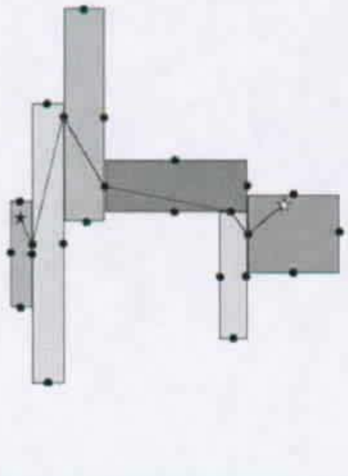


Figure V.I.7 *Path computed between the 2 stars using the points located in the middle edges of the ENVs surfaces.*

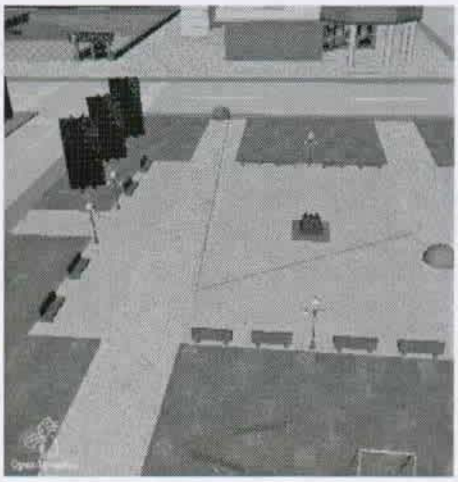


Figure V.I.8 *Path computation without optimization*

To avoid this incoherence we have improved the path computation by adding optimization.

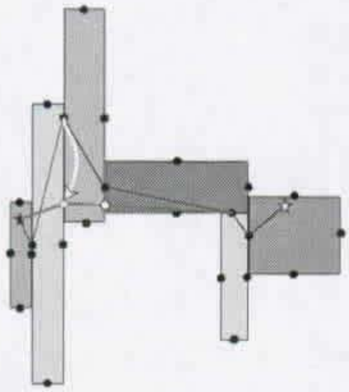


Figure V.I.9 *First try of optimization for the path computation, in light grey the new path with the displacement of the points along the edges of the ENVs*

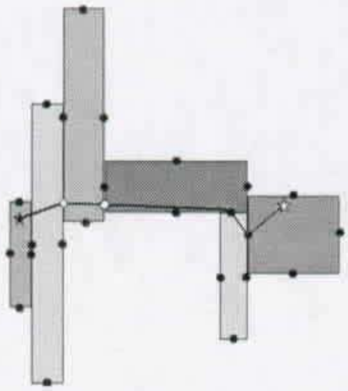


Figure V.I.10 *Final view of the optimized path computation between the 2 stars*

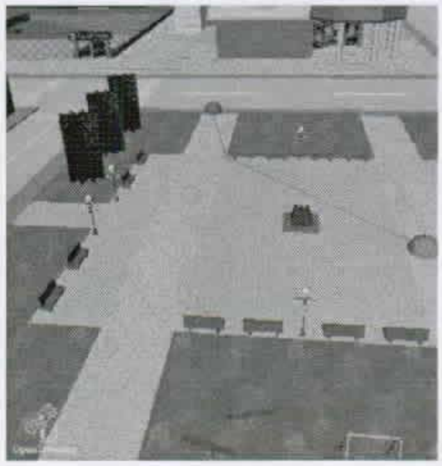


Figure V.I.11 *Path computed with optimization*

The first performed optimization consists in smoothing the path by analysing the path and for each point in the path, to compute a new location for the entry/exit point on the edges linking the two connex ENVs. We leave a minimal distance from the ENV border to the new point location. Figure V.I.7 and Figure V.I.8 show an example of path computation for a pedestrian in the park, and Figure V.I.9, Figure V.I.10 and Figure V.I.11 present the same path using optimization.

However, Figure V.I.7 and Figure V.I.10 illustrate the incoherence of such simplification, due to the use of entry/exit points. The optimization find the best path corresponding to the initial path choice. Nevertheless the distance associated is false (distance using the entry/exit points computed in the Figure V.I.7). A better solution should be applied such as the algorithm of Lozano Perez (ref. [50]), however due to the need of fast path computation, we cannot apply it to a large scene such as a city. In order to use the hierarchical decomposition associated with the pre-processing stage the algorithm of Lozano Perez can be locally applied (in the parcel ENV of the park for example). This algorithm decomposes a surface into a set of sub-surfaces with a minimum of size corresponding to the mobile entity's surface. This method is a solution for the creation of surfaces without objects inside for close environment such as a room or a parcel surface, but not for such large scene as a town quarter. According to the analyse of the complexity in the second section, the accuracy needed in order to route an humanoid in a realistic way would be 0.5*0.5 metre and this would create too much entities in the graph for path computation (complexity of the Dijkstra algorithm in NlogN, with N corresponding to the number of the entities inside the graph).

In order to enhance the path planning solution we try to analyse various possible configurations.

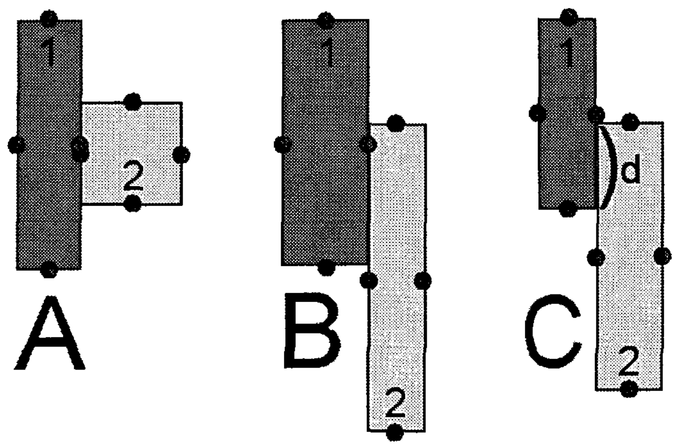


Figure V.I.12 *Various configurations of connection between two ENVs geometrically connex.*

We can have three different type of configurations for connections between two ENVs that can be used by a mobile entity. The first one, case A in FigureV.I.12, corresponds to a border of an ENV 1 equal or inside the border of the second ENV 2. The point of connection used inside the graph is the point belonging to the ENV 2 or the point of the ENV 1 if it is also inside the common edge of connection. The second one, case B in Figure V.I.12, is a segment not fully inserted inside both edges of the ENVs 1 and 2. If the segment is enough large to be used by the mobile entity (superior to the minimum size "dmin" allowing the mobile entity to pass thought it), the point used is the point of the ENV 1 inside this edge. The last case C corresponds to a common segment without common point coming from the ENVs 1 and 2 in the previous computation, the ENVs 1 and 2 are declared not connex.

Say that the common segment between ENVs is superior to the minimal common surfaces between two ENVs (d>dmin).

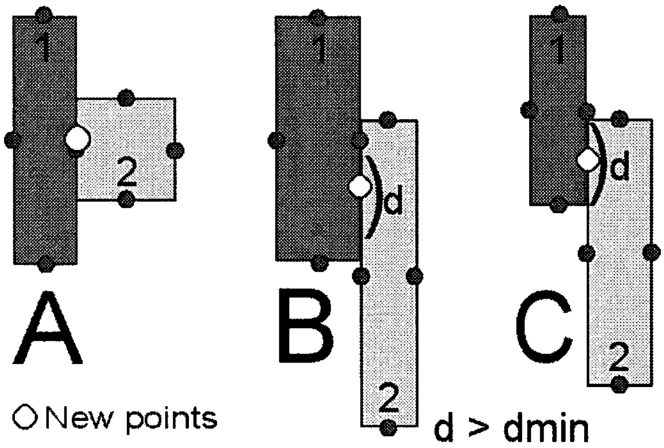


Figure V.I.13 *optimization of the entry/exit points localisation inside the common edge of two ENVs.*

A solution for jagged paths, is to update the location of the entry/exit points between the ENVs. New points are created in order to minimise the possible error when the path is computed and optimized. The configuration of the Figure V.I.12 can be changed as shown in FigureV.I.13 and the problem of the path planning optimization can be seen such as a set of possible positions for each point inside the common segment.

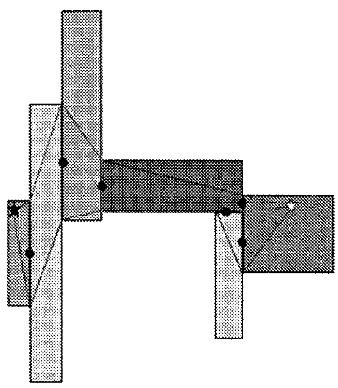


Figure V.I.14 *Representation of the possibility of points' position for a path computation optimized*

The darker surface inside the ENVs surfaces corresponds to the possibility space for each points, the best solution would provide an exact and a shortest distance between the two stars. To deduct such a result, we can segment the common edges into a finite set of points, with the weights of associated distance. This solution is not useable for our problem of large scene due to the explosion of both the number of entry/exit points and then the size of the graph. Another solution is to compute the best location of the points, keeping the initial graph and the initial weight and resolving the minimisation of an equation of n variables in 3D. The drawback of such solution is the computation time if a solution exists. We have adopted an intermediate solution, consisting in optimizing the entry/exit points localisation, minimizing the errors and optimizing the solution found in a last phase. The next draws illustrate our strategy, Figure V.I.15, shows the graph simplification in this example we assume that all the common edges are large enough to let the mobile entity to pass through.

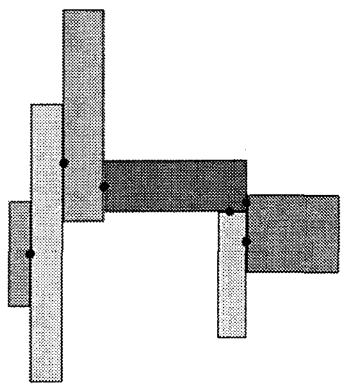


Figure V.I.15 *Representation of an optimization of errors on the Entry/Exit points location inside the common segment between two ENV*

The second phase is the path computation using the weights corresponding to the distance between the points (Figure V.I.16).

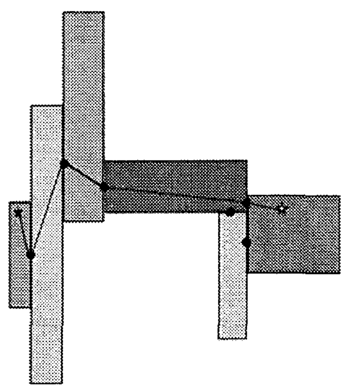


Figure V.I.16 *Path computation using optimized points location*

The optimization of the path consist in trying to join two separated points by another point without getting out of their ENVs surfaces; if it is possible, the line cutting their common edges segment corresponds to the new point location.

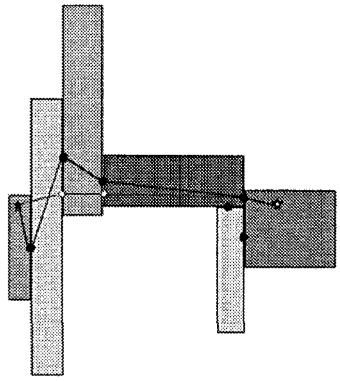


Figure V.I.17 *Path computation with optimization using optimized points location: view of the necessary displacement of the points*

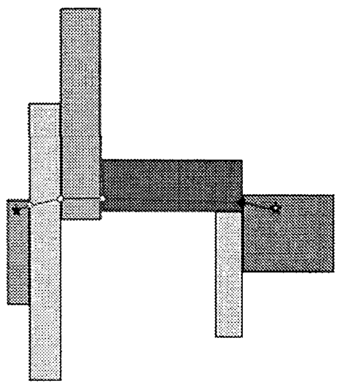


Figure V.I.18 *Path computation with optimization using optimized points locations*

The Figure V.I.17 and the Figure V.I.18 present this solution that is fast but has the drawback to provide a path with a false distance between points: the solution found is not the right one, according to the average of the error possible but the time of computation is optimal.

V.I.IV Object avoidance

Once a path is computed, the collision avoidance with objects can be performed via two different methods. First, the database provides the location of all the objects inside a surface when this surface is used by a mobile entity, thus the module handling the displacement of the virtual human deals with this list of objects. The second solution is to segment the surface according to the objects. At the initialisation stage, all the surfaces containing objects are segmented (Figure V.I.19).

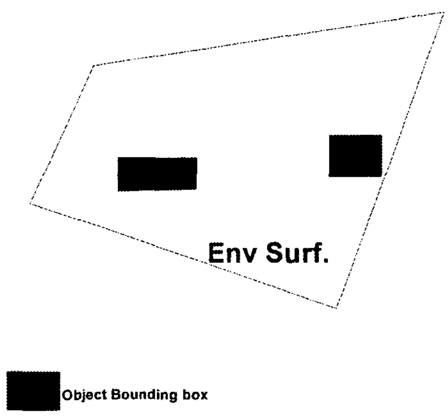


Figure V.I.19 *Surface of a ENV with objects inside the surface and represented by their bounding boxes*

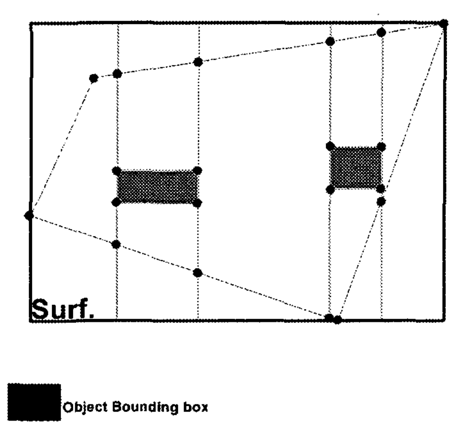


Figure V.I.20 *Computation of the segmentation of the surface ofan ENV, according to objects inside this surface*

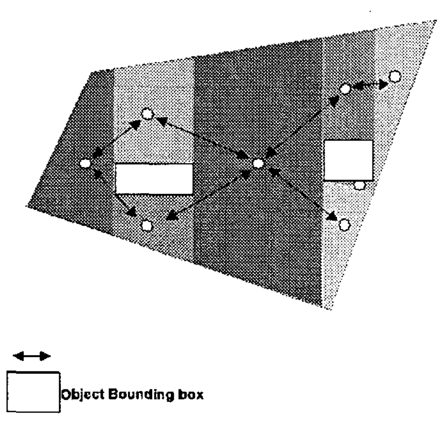


Figure V.I.21 *Segmentation of the surface according to the objects, creation of sub_entities and a sub-graph linked to the ENV taking care of constraint.*

The segmentation of a surface without a rectangular shape is done with the bounding box around the surface of the ENV, and the bounding boxes for the objects (Figure V.I.20). A set of sub_ENV are created and associated to the ENV containing all the objects. A sub_graph is computed and stored inside the memory (Figure V.I.21). This sub_graph is used once a path has been computed and uses this ENV. This sub_graph takes into account the constraints such as a minimal dimension for a common segment between two ENVs (see chapterV.V -).

V.I.V Some Results

We have performed some simulations using the path computing tool. These paths, crossing our scene, were pre-computed or computed in real time during simulation for various mobiles entities (pedestrians and buses). The next figures (Figure V.I.27, Figure V.I.28, Figure V.I.29) represent some snapshots made during these simulations, others research works involved in these simulations are a crowd module (S. Raupp Musse ref. [82]), a rule based behaviour, a smart object tool (M. Kallmann ref. [44]), a client-server for all the connections (N. Farenc and al., ref. [30], E. Schweiss and al., ref. [96]). S. R. Musse had developed the crowd simulation software. This software handles the displacement of virtual humans with a "crowd behaviour" according to the computed paths (points and surfaces), the crowd reaction to events such as the arriving of the bus, the crowd behaviour and the display. Most of the next pictures have been done during simulation inside this framework

Example of simulations using computed paths (Figure V.I.22 to Figure V.I.33). These paths are furnished as input for the crowd module (E. Schweiss and al., ref [96], S. Raupp Musse and al., ref. [82], N. Farenc and al., ref. [30]).

Figure V.I.22 *Pre-computed path inside the applicity application*

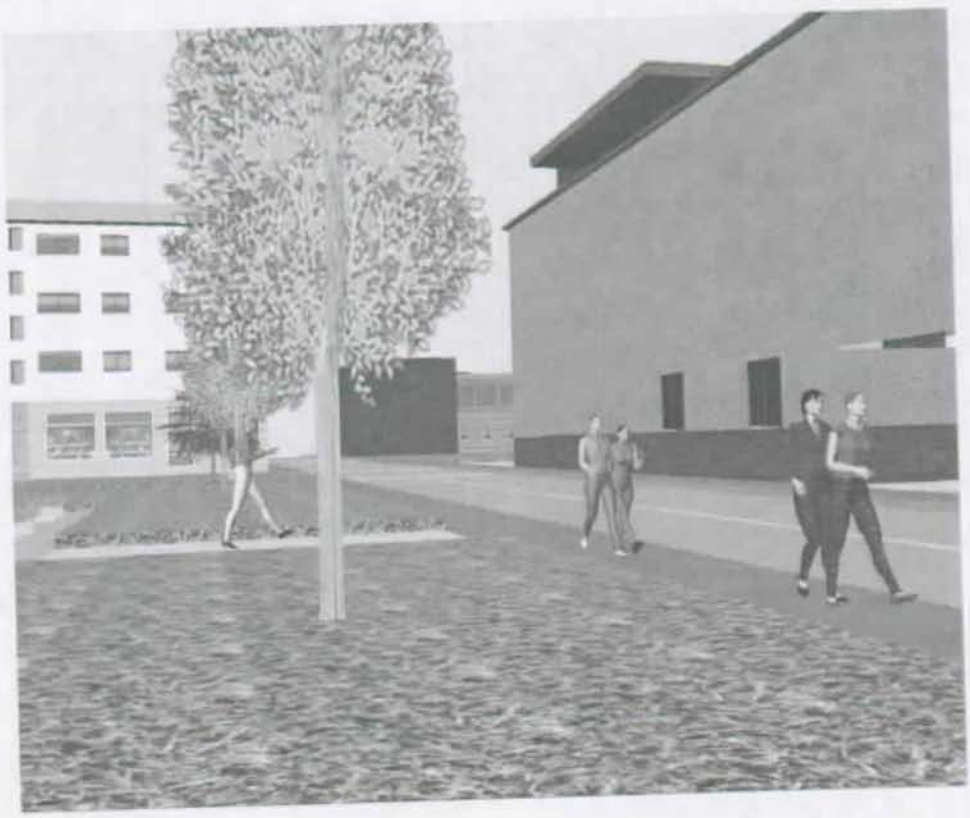


Figure V.I.23 *Group leaving the park*

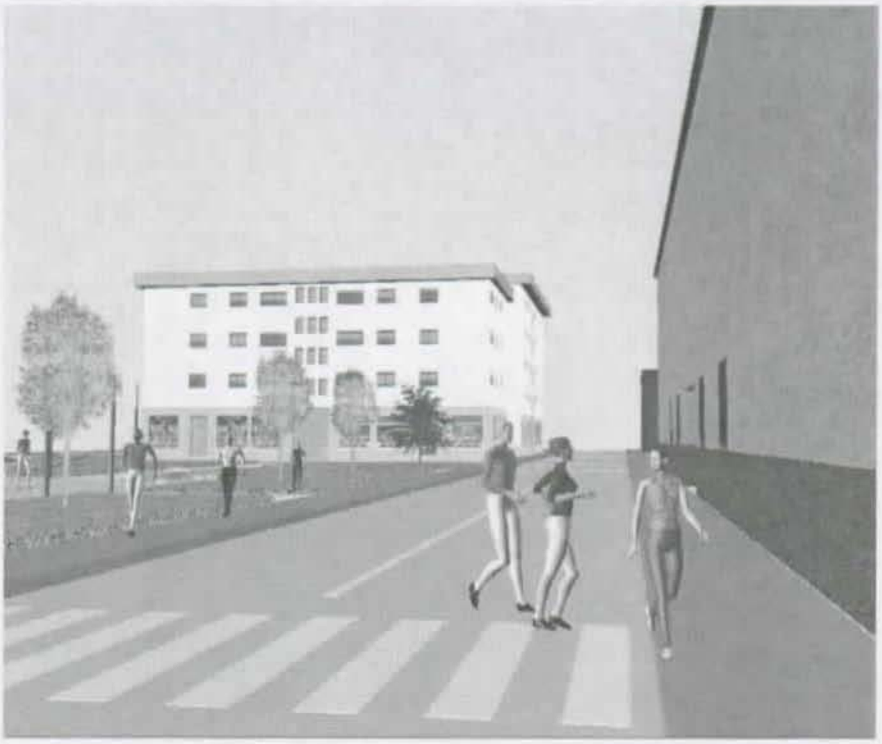


Figure V.I.24 *Group of crowd following a pre-computed path*

Figure V.I.25 *Group following a pre_computed path avoiding object collision.*

Figure V.I.26 *Group following a pre_computed path avoiding object collision.*

Figure V.I.27 *Top view of the inhabited city*

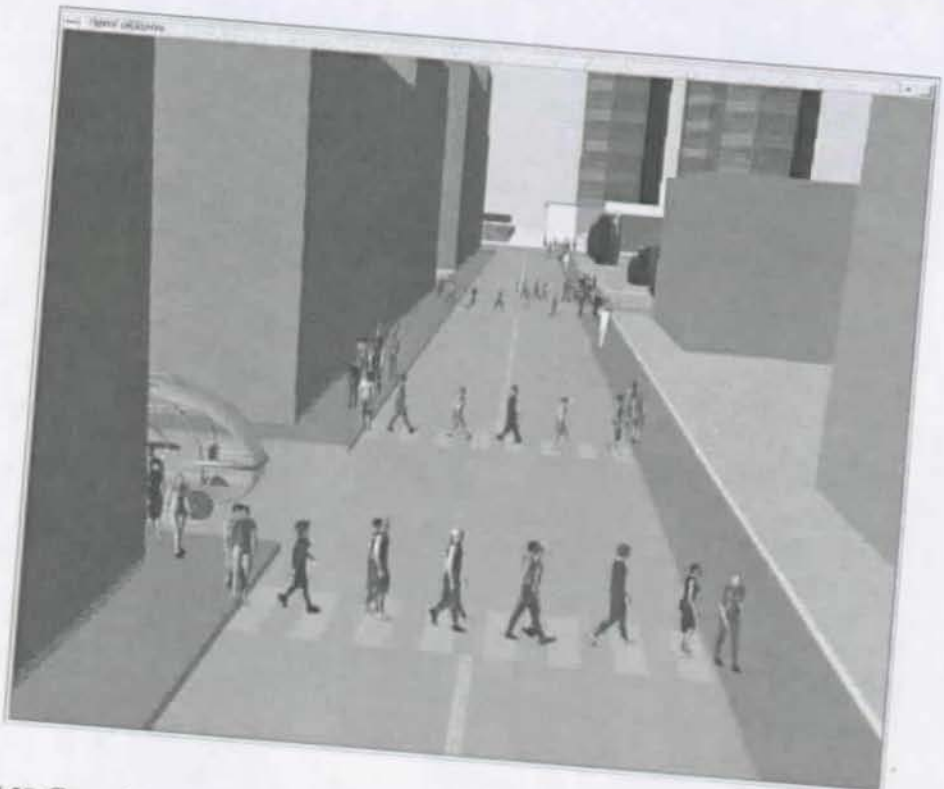


Figure V.I.28 *Crowd following pre-computed paths for pedestrians and a bus following a path for cars*

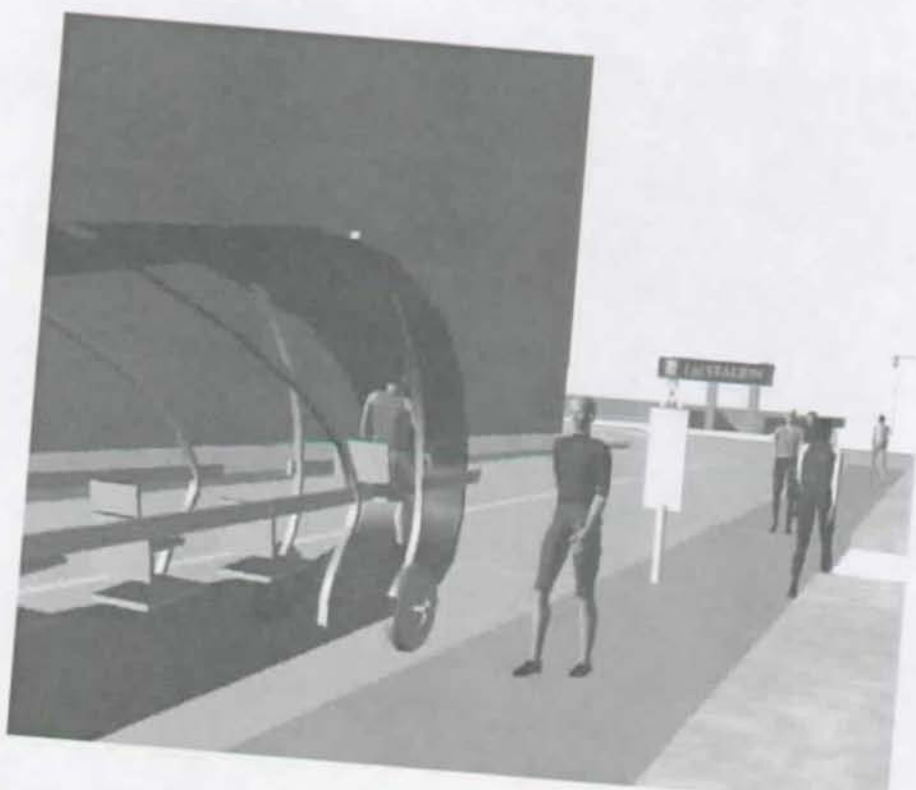


Figure V.I.29 *Bus waiting at the bus station and crowd following pre-computed paths*

Figure V.I.30 *Simulation of inhabited city with pre_computed paths and paths computed during the simulation*

Figure V.I.31 *Simulation of inhabited city with pre_computed paths and paths computed during the simulation*

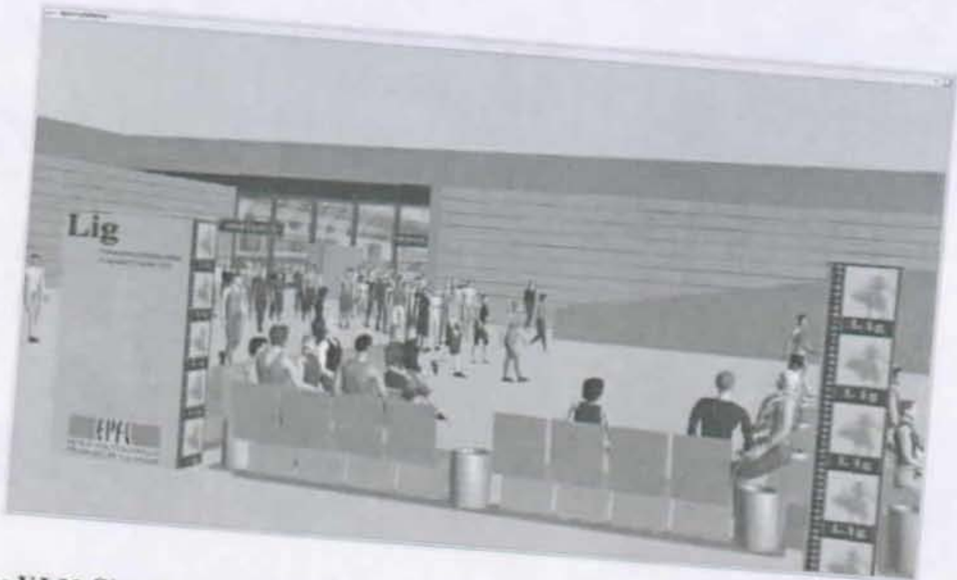


Figure V.I.32 *Simulation of crowd inside a building, the station, with pre-computed paths for the autonomous crowd and path on the fly for guided crowd (E. Schweiss and al, ref. [96])*

Figure V.I.33 *Simulation of crowd inside a building, the station, with pre-computed paths for the autonomous crowd and path on the fly for guided crowd (E. Schweiss and al, ref. [96])*

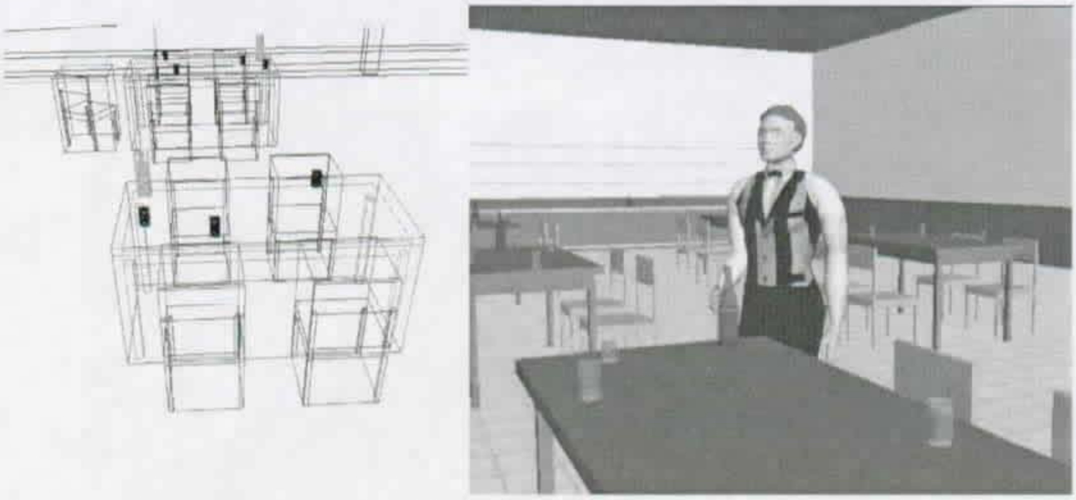


Figure V.I.34 *Interior of the museum, with a set of objects stored in the database (on the left) and some interactions using perception initialised querying the database (C. Bordeux ref. [12])*

Figure V.I.34 is a snapshot made by C. Bordeux, the database provides to his perception tool, the list of objects present in the scene and using the perception the virtual human is guided to an object moveable such as a bottle that it grasps.

V.II - Autonomous inhabited city

C. W. Reynolds (ref. [86]) presents a good definition of "autonomous agent" together with a presentation of a motion behaviour relying on various levels of behaviours from simple locomotion to strategy development. An extension of the Informed Environment is to provide data for simulation of autonomous inhabited city. For this purpose, the database can store other data types such as actions or behaviours. An example of action planning during simulation is the stairs example. An agent needs to follow a path to go from one place to another, and during his displacement, he has to climb stairs. To handle such situation the database associated to the scene needs to provide a way to inform the virtual human about this constraint and to provide a way to deal with it. Three solutions are possible. The first one is to define the stairs as smart objects and to let the stairs take the control of the virtual human for the climb. The two others solutions are data computed in pre-processing phase and stored in the database and more precisely in some ENVs on the top and at the bottom of the stairs. When the agent arrives near the stairs, the environment provides the agent with one key frame file for performing the movement to climb the stairs. The key frame file was pre-processed before simulation. The database provides a link between the ENV and the key frame file. During simulation, the ENV informs the agents of which key frame file to play for the ascension. Other solution is to have an information about the topography of the stairs (number of steps, size and height of the steps) and to modify the walking motor of the virtual human when it is arriving at the top or at the bottom of the stairs. For the case of a town we think that the third method is the most appropriated for such large scene, but this method as the drawback to add information to be stored by the designer via the naming of the objects stairs.

The database can be also used to simulate a random behaviour, via a list of basic action stored in the entities. The tool handling the virtual human can choose randomly the action for the virtual human from this list of actions. Another use of the database is to store complex behaviour in the ENV. For example, all the behaviour for taking a train can be store inside the train station building entities. The rule based behaviour system (A. Caicedo and al., ref. [17]), loads all the needed rules in order to perform behaviour computation adapted to the place where the agent is.

V.III - A tool for scene composition

For the time being the scene is created using 3DSMax, and converted to the Open Inventor format (as we are working at the geometric data level) equivalent to the VRML1 format. We have produced various databases: one for the city layout, and others for the parcels containing information. The city was created on the model of the decomposition rules and objects naming defined jointly by the designer (see chapter on the modelling phase) and database manager. The modelling of the city is made using a method of subdivision. According to the data model, a part of the city such as the block has its layout in the city defined as a set of streets, junctions and external definition of the parcels. According to this layout an associated database can be produced and used. On another hand, we define all the parcels as a part of the layout and specific databases are attached to each parcels definition. In this manner we have the park that can be considered as a database, a museum used for other simulations and some buildings with or without an associated database. To obtain a global database covering the layout and all the parcel contents, we have two possible methods:

* to create the database from one entire file scene composed of all the parcel presentations.

or

* to mix the scene and the associated databases adding or substituting specific parts of the scene.

The first method is to create the full scene, using all the labels present in the scene we extract all the ENVs and construct directly the global database. This method is more adapted for small scene, or scene not composed of sub-parts with existing associated databases.

The second method used the labels present in the 3D scene and in the databases.

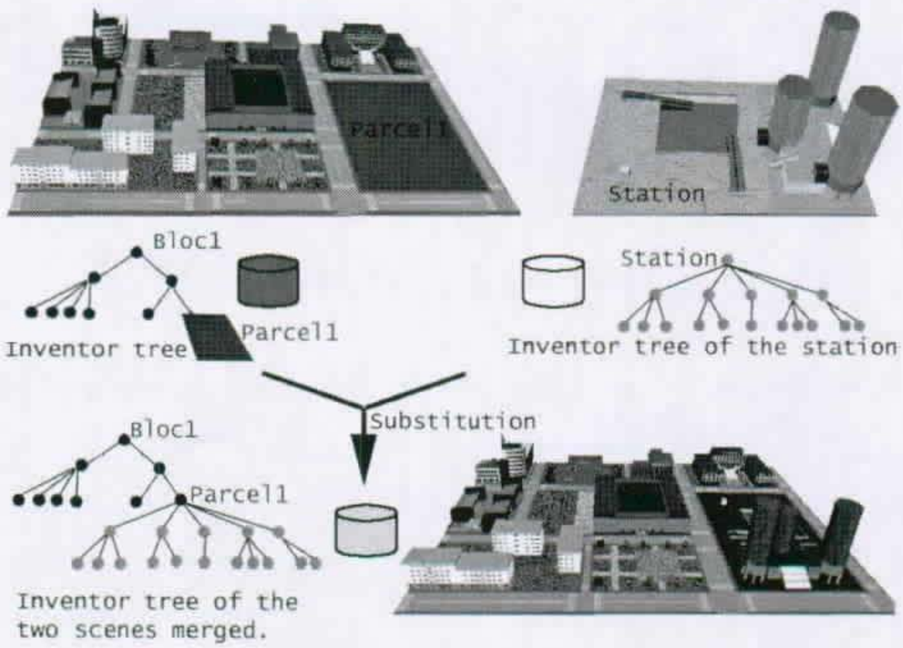


Figure V.III.1 *View of the Inventor hierarchy and a method to create new scene with database associated.*

Figure V.III.1 presents the method applied to construct the city. The layout of the city is used and each parcels corresponding to one file substitutes the simplify representations of the parcels. The databases of the city layout and all the parcels are mixed using the labels, for example the name of a parcel "Parcel1" present in the layout database and in the database to be added, "station". We make a substitution and add new data corresponding to the entity "station" in the main database under the entity "Parcel1". For the 3D model, the same method is applied. Normally, if the user can access to different parts of the scene with associated databases, the scene representation is segmented in a set of files corresponding these parts. The global 3D scene at the end of the construction is not very large due to the use of "inline" nodes for importing the various parts of the scene. For a block modelling, the initial file is the layout composed of files for each street segments and junctions. Then, for each parcels, an address file creates a link providing the 3D model of the parcels. Modifications are very simple (requirement to know the label of the parcel in the global model and the name of the new representation file) and the scene resulting is represented via simple files linked together. The scene construction can be done in the same time as database merging or it can be dissociated. In this case only the visual representation of the scene is created and the user has to verify that the associated database still stick to the visual scene. We can create an informed Environment for a scene with empty buildings, and when it is needed, replace the part of the database corresponding to the parcel containing the building by a new parcel with all the data attached to the interior of the building. For the scene of our block (Figure V.III.1), we have defined a database. Containing 313 ENVs, 552 in/out points and 42 objects, it includes several streets and junctions, a park, a supermarket, a house, a museum and a train station

In order to have a common set of scenes for simulations, we have also define a method to easily add furniture in the scene and inside the database. Using the location of the objects and the database, we associate the objects location with the ENVs' surface. Some constraint on the height of the object is used to avoid the computation of collision avoidance with lamp ceiling for example. The added scene containing the objects can be parsed once in order to create a small database associated to the scene. Then, only files corresponding to he 3D model can be used for object descriptions. By this way, a scene can easily be modified and the database easily updated.

This scene tool composition can be useful in order to have various levels of details for a part of the town. The user can try various representations for a parcel and store the names of the various files used with information for display optimization inside the database. These information can bring data concerning the weight of the file in number of triangle and textures. The user can also associate information such as the distance for switching from one level of detail to another one. The next chapter presents a method to optimize the display of the scene once the user has associated, via the tool of scene composition, various possibilities of display.

V.IV - Display optimization

V.IV.I Introduction

The field of level of detail associated with scene display optimization research is active and it is possible to find several interesting works generating various level of details for a polygonal model during a simulation. Some methods, like in games optimization, propose to mix 3D objects and warped images for distant parts of the scene. In this section we study the use of the knowledge informing the scene composition (through database queries) in order to optimize the scene display or to enlarge the viewed scene.

V.IV.II Global overview of our approach

Our approach is to have a set of static LODs for the scene and to select the most adapted ones according to the point of view. We adopt the strategy to minimise time computation for the scene display, even if a lot of pre-processing design has to be done. The database attached to the environment provides useful information in order to improve our LODs management for an urban scene. With the same notion of cell-visibility, we use the database and the connectivity of the ENVs with their automatically computed bounding boxes in order to facilitate the determination of the visible part of the scene. Thus, during a simulation with display optimization, we have two scenes, the first one corresponds to the scene observed with mesh and textures, the second one corresponds to a "symbolic" representation of the same scene via cubes representing the bounding boxes of each part of the scene. According with the viewed bounding boxes, the scene is updated with the good level of detail. In the same manner, the infinite model via scene part displacement can be implemented (see next chapter). For this purpose we need a set of rules for an efficient management of various level of detail for the scene parts, and we present the various techniques for display optimization always using the database and ENVs' information.

V.IV.III A Standard: LOD of VRML

The Level Of Detail technique is an interesting way for increasing scenes rendering. As scene with a a complex geometry takes long time to be rendered, the designer models various versions of the same scene with different levels of complexity: the Level of Detail technique (LOD). The LODs are associated with a range of distance and the "visualiser" tool switches between the models. Because further is the viewpoint, lower details can be displayed, then time rendering is optimized. This node is implemented in VRML and Open Inventor languages (ref. [118] and ref. [73]).

V.IV.III.I Some rules to link LOD and future database

We choose to define some rules in order to let the user easily find various LODs for a same part of the scene. In the directory dedicated to a specific object (parcel, building or telephone see Figure III.V.6) the designer creates a set of LODs for the object changing only the name of the file.

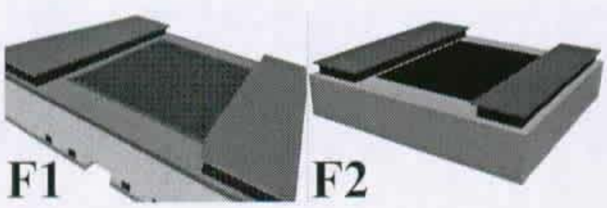


Figure V.IV.1 *Various low Level of Details for a museum*

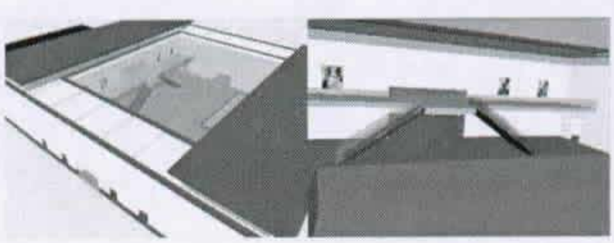


Figure V.IV.2 *Views of a high Level of Detail for a museum*

Figure V.IV.1 and Figure V.IV.2 present various LODs for a museum building. F2 correspond to three boxes with only material colour whereas F1 is a little more complex but without rooms, inside stairs and pictures on the walls as shown in Figure

V.IV.2. For an object named "myobj1.iv", the designer creates various versions using some extension about the refinement associated to the considerate version. For the object "myobj1.iv", there could be in the directory containing "myobj1.iv", "myobj1_F1.iv", "myobj1_F2.iv", "myobj1_F3.iv" (extension F for far) or "myobj1_N1.iv", "myobj1_N3.iv" (extension N for Near). The number indicates the degree of accuracy or simplification of the model. In a text comment at the top of the file, the designer informs the user about the scale of the scene (metre, centimetre or km) and also the ideal range for the distance for the camera. An LOD inline file could have this type of stored configuration in the database for a Building_station_N2.iv file:

```
{ # scale metre
  # distance 20 - 50
  # weight triangles 150000
  # weight texture memory 5 gb
  ...
}
```

Furthermore, ENVs of the database contain information about the files covering their surface, and we add the information concerning various level of details associated. Using the inlines present in the scene and in the database, we construct a simplified scene named Bdbscene corresponding to the inlines's computed bounding boxes. This simplified scene is stored in the database and is used to perform optimal computation of viewed inlines according to a camera location and configuration. Thus for a point of view, with set-ups camera, only using boxes, the optimization process computes the visible inlines and their associated distances. This is important in the LOD management for time consuming optimization. In case of embedded inlines such as a stair inside a building, the distance does not only determine the inline observed. The "semantic" location of the camera using the database (for example inside the building) can enhance the computation of visible inlines. We use the geometrical/semantic information for display optimization and LOD choice.To provide user with larger scene, we have explored others techniques of optimization.

V.IV.IV Trompe l'oeil technique and scene part displacement

Surrounding three-dimensional block representations are not necessary for simulation situated in the centre of a block. Two concepts can be used to enlarge the urban view without loading unnecessary objects: the first one relies on trompe l'oeil technique, simulating other surrounding blocks views, the second one consists in moving, during the simulation, a block situated behind the camera in front of the current block in order to give a sensation of infinite perception. Both are dependent on the geometrical decomposition of the urban model. We have defined two versions on which these techniques could be applied on: the ideal geometric decomposition and the generic one. We present these models and then the techniques.

V.IV.IV.I The ideal geometric decomposition,

The ideal geometric decomposition is made of nine cubic blocks forming a quarter and a block is composed of nine cubic parcels. With this ideal model there is always a central unit surrounded by eight others units. Even if this seems unrealistic, concrete examples exist in reality as in Manhattan centre with a ratio of 1/3 between width and length for all the parcels of this quarter

V.IV.IV.II The generic geometric decomposition

The most generic model of decomposition on which the techniques of enlarged view presented here can be applied is a n faces convex polygon. Having in mind an ideal/generic geometric model decomposition we present the techniques of trompe l'oeil and infinite model.

V.IV.IV.III Trompe l'oeil technique

The trompe l'oeil technique concerns simulations of a centralised view of neighbourhood ENVs, from inside an ENV such as a block. There are two methods to display pictures of surrounding blocks.

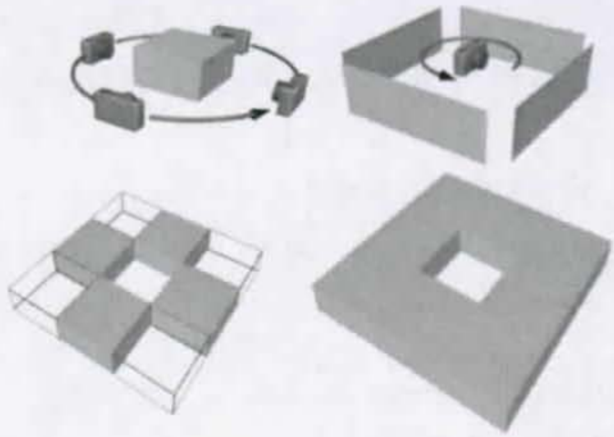


Figure V.IV.3 *Trompe l'oeil techniques*

The first one is to turn around the block, and to take perpendicular pictures of all the edges of the block. The cube (ideal decomposition) or a more complex polygon simulates the viewed block with these pictures mapped on the external faces. For a simulation all surrounding blocks are replaced by these lures. This can be a LOD for the block. The main drawback of this technique, regarding the ideal decomposition is the non-realistic effect created because the diagonally opposite blocks aren't seen and there is a huge feeling of being in a box. This impression is worst in a generic modelling (Figure V.IV.3 left part).

On other hand it is possible to compose a cubic panorama of the eight surrounding blocks (ideal model) from the simulation centre block, and wrap these cubic maps on the inside face of a global cube covering the simulation scene (FigureV.IV.3 right part). D. Aliaga presents an example of this kind technique in order to accelerate time rendering (ref. [2]). The difficulty of such a method is to load and visualise all the blocks (without the central one), and to compute the maps (due to inline visibility and space memory). Under 3DsMax (ref. [122]), if it is possible to view such a scene, a tool PANORAMA (ref. [119]) is accessible and generates automatically the maps and the cube (with the ideal model). For a generic model the problem is more complex and problem of non-perpendicular views mixed with various points of view can create visual artefacts.

V.IV.IV.IV Infinite model

This solution consists in moving parts previously loaded and hidden in front of the camera during a camera movement. The board effect when the user arrives near the end of the scene is removed, and the user doesn't have the feeling to be movement restricted in a box. The film "The Cube", presents this type of technique (ref. [121]).

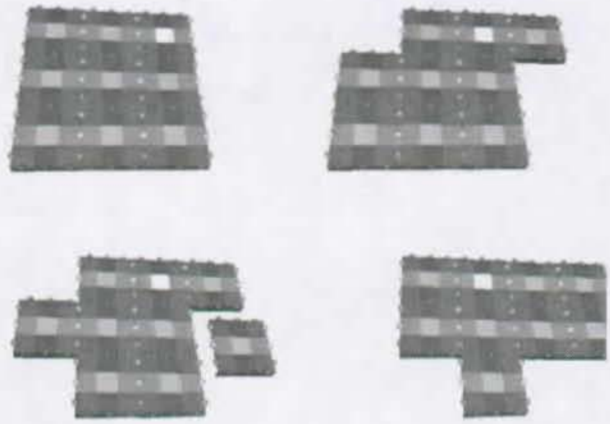


Figure V.IV.4 *Example of "infinite" model with scene part displacement*

The main drawback is the constraints emerging for the scene design. All board connections have to be planned in order to connect all the sides with each other. Moreover the virtual autonomous human has to be "transferred" and repositioned inside the scene when the scene is updated. The database also has to be updated at each scene part movement in order to assume data consistency.

The database is used for several achievements:

* Using the bounding boxes representing the ENVs, the application can easily determine what is situated in the field of view and at which distance.
* Some parts of the scene can be connected logically through information stored in the database. Thus, we can create one or several models for updating the model when the camera moves in the scene. The information specifies the type of link allowing the extension of an entity. Concretely, it can be the location of a street or a junction and, if we don't use an ideal model, the size of the ENV to stretch. The database has one solution of connection for each side of an ENV, but also can propose several options chosen according to some criteria defined by the user (randomly, part of the scene already seen...)
* The last use can be the displacement of the autonomous agents from one position in the ENV deleted to a new "correct" position inside the extension

V.IV.V Choice for the good representation for a part of the scene

The scenario composition depends on camera's location or simulation of user location. The user can perform some pre-simulation computation storing some camera locations and scene visualisation in the database. The user defines his/her camera and a surface where the camera is going to move. ENVs displayed associated to the camera movement are determined and stored in a list. The connective links between ENVs help to determine the visibility at the first level. The scene corresponding to the camera displacement surface is the sum of the entire part scene observed with the higher LOD computed. In the database we store the surface of displacement of the camera, its configuration and all the inlines to load with this scenario. If our camera is moving, passing from one area to another one, the scene can be or a list of camera scenarios or a total sum of the "inlines" linked to the camera displacement surface.

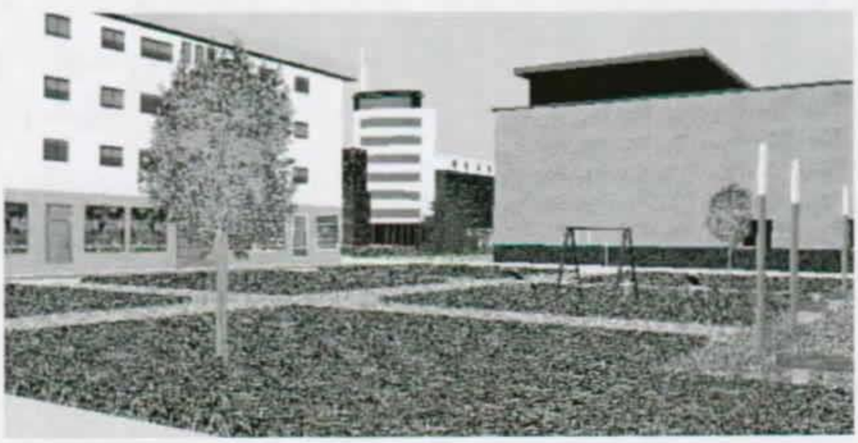
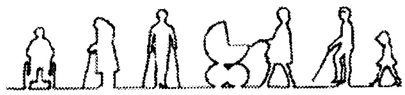


Figure V.IV.5 *Example of city composition with various levels of details, a low one for the museum building and a high one for the park composition*

This method is efficient if the camera does not cover a big area. Conversely, the constructed scene could be too huge and the best solution is to have various LOD handled with the "switch" nodes. Previous figure shows a picture taken from a scene with a low LOD for the museum (draw F1 in Figure 6) inside our virtual city.

V.V - A tool for handicap assistance



V.V.I Introduction

Disability is the result of a process initiated by an underlying disease, an accident or an abnormality, which leads to a functional deficit in various situations of everyday life. This definition derived from the International Classification of Impairments, Disabilities, and Handicaps (ICIDH), reflects the desire of the elderly and handicapped for social integration. It may also form the conceptual basis for analysis of the capacities of individuals and population. The number of people touched grows with two main factor: the ageing of the population and the various injury due to accident of car or sport practice. Handicap touches most of the population at various degrees. None said that everybody is a or a future handicapped people. Annex B we present an overview of the impact of the various criteria such as the height of steps.on the population' daily life. The VAHM project, an example of project in the field of assistance for handicapped people is also presented in this section. Our purpose in this field, is to perform path/action planning for entities with reduced mobility in a reconstructed scenes. The mobile entities can try to simulate, the constraint inherent to wheelchairs, pushchair i.e. These scenes could be reconstructed using different tools: via a modeller (such as VUEMS (S. Donikian, ref. [24])), or directly reconstructed by designers or automatically done via image reconstruction (Crosses project (ref. [123])). The model is used to test existing infrastructure making simulations and propositions for enhancements of the model.

Another approach is to help blind people to known their surrounding town quarter via simulation in a virtual reconstruction of the urban model, providing instructions and marks in order to not be lost and to be able to go from one place to another one.

Thus, with Josep Garriga, as part of his final project of diploma, we have analyses the constraints attached to such mobile entity and we have introduced these constraints inside the path planning tool. We have defined the possible use of the informed environment for blind people for local space learning but we did not have time to implement it.

V.V.II Evaluation of the need and the infrastructure adaptability

In order to root mobile entities with constraints such as rolling chairs, we try to define the major constraints attached to path planning adapted to such mobile entities. There are three major constraints in the case of a rolling chair path planning:

* The place must be adapted semantically to moving as defined previously for pedestrians, thus we link mobile entities with constraints with pedestrians mobile entities,

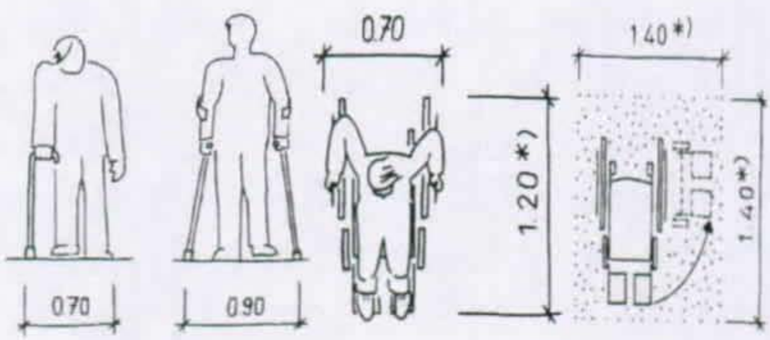


Figure V.V.1 *Example of official characteristics for reduce mobility definition(F. Falek, ref. [29]).*

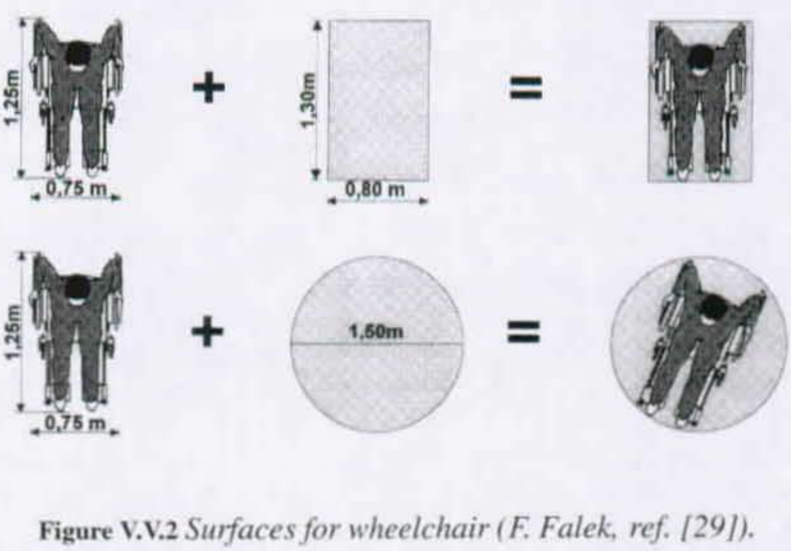


Figure V.V.2 *Surfaces for wheelchair (F. Falek, ref. [29]).*

* The surfaces must be large enough to allow displacement inside, for these constraints we used official documents related to architectural construction rules. For

example, a common rolling chair need a minimal larger of 0.7 meter (see Figure V.V.1 and Figure V.V.2).

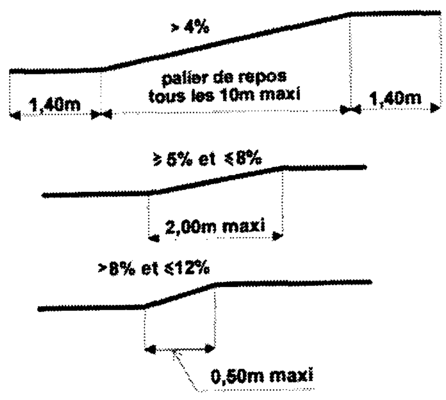


Figure V.V.3 *Norms for wheelchairs (F. Falek, ref. [29]).*

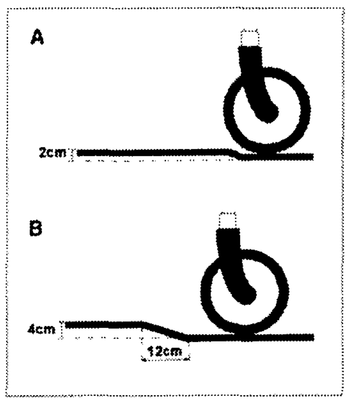


Figure V.V.4 *norms for stairs that a wheelchair can pass without exterior assistance (F. Falek, ref. [29]).*

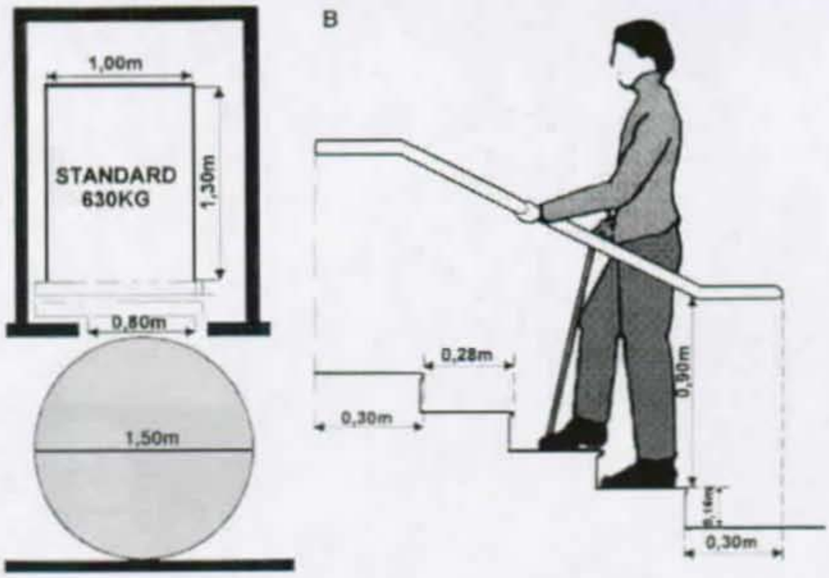


Figure V.V.5 *Stairs and lift conform (F. Falek, ref. [29]).*

* Slopes and stairs can be used inside a path, but they are constrained according to official documents related to architectural construction rules (Figure V.V.3, Figure V.V.4 and Figure V.V.5). In the case of a rolling chair we have a maximum of yy cm for a stair size and a slope of zz % for a surface. Another constraint for side-walks such as described in Figure V.V.6 shows that you can determine a conform width side-walk if the entity connex is a wall or grass. This information can be collected in the database regarding if there is a leaf entity directly connex with the side-walk: two useable geometrically continue, if not it means that there is a wall or a barrier between the two surfaces.

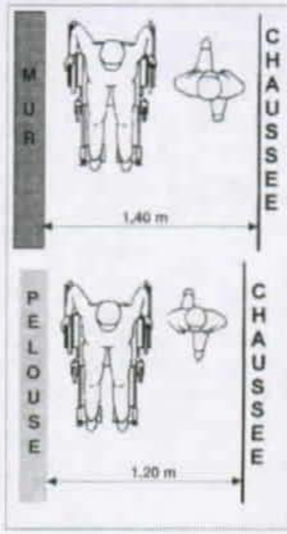


Figure V.V.6 *Example of constraints find on a conform side-walk width, depending on the type of surface connex with the side-walk (F. Falek, ref. [29]).*

The user defines some mobile entities and the constraints inherent to them. The path planning tool adapts its graph in order to take care of the new constraints added to the model.

V.V.II.I Segmentation of the entities according to the mobile entities' constraint

During a simulation of a virtual human displacement inside an entity, the data-base provides all the objects declared inside the surface. The application responsible of virtual human walking computes all the points to reach according to these objects and to other virtual human evolving in the scene. The main drawback of this method is to not have the possibility to determine if a point is reachable or not according to objects' locations in on foot surfaces. Moreover, the path distance estimation can be false if a very big object inside an entity crossing compels the virtual human to bypass it.

Another solution is to segment the entity according to the objects inside its surface. The entities are decomposed into a sub-set of entities according to the objects present in the surface. The surfaces too small to be used by the mobile entity we want to root, are no more known inside the graph and others replace the node corresponding to the initial entity. The segmentation is done according to:

the specifities of the mobile entities (constraints)

the objects present inside the entity.

The information attached to bounding boxes can be enhanced with high of the bounding box used to segment an entity's surface. A mobile entity with constraint such as a robot can modify the position of its joints in order to pass an obstacle or a small surface of displacement.

Once the segmentation is done, the new entities and their links are integrated in the graph connecting all the ENV. New entry/exit points are computed and also stored in the database.

V.V.II.II Path planning with constraint

For a specific mobile entity such as a rolling chair, we refine the graph used for path planning. All the surfaces not enough large or too sloping are extracted from the graph and the connections via entry/exit points are observed in order to evaluate the difference of altitude between two surfaces in order to see if the stair created is correct or to high for the mobile entity. Then for each specific mobile entity with constraint the generic graph of pedestrians is refined and is used after for path planning and shortest path between two points. All the data attached to an entity is available and, in the case of actions, the tool responsible of decision making chooses what to do according to the information it has about the mobile entity it is handling. We have a graph for each mobile entity type according to its constraints.

The database tools, according to the mobile entity to route, perform path computation in order to find solution to these questions such as the shortest path. The mobile entity (objects with mobility such as pedestrians, cars, buses, and bicycles) type (pedestrian, car or buses) uses only surfaces recognised as surfaces usable by it and entry/exit points creating the underlaying graph for path planning. For car or buses we don't have constraints because the designer ensures that a rolling way in a street is larger enough for a car. For pedestrian and handicapped people the problems consist in evaluating surface of displacement free (with no obstacle) in order to determine if the surface is larger enough for displacement inside.

To compute a path, all the ENV are analysed and for all the entities that can be used by a same mobile entity we computed entry/exit points. These points are attached to ENV and depend to the mobile entity able to use them. We construct a graph connecting all the connex entities through their entry/exit points. The weights of the edges correspond to the distance inside the entity between an entry point and an exit point. Some sub-graphs, corresponding to the entity decomposition according to the objects present inside the surface, are computed and stored inside the entity. The path planning includes these sub-graphs when the entity is crossed. With the constraints for mobile entity with reduced mobility, we remove all the edges not corresponding to the constraints.

V.V.III Spatial learning and urban infrastructure analyse

This utilisation of the database can help for decision making concerning urban infrastructure especially for creation/modification of spaces adapted to handicapped people. The list of the object preventing people to access one place can be observed and the objects not at the good place such as a mail box in the centre of a side walk, suppressing some edges can be displaced. This tool can show these objects and the edges not accessible for entities with reduced mobility. Furthermore, this tool can help to learn the surrounding environment for blind people given some computed path in intelligible command.

During simulation of virtual human, at a first step we can command movement using low level order such as "join the position (x1, y1, z1)". The entire path is decomposed into a set of planar surface with points at the border. For a virtual human a path corresponds to a list of points to reach. As we route virtual humans we access to information about the location and orientation of the virtual human body. Higher order can be sent via command in natural language (A. Caicedo, ref. [17]), but also for path entertainment, we can convert the command concerning global location in relative location command. For example, whereas reaching the location (x1, y1, z1) we compute a relative location according to the location of the virtual human and the command becomes "go on two foot steps, turn on your left at 2pm hour." Thus a path normally described via a set of point's no understandable for a user became intelligible and the user can practice the path and remember it. In the case of visual diseases, a good improvement is a vocal explanation of the path. In order to help the user to locate itself, the database when the path pass near an object informs the user of the proximity of the object giving its name (if it is stored in the database) and their relative location. M. A. Anwar (ref. [4]) proposes an on-board system with GPS system for real outside simulation, this could be a nice improvement of the system of place learning. The tool can be improved by adding the list of objects associated to some landmarks such as objects in the path in order to help the spatial positioning during a simulation. Some specific landmarks exist like ground with "bubulles" (experiment in the La Villette Museum ref. [117]) for blind people.

V.VI - CROSSES Project

The CROSSES project is an example of the use of the informed environment in a real reconstructed scene.

V.VI.I Overview of the CROSSES project

CROSSES is an european project (CROSSES IST-1999-10510), "CROwd Simulation System for Emergency Situations" where the informed Environment is used for knowledge about the geometry of the scene and the semantic associated. According to the global presentation of CROSSES (ref. [123]) here is an overview of the project and the informed Environment implication.

This project deals with Simulation and Training and intends to develop advanced simulation system for outdoor urban emergency situations. The three dimension environment will be constructed using real imagery (aerial and on the ground) as well as virtual models. Acoustic simulation from sound samples and sound modelling as well as graphic representation of people and crowd will guarantee a detailed realistic description of the situation. Specific developments on crowd representation, crowd behaviour modelling and control connected to acoustic corresponding effects will complete this dynamic simulation and training system. The foreseen application so far, concerns the training through advanced simulation to emergency situations such as an incident on a Chemical Plant in a suburb, with rehearsing of an evacuation plan, or validation of any safety procedure.

The objectives of this project is to provide Virtual Reality tools for training people to efficiently respond to urban emergency situations, given the need to be prepared and trained to these emergency situations to limit the side effects due to inappropriate behaviour and plans. We intend to demonstrate the use of a simulator to get prepared to these situations by recreating an actual area with high degree of realism (static 3D environment with dynamic smart objects, dynamic crowd and sounds) so that efficient and useful training to dynamic situations and scenarios can be performed (through global and immersive interaction).

Given a realistic simulation scenario Istar proceeds to the real City modelling (or the construction of the 3 dimension static environment) from aerial views of the site.

Figure V.VI.1 *Aerial Images, DEM & True Orthoimages copyright ISTAR*

The acquisition and collection of real high resolution imagery (Figure V.VI.1) allows to build a whole suburb with a high level of realism. Then a few buildings as well as smart virtual objects (doors, windows) are extracted very precisely from higher resolution aerial colour images and digital video acquisitions on the ground. Definition and modelling of typical crowd behaviour in specific situations, and translation into virtual crowd behaviour with a sociological approach will allow to demonstrate virtual crowd behaviour representative of the actual crowd behaviour observable on similar real situations.The actual definition and set up of behaviour for these smart object and crowd would allow to play simulation scenarios either dealing only with virtual crowd, or dealing with virtual crowd interacting with avatars corresponding to actual trainee.

V.VI.II The Informed Environment inside the CROSSES project

The Informed Environment is used as a library and like a client for the crowd module. From a 3D scene in VRML format from the IPI/ISTAR partners (Figure V.VI.2), we construct the database containing all the areas normally used for the circulation, and a set of other surfaces used during a panic event such as grass, flower bed etc. The Informed Environment sends to the crowd module, some informations such as how to go from one point to another one, the location of objects, the location of an agent, a new path according to the location of barriers put by the trainer/trainees during the simulation, the dimension of the environment and the dimension of the entities, some aleatory paths consisting just in reaching some exit point in the surfaces.

Figure V.VI.2 *Grangemouth stadium view with segmented surfaces used for the database creation (image from IPI and Aerial Images, DEM & True Orthoimages copyright ISTAR)*

The adaptations of the Informed Environment tool in order to handle the inherent constraints of this project concern various points. Some tools are going to be developed in order to complete the database set of requests. A panic event modifies some entity types into walkable surfaces. We perform some tests with triangular ENV corresponding to the data coming from the reconstructed 3D model when quadrilateral surfaces are not possible. The path planning is enhanced for various behaviour coming from panic feelings (random path planning) or constraint by toxic plume evolution. Other type of object affect also the path computation such as the placement of barriers by the fireman, cutting some edges in the graph, or the variation of the weight of the edges in the graph simulating the perception of dangerous areas by the autonomous agents. The database contents has to be modified by these objects and constraints during the simulation. The graph and sub-graphs are updated according to the new objects. The visual tools for data verification purpose are very important and the generic model used by the partners responsible for the 3D modelling is shared by other applications.

The first step (end July 2001) uses our town, with a the database already tested for behavioural simulation and tools integration. The next step (end September 2001) concerns the use of the Grangemouth stadium (VRML file (draw on the Figure V.VI.2)), with only quadrilateral entities, then the chemical Chemical Plant model with some triangles, with the database creation and some adaptation to new definitions for the ENVs (triangles, possibility to modify the graph for the path planning computation, association of complementary data for fire emergency simulation)

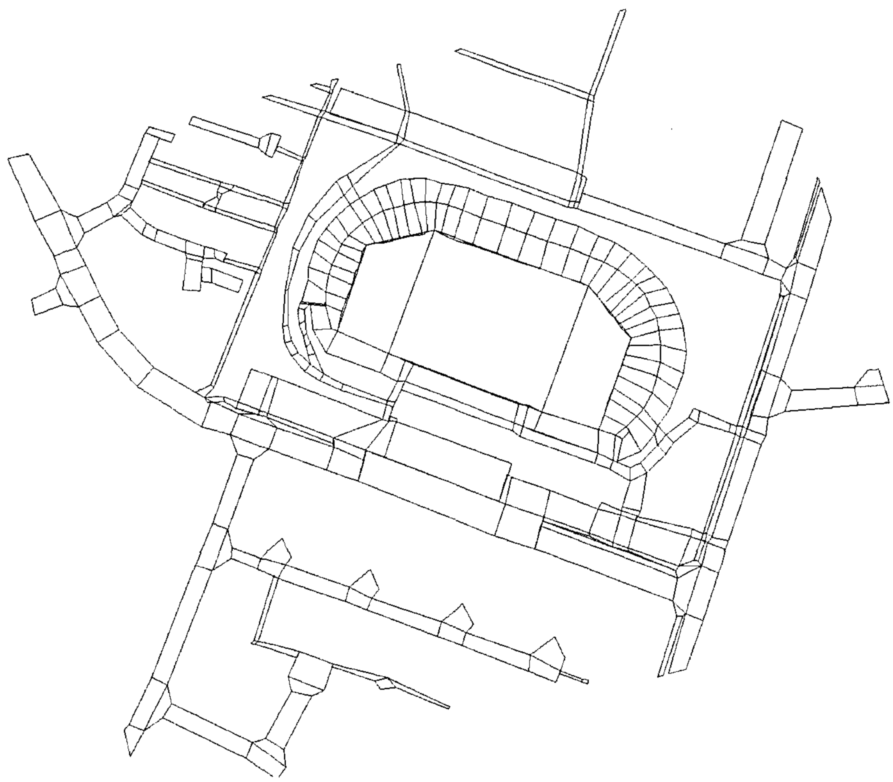


Figure V.VI.3 *View of the Database representing the stadium of Grangemouth*

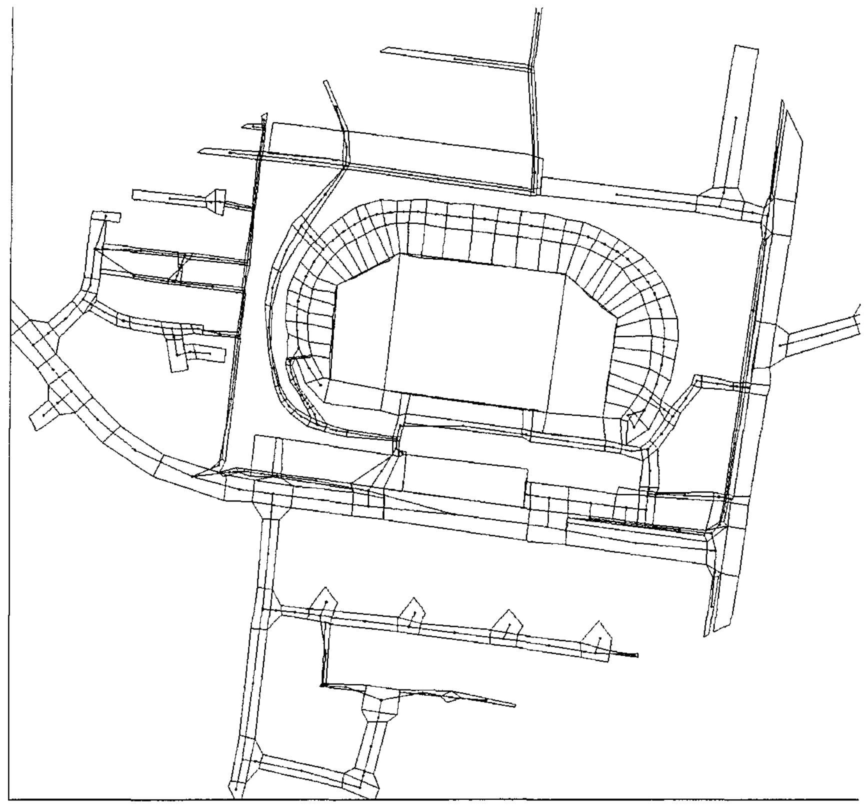


Figure V.VI.4 *Picture of the database created from the stadium entities showing the graph connecting the entities before an accident event*

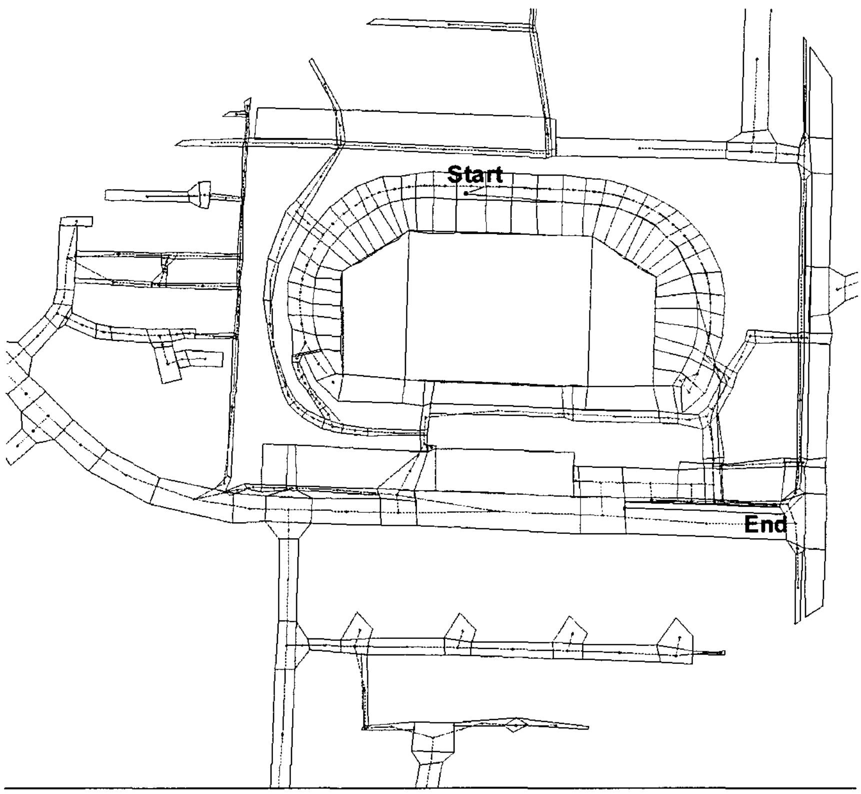


Figure V.VI.5 *Computation of a path inside the stadium scene before an accident*

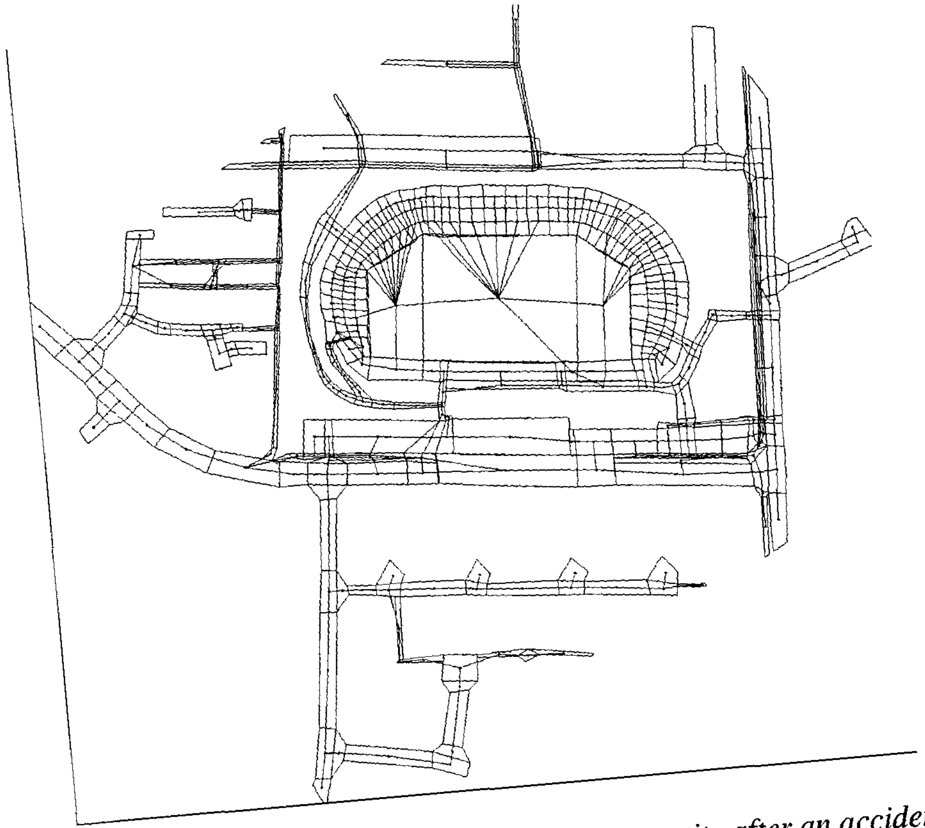


Figure V.VI.6 *View of the graph for pedestrian mobile entity after an accident event*

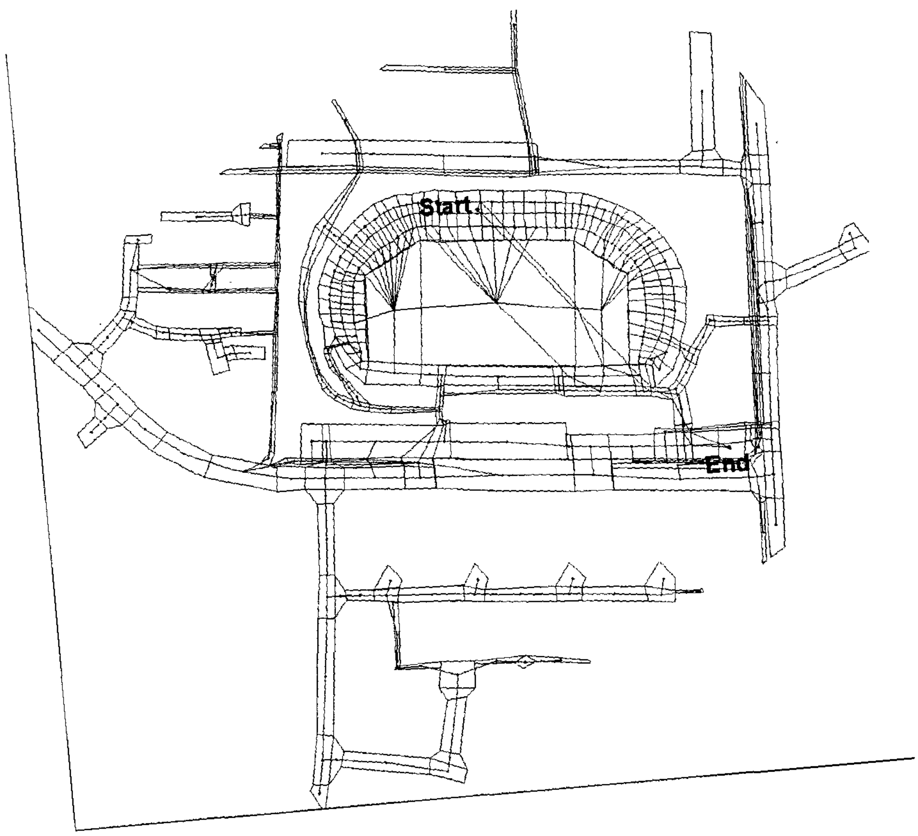


Figure V.VI.7 *Path computation after an accident event*

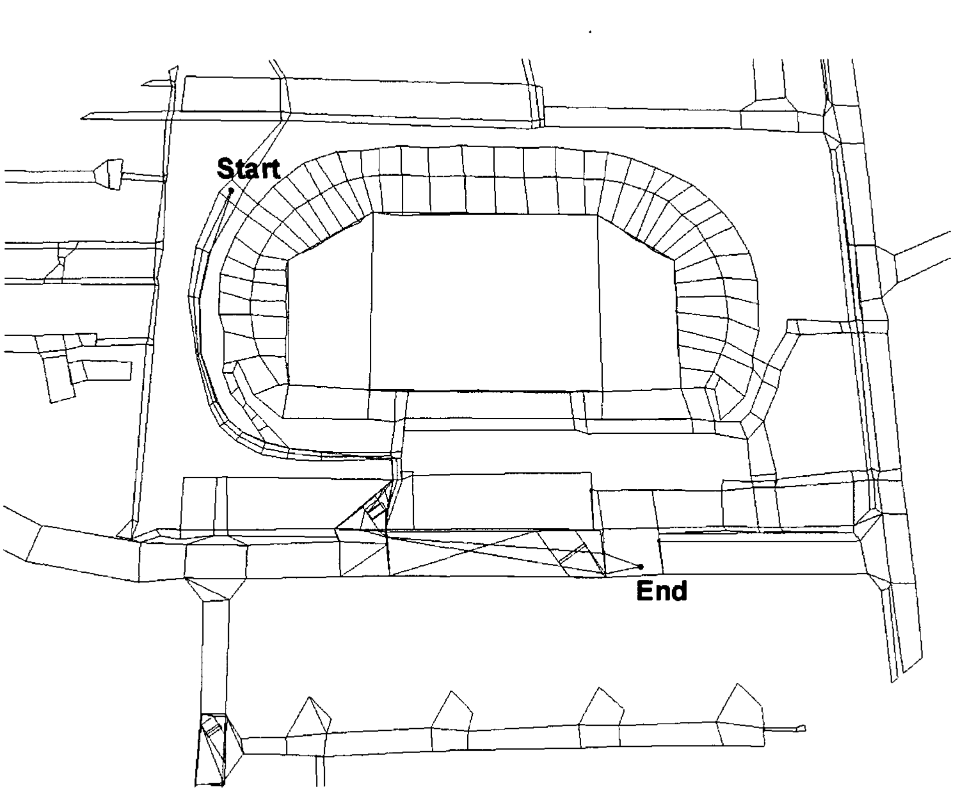


Figure V.VI.8 *View of a path dealing with object collision avoidance*

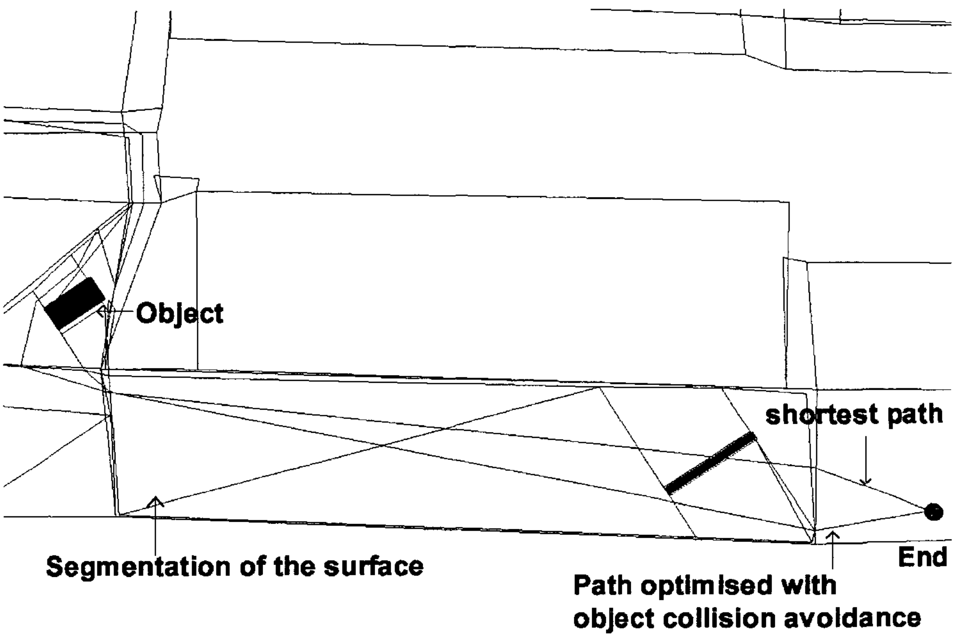


Figure V.VI.9 *Detailed view of a path computed optimised and with object collision avoidance*

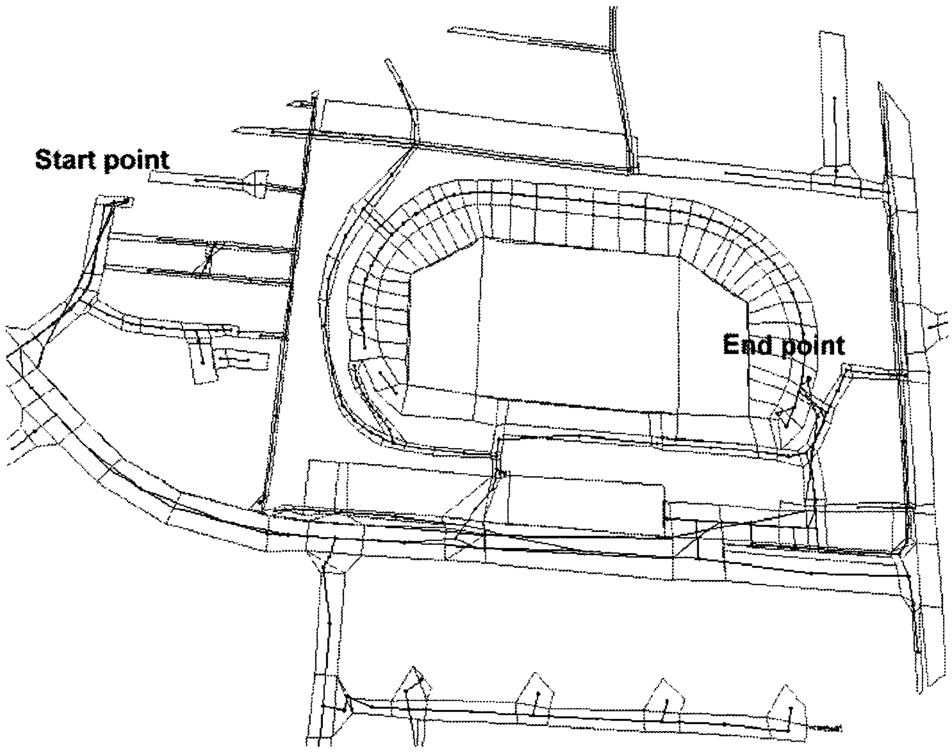


Figure V.VI.10 *Path computation with quadrilaterals and triangular entities*

Figure V.VI.3, Figure V.VI.4, Figure V.VI.5 and Figure V.VI.10 show, via the applicity tool, some drawing representing the database contents with all the entities and the graph linking the entities, a path planning computation using both quadrilaterals entities and triangles. The database representing the Grangemouth stadium scene contents 312 entities and using only circular areas (without entities for panic simulation in case of accident) we create a graph of 418 nodes. For the underlying graph used for path computation, we have put a constraint of 0.8 meter on the entities sides, that explains some disconnection between entities because the common segment between two entities is too narrow. As the CROSSES project is for simulation of accident, the event accident modifies, inside the database, some types for entities normally not walkable (in our model). Then some surfaces are useable after an accident such the grass of the stadium. Figure V.VI.6 and Figure V.VI.7 represent some views of the new graph and the new computed path after an accident event. The paths are computed without objects inside the surfaces given as output. Thus, the surfaces are segmented according to the objects located inside the surfaces used for the path, and this path is also optimised (FigureV.VI.8 and FigureV.VI.9).

VI - Conclusion

The main goal of this doctoral work concerns action planning in a large scale scene such as a town. This action planning in an urban environment allow the simulation of autonomous virtual humans. In order to achieve such human behavioural simulation, we decide to focus on the definition of a scene decomposition allowing the creation of a database. The first stage of this works consisted in defining how to construct a database simulating virtual perceptions (contact with the ground, object perception etc.)and what we called an "urban knowledge". These perceptions, necessary for autonomy simulation but highly costly in time processor consuming, are simulated through database access corresponding to our Informed Environment. We decide to define a 3D modelling of a part of a virtual city, an ideal model for test bed, integrating both geometrical and semantics data. For this purpose, we define rules for the design, storage and display options. These rules also include objects decomposition model that composes the database for virtual life simulation. The definition of "urban knowledge" and simulation of perception are the base of a generic decomposition model and are applied to our scene modelling. This application creates the data associated to a generic model through the use of town modeller. This generic model define in a first time to correspond to our requirements for autonomy simulation in our ideal town, can be easily adapted to new simulations. Such modification implies modifications in the modelling rules at the stage of the design (decomposition of the scene in new type of objects, new naming constraints). The domain of application for autonomy simulation with virtual perceptions and knowledge is unlimited. The main drawback is the difficulty to create and associate information to a model or to a database. The CROSSES project is a good example, the segmentation of a global mesh according to semantic and physical constraint is a real challenge, and the problem of scale precision between the virtual human constraints and aerial data reconstruction precision add new difficulties. Most of the time, human intervention for associating knowledge and semantic to a scene is unavoidable.

The modification of the generic model is also constraint by the application manipulating the database and providing data according to "client" requests. The application must have the internal structure adapted (fields) and procedures able to respond to high level requirements. Nevertheless, the actual application let the user compose and mix different databases with various degree of detail for the generic model used. The textual format for the definition of the generic model and for the database storage allows a large independency and the library in C treating these data let to the user all the possibilities for the choice of its development platform. We have decided such orientation because this application, once the database is structured and tested, must be very light in time processor consuming and independent of the visualisation file format.

We have defined some tools for the database visualisation, verification an modification. These tools can be enhanced in order to be more "smart", offering assistance to the user. The association of semantic and knowledge to object can be done with more or less human intervention and a lot of research are currently performed in this domain. Data visibility and verification is also a field of research and 3D data with attached information are difficult to observe.

Interesting applications with real life utilisation uses existing scene reconstructed in a virtual format. These applications for emergency situation studies or handicapped aids can have an important impact on the population. Discussing with people with reduced mobility, it appears that a simple analysis of urban infrastructure, in the streets for example would certainly simplify their daily life.

VII - Appendix A : Human body modelling

VII.I - Body Representation

The human modelling used in this work and in our laboratory, is briefly presented in this section, L. Porcher Nedel provided a complete overview of the human modelling state of the art (ref. [75]). The human body representation is based upon a multi-layer approach (R. Boulic and al., ref. [14]). The lower one corresponds to an articulated skeleton hierarchy with the joints and their degrees of freedom. This skeleton contains the position and orientation of the joints relative to their parent joints in the articulated structure. The volume entity approximates the body volume by a set of volumic primitives holding a fraction of the body mass. The second layer is composed of volume primitives designed to simulate the gross behaviour of bones, fat and muscles. These primitives fall into two categories: blendable volumes that will blend with other blendable volumes in the same group; unblendable volumes that will not blend with other primitives. The volume primitives, arranged in an anatomically-based approximation, are attached to the proximal joints of the skeleton. The third layer is the skin surface of the body that is automatically derived from the position of the shape of the first and second layers. An implicit surface defined by volume primitives is sampled by ray-casting on semi-regular cylindrical grids. These sample points are used directly as cubic B-spline control points to smooth out the skin stitched together to connect different parts of the human body for final rendering and output. Figure VII.I.1 shows the various stages in the construction of a layered human body model (J. Shen, ref. [97]).

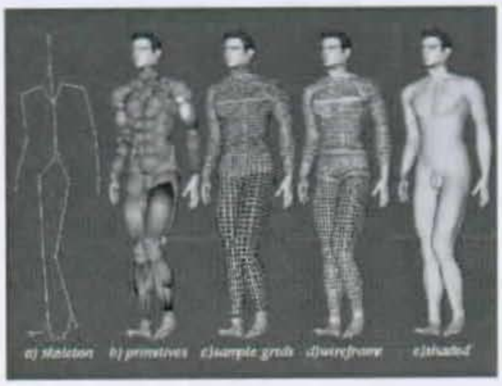


Figure VII.I.1 *Different layers representing the body construction (J. Shen, ref. [97])*

A finer modelling integrates some muscles reconstructed or only muscles influencing quite a lot on the external appearance of the body. These muscles correspond to deformable muscles using mechanical laws of particles, volume control in processing time, linear springs, action lines with points to attach the muscles to the bones (L. Porcher Nedel and al.,ref. [74]).

VII.II - Body Animation

The skeleton and the body skin can be scaled, morphed, for more realistic simulations, facial and hand movement can be a manifestation of verbal or non verbal behaviour, with collision detection and grasping of objects. The movement of the body can be performed applying key frames coming from motion capture or a mathematical interpolation of the movement of the joints simulating the walking model (R. Boulic and al., ref. [13]).

Figure VII.II.1 *Equilibrium on one hand for a virtual human using inverse kinematics approach (R. Boulic and al., ref. [14])*

Another approach is inverse kinematics applied to the articulated structure of the skeleton, it permits to compute one solution for a starting and final positions taking into account some constraints due to the environment or internal joint limit values (Figure VII.II.1 , R. Boulic and al., ref. [14] and P. Baerlocher and al., ref. [8]).

Figure VII.II.2 *Smart object example: interaction of a virtual human with a computer(M. Kallmann and al., ref. [44])*

Interaction with objects can use different techniques such as inverse kinematics, but the objects (the **smart objects**) inform the agents about how to interact with it, providing a key frame or key position for a displacement of the full or a part of the body (M. Kallmann and al., Figure VII.II.2 and ref. [44]).

VIII - Appendix B : Handicap field Overview

VIII.I - Impact of Handicap on the population

An interesting study, a micro-survey, has been held in a French village with an homogeneous community of 532 persons aged 1-92 years with a rate of 94.7% of participation. Made in order to evaluate in the daily life the major constraints and to estimate of the handicap of a population (P. Minaire and al., ref. [60]), this micro-survey's objectives were to obtain a reliable functional representation of the population of the village, comparing self-assessment of functional capacity with observed performance, and analysing the effects of age on capacity. The deterioration in performance observed was proportional to age and can be perceived to begin early at about 30-40 years. The configuration of the tests were on height of steps, moving ground and global motion, prehension, hearing and communication, prehension and visceral functions. Furthermore, a questionnaire was submitted to the individuals in order to evaluate by themselves their difficulties and handicap.The next schemes represent various results obtained and show the importance of the handicap or difficulties encountered by the population for the daily life.

It appears in this study that some norms are necessary for a better life of the population such as a standard size for steps around 15 and 20 cm and a cross hand allows 10% of the population to use higher steps. Also, people are more consciousness of their limits for big difficulties or current situation or various parts of the body implicated in the movements.

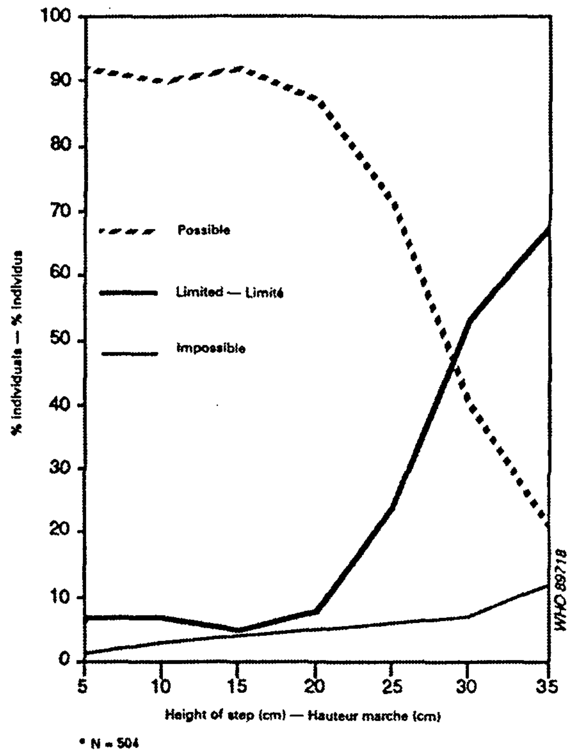


Figure VIII.I.1 *Link between descending three steps without support, height of step and the% of people having limitation with these values* (P. Minaire and al., *ref. [60]).*

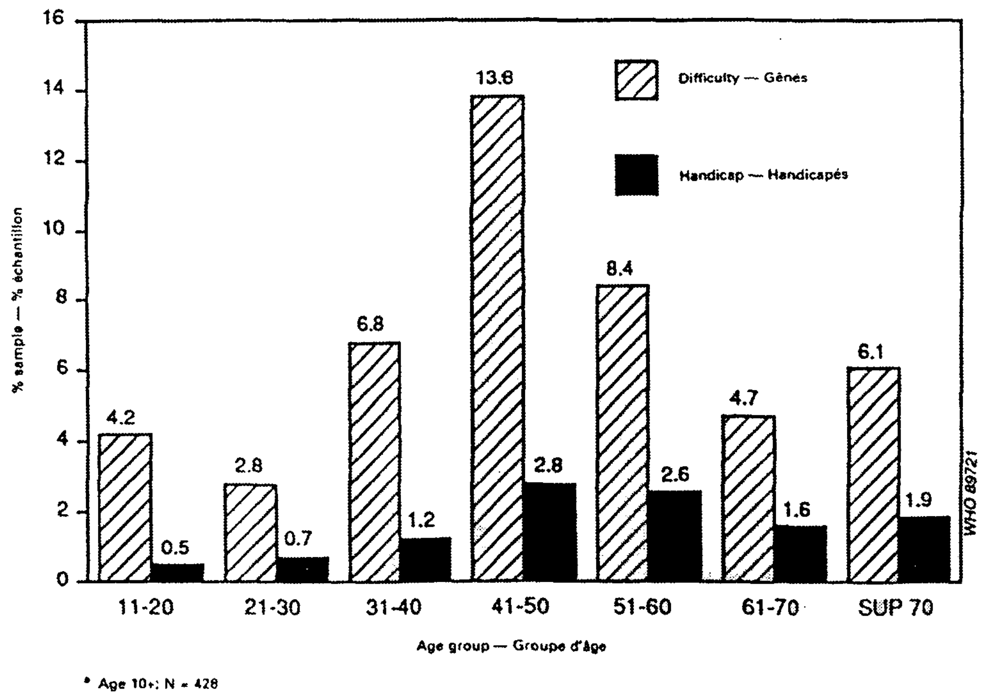
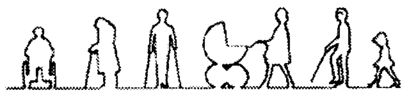


Figure VIII.I.2 *Difficulty and handicap by age group(P. Minaire and al., ref. [60]).*

This approach of the problem of current life difficulties create some new constraint in the architectural domain and the accessibility of various parts of the town, public places or interiors of private home, we talk about ***accessibility***.



Accessibility definition (F. Falek and al., ref. [29]):

"Accessibility makes parts of the general rules of construction such as security, soundproofing and insulation. The dimensional aspects which characterise it has been defined on the base of a occupied wheelchair. It is a template done in order to improve fittings suitable for all users' functional desires. This template, internationally recognised, determines the volumes to have priority for the architectural conception of buildings (rotation area, outskirts area, circular width, etc.)as well as adaptations to manage for using equipment (reaching and gripping area, elevation of a seating posture etc.). All these aspects are defined in term of technical rules defined within the regulation framework. The effective implementation allows all of us:

* to move without collide with obstacles created by designers and others professional of building,

* to access to all buildings

* to use all services dedicated to the public."

More and more towns and collectivities are sensitive to this orientation and tries to adapt their infrastructure to these needs. The museum of "La Villette" in Paris is a very good example of tests and integration of people evolving with reduced mobility in a large public environment (ref. [117]). They have analysed the needs for adults and children in terms of accessibility and make continuous innovations in order to try to update the criteria of assistance in helping accessibility. Another field of assistance is the notion of "robot" associated with wheelchair motors. The robot provided "view, decision making and eventually path planning to help and minimize the constraints and efforts due to the handicap". The VAHM project is one example of these progress in this domain.

VIII.II - VAHM Project

A lot of research concern the field of assistance for handicapped people. The wheelchair are robots with sensors with various level of improvement from vision to collision detection. We have picked out an interesting project VAHM (A. Pruski, ref. [80] & ref. [81]) which proposes an assistance to handicapped people during the use of wheelchair. Based on a system of cooperation between a robot handling various level of autonomies for difficult task, this smart wheelchairs tends to offer to the user more autonomy for person deeply affected or a feeling of comfort and security for ones less handicapped.

Their underlaying model corresponds to four layers plus the human level and based on the fundamental notion of cooperation man-machine. The lower level is the physical level and handles 14 ultra captors and one odometer providing the relative speed of the two tractor wheels. The superior level is a set of standard primitives driven or by the machine or by the user, for example they consist in navigation, wall following, obstacle avoidance and location computation inside a free space. These primitives use the data collected by the captors. The third level corresponds to the "smart level" and analyse the data coming from captors combined with a knowledge database about the environment. The model defines the dimension and the localisation of the elements present in the environment. The data correspond to a set of rectangular polygons (transformed into a topological model) and can be defined by an exterior person or by the user. If the data correspond to the visual input of the user, the model is interpreted as an approximate localisation and updated during the wheelchair evolution according to some rules such as the errors dependant of the distance of the observer from the object described. A map is filled and objects and surfaces are put inside. The path planning is done using the A* method and take in account the sensors data and the knowledge (topological model) coming from the database (with coherency update). The supervisor level treats the high levels decisions according to the coherency of the incoming information, it runs or shuts down orders, such as update a point to reach or to propose another solution like giving back the decision to the user via the human level. The level human or provides new data in order to upgrade the information stored in the database or decide what to do like taking the control manually or to stop all movements.

IX - References

[1] - G. Accardo, "3D Laser Scanning numerical surface definition of fast prototyping of the "basel-head bronze head", Eurographics'99, short paper, Milan Italy, Sept. 99, pp131-135.

[2] - D.G. Aliaga, A. Lastra, "Automatic Image Placement to provide a Garantee Frame Rate", *Computer Graphic Proceedings,* SIGGRAPH'99.

[3] - A. Antonicelli, G. Sciscio, R. Rosicarelli, G. Ausiello, T. Catarci and M. Ferranini, "Exploiting Pompei culturage Heritage: the Plinius project", *Eurographics'99*, Sept. 1999, short paper pp. 100-103.

[4] - M. A. Anwar, T. Yoshida, "OORF: an object-oriented Route Finder", SAC'2000, March 200, Como, Italy, pp 301-306.

[5] - M. A. Arbib,"Interaction of Multiple Representations of Spaces in the Brain", Brain and Space, J. Paillard, Editor.Oxford University Press: Oxford. pp380-403.

[6] - O. Aune, "Definition d'un environnement semantique virtuel pour des acteurs autonomes", *Internal Report LIG EPFL*, December 1998, 32 pages.

[7] - E. Asphaug, "The Small Planets", Scientific American, pp28-37, May 2000.

[8] - P. Baerlocher, R. Boulic, "Task-Priority Formulation for the Kinematic Control of Highly Redundant Articulated Structures", *IROS'98*, Canada 1998, pp 323-329.

[9] - S.S.T. Bandi, "Discrete objects space methods for computer animation", PHD report N. 1802 Ecole Polytechnique Federale de Lausanne, 1998.

[10] - Y. Banjou, H. Takakura, Y. Kambayashi, F. Xerox, "Visual User Interface for Spatial Databases Combined wih Landscape Images and Sensors", in proceedings of the International Symposium on Database Applications in Non-Traditional Environments, DANTE'99, pp 235-244.

[11] - M. Billinghurst, J. Savage, P. Oppenheimer, C. Edmond, "The Expert Surgical Assistant: An Intelligent Virtual Environment with Multimodal Input", In Weghorst, S., Sieberg, H.B. and Morgan, K.S. (Eds.) *Proceedings of Medicine Meets Virtual Reality IV*, pp. 590-607.

[12] - C. Bordeux, R. Boulic and D. Thalmann,"An efficient and Flexible Perception Pipeline for Autonomous Agents", *Eurographics'99*, Milan Sept. 1999.

[13] - R. Boulic, D. Thalmann and N. Magnenat-Thalmann, "A global human walking model with real time kinematic personification", *The Visual Computer, 6(6)*, December 1990.

[14] - R. Boulic, T. Capin, Z. Huang, P. Kalra, B. Lintermann, N. Magnenat-Thalmann, L. Moccozet, T. Molet, I. Pandzic, K. Saar, A. Schmitt, J. Shen and D. Thalmann, "The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters", *Eurogaphics'95*, Computer Graphics Forum, n. 14, V.3, pp337-348, Maastricht, Holland, 1995.

[15] - R. W. Bukowski, "Developing Rational, Performance-based Fire Safety Requirements in Model Building Codes", Internal Report.

[16] - R. W. Bukowski, "Interactive Simulation of Fire in Virtual Building Environment", *Proc of Siggraph'97*, Los Angeles, 1997, pp35-44 .

[17] - A. Caicedo, D. Thalmann , "Virtual Humanoids: Let them be autonomous without losing control". The Fourth International Conference on Computer Graphics and Artificial Intelligence. Limoges, France. May 3 - 4, 2000.

[18] - T. Catarci, G. Santucci, M. F. Costabile, I. Cruz, "Foundations of the DARE System for Drawing Adequate Representations", DANTE'99, Tokyo, Japan 1999, pp 53-62,

[19] - C. Claramunt, M. Mainguenaud, "A Spatial Data Model for Navigation Knowledge", *7th International Symposium on Spatial Data Handling*, Kraak, M. J. Molenaar, M. eds, Taylor and Francis, Delft, The Netherlands, 1996.

[20] - J. Cohen, A.Varshney, D. Manocha, G. Turk, H.Weber, P. Agarwal, F. Wright, "Simplification Envelopes", Computer Graphics Proceedings, New Orleans, 1996, pp119-128.

[21] - J. Cohen, M. Olano, D. Manocha, "Appearance-Preserving Simplification", Proceedings of SIGGRAPH'98, in Computer Graphics, Anunual Conference Series, Orlando, Florida, July, 1998.

[22] - B. Cramblitt,"Start your Engines", *Computer Graphics World*, Jan 1999, pp40-45.

[23] - M. S. Dahir,"Boot the Computer, Strike Up the Band", *Computer Graphics World*, Jan 1999, pp 15-16.

[24] - S. Donikian, VUEMS : "A virtual Urban Environment Modeling System "; *Computer Animation'97* pp.127-133.

[25] - P. Doyle and B. Hayes_Roth, "Agents in Annotated Worlds"; Report No KSL 97_0 Knowledge Systems Laboratory Stanford University California 94305.

[26] - P. Doyle and B. Hayes_Roth, "Agents in Annotated Worlds", Proc. of the Autonous Agents'98,p173-180, Minneapolis May 1998.

[27] - B. Eberhardt, H. Gürcay, F. Hanish, T. Hüttner, O. Licht, B. Nill, "Books and Devices from Old -their Rennaissance in Computer Graphics-", Eurographics'99 short papers, p167-174

[28] - C. Ezzel and all, "The Business of the Human Genome", Scientific American, p38-57, July 2000.

[29] - F. Falek, P. Dubois,"Accessibilité des bâtiments et de l'Envionnement", *Déficience Motrices et Handicaps*, Ed. Association des Paralysés de France, 1996, Vuibert, ISBN 2 7117 5236 4, pp 445- 459.

[30] - N. Farenc, S. R. Musse, E. Schweiss, M. Kallmann, O. Aune, R. Boulic, and D. Thalmann, "A paradigm for controlling Virtual Humans in Urban Environment Simulation", to appear in *Applied Artificial Intelligent Journal*, Special Issue on Intelligent Virtual environment, 2000.

[31] - N. Farenc, R. Boulic and D. Thalmann, "An Informed Environment dedicated to the simulation of virtual humans in a urban context", ", *Eurographics'99*, Milan Sept. 1999 ppC309-317.

[32] - G. Francon, "Creation et analyse de graphes pour une ville virtuelle", *Internal Report LIG EPFL*, Juin1998, 34 pages.

[33] - T. Fuji, K. Imamura, T. Yasuda, S. Yokoi and J. Toriwaki (1995), "A Virtual Scene Simulation System for City Planning", *Computer Graphics Proceedings*: Development in Virtual Environments 1995 pp.485-496.

[34] - J. Funge, X. Tu, D. Terzopoulos, "Cognitive Modeling: Knowledge, Reasoning and Planning for Intelligent Characters", *Computer Graphics Proceedings*, Annual Conference Series, Los Angeles CA, 1999.

[35] - T. Funkhouser, S. Teller, C. Sequin and D. Khorramabadi, "The UC Berkely System for Interactive Visualisation of Large Architectural Models", In *Presence*, Vol.5, N. 1, winter (Juan.) 1996, pp 13-44.

[36] - T. Funkhouser, P. Min, I. Carlbom, "Real-Time Acoustic Modeling for Distributed Virtual Environments", *Computer Graphics Proceedings*, Annual Conference Series, Los Angeles CA, 1999.

[37] - A. Gruen, X. Wang, "Urban 3-D Mapping for a hybrid GIS", UM3'99 (Int. Workshop on Urban 3D/MultiMedia Mapping), Tokyo, 30 Sep. 1999.

[38] - M. Hodges, "Seeing Data In-Depth", *Computer Graphics World*, May 2000, pp.43-49.

[39] - M. Hodges, "It Just Feels right", *Computer Graphics World*, October 1998, pp.48-56.

[40] - H. Hoppe, "Progressive Meshes", Computer Graphics SIGGRAPH'96 proceedings, p 99-108 New Orleans USA, 1996.

[41] - K. R. Howard, "Unjamming Traffic with Computers"; *Scientific American* October 1997.

[42] - R. Ingram, S. Benford and J. Bowers, "Building Virtual Cities : applying urban planning principles to the design of virtual environments", *VRST96* pp.83-95.

[43] - W. Jepson, R. Liggett and S. Friedman, "Virtual Modeling of Urban Environments",*Presence* Vol. 5 Winter 1996 p.83-95.

[44] - M. Kallmann and D. Thalmann, "Modeling Objects for Interaction Tasks", *EGCAS'98* - 9th Eurographics Workshop on Animation and Simulation, 1998, Lisboa, Portugal.

[45] - J. J. Kuffner, J.C. Latombe, "Fast Synthetic Vision, and Learning Models for Virtuals Humans", in *Proc. CA '99 : IEEE Int'l Conf. on Computer Animation.*, Geneva, Switzerland, May 1999, pp-118-127

[46] - B. Kuipers, "The Spatial Semantic Hierarchy", Artificial Intelligence 119 (2000), pp 191-233.

[47] - J. C. Latombe, "Motion Planning: A Journey of Robots, Molecules, Digital Actors, and Other Artifacts", Int. J of Robotics Research, special issue on Robotics at the millenium, part I, 18 (11), Nov 99, pp 1119-1128

[48] - Laumond, J. P, Simeon, T. ,"Notes on Visibility roadmaps and Path Planning", LAAS - CNRS report 00073, Feb. 2000.

[49] - R. Liggett, S. Friedman, W. Jepson, "Interactive Design/Decision Making in a Virtual Urban world: Visual Simulation and GIS", proc. 15een Annual ESRI User conference, Palm Springs (CA), May 1995

[50] - Lozano-Perez , "An algorithm for Path Planning Collision-Free Paths Among Polyhedral Obstacles", Communication of the ACM, Vol. 22, Number 10, pp.560-570, 1979.

[51] - N.Magnenat-Thalmann, I.S.Pandzic, J-C.Moussaly, "The Making of the Terra-Cotta Xian Soldiers", Digital97, Geneva, Switzerland, Digital Creativity, 8(2), July 1997, pp.66-73.

[52] - D. P. Mahoney, "Philadelphia 2000", *Computer Graphics World*, June 1997, pp 30-32.

[53] - D. P. Mahoney, "High expectations for virtual Therapy", *Computer Graphics World*, Sep. 1994, pp 14-15.

[54] - D. P. Mahoney, "Moving Beyond", *Computer Graphics World*, June 1997, pp.20-26.

[55] - D. P. Mahoney, "A Star-studded Site", *Computer Graphics World*, Jan 2000, pp 51-52.

[56] - D. P. Mahoney, "Insight", *Computer Graphics World*, Jan 2000, pp 72.

[57] - M. Masuch, B. Freudenberg, B. Ludowici, S. Kreiker and T. Strothotte, "Virtual Reconstruction of Medieval Architecture", *Eurographics'99*, Sept. 99, short paper pp.87-90.

[58] - M. Michal-Maguire, "Pour reconstituer le Bloody Sunday, la justice invente le proces high-tech", http://letemps.ch, Jeudi 30 Mars 2000.

[59] - W. J. Mitchell, City of Bits: Space, Palce, and Infobahn, Cambridge, MIT.

[60] - P. Minaire, J. cherpin, J.L. Flores, D. Weber, "La mesure du Handicap dans la communauté: une micro-enquête dans un village fran¢ais", *Rapport Trimestriel Sanitaire Mondial*, 42, 1989.

[61] - K. Moltenbrey, "Creating a Winning Design, how 3D design tools were used to construct the world's most expensive resort", *Computer Graphics World*, April 1999,pp.50-51.

[62] - K. Moltenbrey, "A Tale of two cities", *Computer Graphics World*, June 1999,pp.34-40.

[63] - K. Moltenbrey, "Virtual Puzzles, animators turn 3D Jigsaw puzzles into interactive adventures", *Computer Graphics World*, Aug. 1999,pp.81-83.

[64] - K. Moltenbrey, "Engineering Metamorphosis", *Computer Graphics World*, Jan 2000,pp.52-53.

[65] - K. Moltenbrey, "Unraveling the Mysteries of the Mummy", *Computer Graphics World*, May 2000,pp.51-53.

[66] - P. L Mokhtarian (1997), "Now That Travel Can Be Virtual, Will Congestion Virtually Disappear*?", Computers Scientific American* October 1997.

[67] - S. R. Musse, C. Babski, T. Capin and D. Thalmann, "Crowd Modelling in Collaborative Virtual Environments". *VRST'98* November, 1998. Taiwan.

[68] - T. Nagakura, "The Danteum", *Eurographics'99*, September 1999, short paper pp. 90. Computer Graphics World June 1997, pp30-33.

[69] - H. Noser, D. Thalmann, "Toward Autonomous Synthetic Actors", Cyberworlds, Springer, Tokyo, 1998, pp143-158.

[70] - J. B. Novitski, "Team Building",*Computer Graphics World*, April 2000, pp.34-40.

[71] - M. E. Orlowska, X. Zhou,"A Spatial Database as a Component of Integrated Database System", in *Proceeding of DANTE'99 International Symposium on DB Applications in non-traditional Environment*, Nov. 1999, Kyoto, Japan, pp 104-.114.

[72] - D. Osorio, W.M. Getz, J. Rybak, “Insect Vision and Olfaction: different neural architectures for different kinds id sensory signals”, *From Animal to Animat*, Vol3, pp73-81.

[73] - Open Inventor C++, Reference Manual, *The official Reference Document for Open Inventor, release 2*, Addison Wesley 1994, ISBN 0_201_62491_5.

[74] - L. Porcher Nedel D. Thalmann, "Modeling and Deformation of the Human Body Using an Anatomically-Based Approach", Proc. Computer Animation'98, Philadelphia June 1998, pp34-40.

[75] - L. Porcher Nedel, "Simulating Virtual Humans", *Tutorials notes Sibgraphi'98 - Simposio Brasileiro de Compata¢ão gràfica e processamento de imagens*, Oct. 98.

[76] - C. D. Potter, “A new Role for 3D plant models”, *Computer Graphics World*, June 1997, pp.37-39.

[77] - C. D. Potter, “SimFactory2000”, *Computer Graphics World*, Feb 2000, pp.27-30.

[78] - T. J Prescott, "Spatial Learning and Representation in Animats", *From Animal to Animat*, Vol. 3, pp164-173.

[79] - C. Prins, *Algorithmes de graphes*, Edition Eyrolls, 1994.

[80] - A. Pruski, "Le projet VAHM: un robot d'assistance à la conduite de fauteuil pour personnes handicapées", *conf. Automatique et Santé*, Montpellier France, June 1997.

[81] - A. Pruski, O. Habert, "Obstacle Avoidance Module for the VAHM-2 Wheelchair", AAATE 99, the 5th European Conference for the Advancement of Assistive Technology, Düsseldorf, Germany, Nov 1999.

[82] - S. Raupp Musse, "Human Crowd Modelling with Various Level of Behaviour Control", Phd Report, Polytechnical School of Lausanne Num. 2128,

[83] - M. Reddy, L. Iversion, Y. G. Leclerc, “TerraVison II: Visualising Massive Terrain Databases in VRML”, *IEEE Computer Graphics and Applications*, March 1999, pp30-38

[84] - M. Reddy, L. Iversion, Y. G. Leclerc, “Under the Hood of GeoVRML 1.0”, *proc. of the fithWeb3D/VRML Symposium*, Monterey (California), Feb. 2000.

[85] - O. Renault, N. Magnenat Thalmann, D. Thalmann, "A Vision-based Approach to Behavioral Animation", *The journal of Computer Graphics and Computer Animation*, Vol. 1, No 1, pp18-21.

[86] - C. W. Reynolds, "Steering Behaviors for Autonomous Characters", Game Developers Conference, 1999, web page: http://www.red.com/cwr/steer/

[87] - B. Robertson, "Virtual Humans at Work", *Computer Graphics World*, July 1997, pp 33-37.

[88] - B. Robertson, "The toys are back", *Computer Graphics World*, Nov. 1999, pp27-32.-

[89] - B. Robertson, "Crowd Control", *Computer Graphics World*, Feb. 1998, pp30-40.

[90] - B. Robertson, "Faces and Crowd", *Computer Graphics World*, Jul. 1998, pp61-63.

[91] - B. Roehl, "Virtual Archeology bringing new life to ancient worlds",Innovation3, summer 1997, pp. 28-35.

[92] - A. Rowell, "CAD/CAM Special Report, The Challenge of CAd Interopability", *Computer Graphics World*, June 1997,pp.57-72.

[93] - R. Rouse III, "Do Computer Games need to be 3D?", *Computer Graphics World*, May 1998, pp 64-66

[94] - T. A. Russ, R. M. MacGregor and B. Salemi (1996), "VEIL: Combining Semantic knowledge with Image Understanding", ARPA Image Understanging Workshop 1996

[95] - B. O Schneider, P. Borrel, J. Menon, J. Mittleman and J. Rossignac, "BRUSH as a Walkthrough System for Architectural Models", IBM Research Report RC 19638, Oct. 1997, http://www.cs.cmu.edu/afs/cs/user/garland/www/multires/bib.html .

[96] - E. Schweiss, S. R. Musse and F. Garat, "An Architecture to guide Crowds based on rule-based systems", *Autonomous Agents'99*, 1999.

[97] - J. Shen, "Human Body Modeling and Deformation", PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 1996. Advisor: Prof. Daniel Thalmann.

[98] - Shih-kai Chung and J. K. Hahn, "Animation of Uman Walking in Virtual Environments", *Computer Animation'99*, May 1999, pp. 4-15.

[99] - J. Silvetti, C. Smith, T. Dumbleton, S. Williamson and G. Taylor, "Site and Structure at the Vatican: from the earliest settlement to the present"", *Eurographics'99*, Sept. 1999, short paper pp. 92-95.

[100] - S. Smith, D. Duke, M. Massink, "The Hybrid world of Virtual Environments",*Eurographics'99*, Sept. 1999, short paper

[101] - D. Sorid, s. K. Moore, "The Virtual Surgeon", IEEE Spectrum (ISSN 0018-9235), July 2000, pp 26-31.

[102] - C. Strohecker, B. Barros,"Make Way for WayMaker", in the journal *Presence*, Vol.9, N.1, feb. 2000, pp 97-107.

[103] - G. Thomas, "Environnments virtuels urbains: modelisation des informations nécessaires à la simulation de piétons", Rapport de thèse num. 2300, Université de Rennes, Dec. 1999.

[104] - G. Thomas, S. Donikian, "Modelling virtual cities dedicated to behavioural simulation", *Eurographics'2000 Vol. 19, N.3*, Augt. 2000, pp. 71-80.

[105] - P. M. Todd, S. W. Wilson,"Environment Structure and Adaptative Behavior From the Ground Up", in From Animals to Animats 2: Proceedings of the 2d International Conference on Simulation of Adaptative Behavior, edited by J. A. Meyer, H. L. roitblat, and S. W. Wilson, pp11-20. Cambridge, MA:MIT Press/Bradford Books, 1993.

[106] - K. R. Thorisson, "Gandalf: an Embodied Humanoid Capable of rreal-time Multimodal Dialogue with people", 1st ACM, International Conference on Autonomous Agent, Feb. 1997, California.

[107] - K. Tyrka, "Reaching for the Sky", *Computer Graphics World*, Nov. 1999,pp.56-57.

[108] - C. Van Geem, T. Simeon, J.P. Laumond, J.L. Bouchet, J.F. Rit, "Mobility Analysis for Feasibility studies in CAD models of Industrial Environments", CNRS LAAS report 98417, Oct. 1998.

[109] - The Graphisoft, international competition, (ArchiCAD Program) official web page: http://www.gsprize.com/

[110] - DigiScents web page: a smelling interface tool, http://www.digiscents.com

[111] - A CAD web page presenting different links and software. page : : http://www.tenlinks.com/CAD/

[112] - Metaphase Web Page, Managing Products Information in the Virtual Entreprise page: http://www.sdrc.com/nav/software-services/metaphase/

[113] - Web Page presenting Alstom Transport project using Metaphase : http://www.cadcam-magazine.co.uk/htm/_casestudies/080002.htm

[114] - Web Page presenting a 3D model of a temple of Horus, page: http://www2.sis.pitt.edu/~jacobson/horus/horus-3d.html

[115] - Web page presenting parts of Pompei reconstructed, page: http://www.pitt.edu/~jljst15/pompeii/pompeii-shots.htm

[116] - Web page of the French National Institut of Geography, IGN, http://www.ign.fr .

[117] - Web page of the museum La Villette, http://www.cite-sciences.fr .

[118] - VRML2 Specications Web Page: http://www.vrml.org/specifications/

[119] - Panorama tool web page,http://www.habware.at/duck3.htm#PANORAMA.

[120] - Web page Casus Presenter home page. http://www.igd.fhg.de/CP.

[121] - Web page The Cube (film). http://cubethemovie.com/cube.htm

[122] - Web Page for 3DSMax product : http://www2.discreet.com/products/products.html?prod03dsmax

[123] - Web Page for CROSSES Project : http://crosses.matrasi-tls.fr

Curriculum Vitae

Nathalie Farenc Fernandes
French citizen
Born in July 12th , 1970
Adress : Rte de la Maladiere, 16
1022 Chavannes Switzerland or
2 Place des Jacobins, 38130 Echirolles France
E.mail : nathalie.farenc@epfl.ch
www : http://ligwww.epfl.ch/nathalie.html

Education:

I made a DEA (diplome d'étude approfondie) in Saint Etienne , France, on synthesis and image analyse (1994-1995). I have a background in computer science (DUT and maitrise) from University of Chambery and Grenoble (France)

Experience:

Research Assistant (May 1996 - present) Computer Graphic Lab. (DI-LIG)
Swiss Federal Institute of Technology, Lausanne, Switzerland

Teaching Assistant (OpenGL 1999 2000), project supervisor (9 semester projects, 2 diploma projects)

Publications:

N.Farenc, F.Sidler, A.Ferrando, D.Thalmann, Database and Modelling Strategy: A Compliant Way for Display Optimisation, Journal of Applied Systems Studies (to appear)

N.Farenc, D.Thalmann, Autonomous Virtual Humans in a Virtual City,VPK Journal, special issue on visualization (to appear July 2001)

Farenc N., Raupp Musse S., Schweiss E., Kallmann M., Aune O., Boulic R., Thalmann D. "A Paradigm for Controlling Virtual Humans in urban Environment Simulations". Applied Artificial Intelligence Journal - Special Issue on Intelligent Virtual Environments. Vol 14 Nb 1, Jan 2000,p69 - p91.

N. Farenc, C. Babski, F. Garat, D. Thalmann, " Database and Virtual Human Representation on the WEB", Advances in Databases and Multimedia for the NewCentury -A Swiss-Japanese Perspective-, World Scientific (to appear)

N. Farenc, F. Sidler, A. Ferrando and D. Thalmann, "Database and Modelling Strategy: A Compliant Way for Display Optimisation", Proc. WISE 2000, Hong Kong Jun 2000.

D.Thalmann, N. Farenc, R. Boulic " Virtual Human Life Simulation and Database: Why and How", Proc.International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), IEEE CS Press (to appear).

Nathalie Farenc, Ronan Boulic and Daniel Thalmann "An Informed Environment dedicated to the simulation of virtual humans in urban context" Eurographics'99, September 1999, Milan Italy.

Farenc N., Raupp Musse S., Schweiss E., Kallmann M, Aune O., Boulic R., Thalmann D. "One step towards Virtual human management for urban Environment Simulation." ECAI 98 Workshop on Intelligent Human Interface, Brighton.

DEA report on perspective perception and history : "Nouvelles perspectives et effets d'optiques en synthèse d'images" LISSE, Ecole des Mines de Saint Etienne, France 1994