Virtual Human Life Simulation and Database: Why and How

Daniel Thalmann, Nathalie Farenc and Ronan Boulic
Computer Graphics Laboratory
Swiss Federal Institute of Technology
Lausanne, Switzerland
nathalie, boulic, thalmann@lig.di.epfl.ch

Abstract
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. We present an Informed Environment, which corresponds to a database dedicated to urban life simulation. We discuss about use of the database, methods and tools for creating and providing the information necessary for animating virtual humans in a city. The Informed Environment is based on a hierarchical decomposition of an urban scene into Environment Entities providing geometrical information as well as semantic notions, thus allowing more a realistic simulation of human. In this manner, virtual humans can be provided with a certain kind of urban knowledge. Moreover, the database can offer other types of services for scene creation for example.

1. Introduction
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 quantity of data attached to the virtual scene we use for simulation a simple database. This database corresponds to our Informed Environment and contains different kind 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 quantity of geometrical data originating from the model (altitudes 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 sidewalk 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 [4], the Informed Environment is a good way for substituting perceptions and analysis by information obtained through database access. The decomposition of action such as "walking on a sidewalk" involves the recognition of places called "sidewalk" (location in space and verification of the coherence of the walking action associated with the mobile entity) and the animation of the body in the virtual scene using a walking model (according to the previous data given as parameter). Human motion (by key frame system or walking motor) is handled at a lower level and is not addressed in this paper [5]. This environment provides the information necessary for the recognition of places and the location of various objects.

In this paper, after a related work presentation, we focus in a second part on a definition of an Informed Environment, in a third part on the creation of an Informed Environment. Section 4 presents the use of such environmental database and section 5 the mix of different databases. Finally, Section 6 shows some results and ends with a conclusion.
2. Related Work

We present a method for building a virtual scene with associated semantic information as well as for the exploitation of such scenes. These scenes with their database associated can be easily manipulated and used to construct new environment for specific simulation. Several studies have addressed the topic of virtual building or city construction [1], [17], [20], [26], such as reconstructing a real city using image processing, archaeological data, sophisticated tools [6], [23], [10], or using the urban context to deal with traffic problems [10], [8], [18], urban displacement [9], [10], [2] or city modelling [14], [10]. The main difference between our approach and previous studies concerns the simulation of a city with realistic virtual humans. By comparison with simulations using icons that represent humans, simulations with realistic humans allow a better approach for evaluating real constraints in a real life city simulation. However, the animation of realistic virtual humans has the drawback of adding a lot of constraints as well as data needs. Thus, we aim at obtaining an urban model with integrated knowledge adapted to human behaviour simulation. As in [10], our work is based on environment decomposition with links to traditional two-dimensional Geographic Information Systems (GIS) and databases. In order to structure the virtual city data, we define a hierarchical model valid for different kinds of simulation of human life. This model is linked to methods of scene construction and usage.

3. Environmental Entities

3.1. Definitions

At a first stage, an analysis permits the identification of major constraints for urban life simulation with virtual humans. These constraints concern the information that is 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 needs all the geometrical information associated to this place. This information includes a good definition of the space surrounding the agent, the whereabouts 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 entities. Knowledge about objects is used for dealing with collision avoidance or for interacting with them. 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 behaviours 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 a urban context are too numerous and complex. Another possible method is to distribute information or “knowledge” to specific applications [11]. In this manner, one application deals with internal crowd management [19], 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 [24] and finally, a last 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 [11], 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 problematic 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.

3.2. 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 of 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
the city. At a lower level, these axes will be recognised as streets. In the database, the street level provides information about crosswalks and sidewalks. As we use the notion of encapsulation, the same surface can belong first to a sidewalk, 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. The figure in annex shows a graph representative of this structure. Our model can decompose a city in twenty-nine different Environmental Entities. In the hierarchical decomposition tree, there are ENV's corresponding to the same ENV type but with different ascendant ENV such as, for example, 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 1. View of the hierarchy for a selected part of the street

3.3. 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**
  - type : 1
  - nb_under : 3
  - notraffic_zone
  - building

- **name of the entity**
- **type of the entity**
- **number of entities composing the entity**
- **list of the sub_entity names**

- **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 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)

Figure 2 represents the database model. Briefly, according to hierarchical decomposition model an ENV composes its parent ENV, and its parent ENV can be decomposed in one ore more ENV(leaf or not leaf). The points belong to a leaf ENV and a leaf ENV has 4 points associated. The paths stored in the database are across the main ENV to which they are associated. The objects are detected and linked to the surface of an leaf ENV and we can have several places such as a park, but all the leaf ENV's are not associated to a place.

Figure 2. The Database model

3.4. 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 type to the objects in an urban model. This decomposition has been defined in collaboration with a designer [3] 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. 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 key in the scene and after in the database. The labels associated to the current ENV and all ENVs above in the scene hierarchy compose these names. The figure in annex represents names given by the designer and the re-naming using labels or file names.
An example is the case of a sidewalk. The designer gives the object the name "TRxx" for a sidewalk, this object is under a group labelled "RUE_VH01_INT" representing a street segment. This entity represents the entire street, and by using part of the name, we are able to fully name the sidewalk "RUE_VH01_TRxx". The link between sidewalk and street is established at this stage by the new name of the object. All the objects are named by the designer based on the object naming previously defined. Applying the rules of re-naming of ENV, the entire scene is read and all ENVs are fully named.

4. Extraction of ENV and Database Creation

4.1. Generalities

The three-dimensional scene provided by the designer is divided into two parts, one for visualisation and another for database construction. The scene is parsed and all the object names (the full names) are analysed using the ENV prototype previously defined. With this prototype, for all the ENV, we know their type and location in the hierarchy tree. Then, the ENV are extracted from the scene with their geometrical characteristics (surface, volume, location in space with matrix, bounding box) and associated semantic. The ENV's are recognised and the new information is stored in the database. An ENV is represented in the database by its geometry, location in the scene (position matrix), and links to other ENV (link of decomposition (ascendant and descendant) or connectivity with other ENV having the same ascendant ENV). Depending on the type of the ENV, the ENV geometrical representation can be deleted from the scene in order to create a visual scene. In order to create some graph representing the connection between ENV's, we need points to pass to an ENV to another one. Then, in a second stage, for each ENV useable by mobile entities, an entry/exit points computation is performed depending on the ENV type. We create in the database a list of points linked to the ENV. A point belongs to one ENV and an ENV can have several points associated. With this method a location in space can be represented once, twice ore more. Like points representing the surface of an ENV, one of the next evolutions of the work will be to eliminate such redundancy and to associate the same location to one point belonging to one or more ENV. For each point, we add the information concerning the mobile entity type that is able to use them (bus or pedestrian for example). In the case of the sidewalk, two points are created at the sidewalks extremities, the associated ENV prototype being the sidewalk, and only the pedestrian mobile type is the mobile type allowed to use them. Another kind of information concerns the connectivity links between the ENV. We have chosen to compute and store for the ENV only connectivity links having the same ascendant ENV, in order to minimise the number of stored connectivity links. In the sidewalk example, its ascendant is a street, and the rolling ways and crosswalks are connected entities having the same ascendant. Only these links are stored in the database. More connectivity computation with other streets and parcels is carried out during the simulation.

4.2. A Segmented Scene

Database creation relies on a segmented scene. All regions on the ground that provide specific information in terms of functionality are modelled as separate objects, that are distinct from the rest of the scene. To illustrate this idea, Figure 1 shows a selected sidewalk. Only its own representation is displayed prominently. Figure 4 illustrates the Inventor hierarchy [21] corresponding to the scene segmentation, the selected object and the different labels used to provide information.

This segmentation of the scene is for the moment carried out at the design level. It could be done in another way through a user interface, with a virtual scene modelled without object segmentation or by object recognition from a real image. Once we have a well-segmented scene, the associated Informed Environment is created on the same model of decomposition.

![Figure 3. View of the database representation for the block and for a junction.](image)

4.3. Extraction of moveable object, lures and Smart Objects

A scene corresponding to the hierarchical model of decomposition includes both graphical objects for visualisation purpose and objects providing via their own visual representation some semantic information. First of all, we detect the moveable objects in order to extract their representation in files to be able to change their location during the simulation. In the database we associate the notion of moveable objects to them and the scene display tool, must takes care to these additional objects. We have also defined some transparent objects that are created only for carrying semantic information and that are subsequently deleted from the final visualisation scene. Using the same notion of objects naming to extract data from the scene, special objects are created only to provide information to the database. Called lures, ones of these objects represent additional information used for collision detection in the case of...
complex objects. We detect the objects attached to a circulation area during database creation. We automatically compute their bounding box to facilitate collision detection during simulation. If the objects have specific shapes, such as a U-shape, it may be useful to give access to the inside of this U-shape. In this case, during scene building, we create some lures associated with the visual object, in order to yield bounding boxes adapted to simulation. In a station scene we have designed a group of seats. These seats must be defined in order to avoid an agent colliding with them during simulation. As a global bounding box keeps virtual humans from using the area inside the group of seats, we defined some lures so as to still provide information for collision avoidance, but also in order to minimise the bounding box size. These lures are not in the visualisation file.

Other kinds of lures contribute to action knowledge such as climbing stairs, for example (with parameters for stairs such as size of the step, depth and height, number). This information can be analysed before a climbing motion in order to modify the walking model using these parameters, or can indicate a key frame file stored in the stairs place to perform the body motion during simulation. In this case, the lure is a surface at the top and bottom of the stairs. During database creation, this object is also deleted from the visual scene.

Other objects, named smart objects, market out by their specific labels, are also extracted from the scene. Their graphical representations create some archives as starting points for their internal human interaction specification [15]. These objects are referenced in the database with their location and links with ENV.

5. Utilisation of the ENV Database

5.1. Localisation of Objects, Areas and Mobile Entities

The database contains geometrical and semantic information for mobile entity simulation. An area defined as an Informed Environment provides sub-areas (ENV), along with the list of objects with which to avoid collision. In order to minimise the number of objects, peripheral objects such as walls are not included in the ENV surface. The database corresponds to perception such as being aware of all the objects inside an ENV or all the surfaces adjoining an ENV in order to more efficiently perform virtual perception. 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 definition, 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 etc".

5.2. Path Creation by the means of an Informed Environment

The Informed Environment provides also the possibility of rooting mobiles through the city using the database. The idea is to compute the best path from one point to another one, according to the mobile type used. A list of points can be enough for one agent but not if we want to simulate more mobiles. For this purpose our paths are represented by some areas, which are our ENVs with some Entry/Exit points to exit and enter in a new surface according to the mobile and to the semantic attached to the zones. 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 sidewalks, crosswalks and circulation areas. The method implied correspond to a graph with the nodes representing the ENV and the edges corresponding to the points allowing passage from one ENV to another. The weight associated to the edges is the result of the computation of the distance between the Entry/Exit points. A modified version of the Dijkstra algorithm [13], [22] adds edge dependencies and allows to find the best path (shortest, fastest). We have a specific graph for each kind of mobile entities and some ENV allow a connection between graphs. For example the bus stops are the links between graphs for pedestrians and buses. Using GUI interface, the user selects the start and the end point for its path. More about path constraints and computation can be found in [12].

5.3. A Database associated to the Notion of Places

In order to guide 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 sidewalk named "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-sidewalk, and these names have been provide by the user and attached to ENVs in the database. We can have several parks and different methods can be applied in order to know which one is mentioned. Using a GUI interface the user associates a place to the different 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 for example 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 (for more details about the treatment of such order [11]).
5.4. Action Planning

The Informed Environment can store other types of data 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.

6. Construction of an urban Informed Environment

6.1. A City: mix of different databases

For the time being the scene is created using 3DSMax, and converted to the Inventor format (as we are working at the geometric data level). We have produced different databases: one for the city layout, and others for the different parcels containing information. The city was created on the model of the decomposition rules and objects naming defined jointly by the designer [3] 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, the layout of the city corresponds to 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 had the park which can be considered as a database, a museum used for other simulation and some buildings with a database associated or not. To obtain the final database we have two possible methods. The databases of the city layout and all the parcels are mixed using the different labels present in the layout database and in the database to be added. We make a substitution and add new data. The other 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.

By the way 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 building interior. For this scene we have defined a database. Containing 313 ENV's, 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.

![Figure 4. View of the Inventor hierarchy and a method to create new scene with database associated.](image)

Figure 4 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. As most of the scene representation is segmented in a set of files the main scene at the end of the construction is not very large due to the use of “inline” for importing the different parts of the scene. In the bloc case, the start file is the layout composed of files for each street segments and junctions. After, for each parcels, only one line in the main file represents the new part of the town, modification are very simple (need to know the label of the parcel and the name of the new representation) and the scene resulting is represented via simple files. This scene construction can be done in the same time as database merging or can be dissociated, in this case only the visual representation of the scene is created and the user has to verify that the database associated still stick to the visual scene.

In order to have a common set of scenes for simulations, we have also define a method to add easily furniture in the scene and inside the database. Using the location of the objects and the database, we associate the objects to the ENVs corresponding to the same surface. Some constraint on the heigh of the object must be used to avoid detection collision with lamp ceiling. The added scene containing the objects can be parsed once to create
a small database associated to the scene and after only files corresponding files can be used for objects knowledge. In this way a scene can be easily modified and the database easily updated.

6.2. A Train Station

A part of the town scene, the train station, was created using the same rules [3], the database associated contains 42 ENV’s and 142 points. The two stairs in the station are the particularity of this scene, one at zero altitude, and another one underground (minus five meters). The humans can go to one floor to the other one using smart objects like escalators [15]. In this case, path computation must use common surfaces between the ENV and the smart object composition. The locations of points in space are really carried out in 3D, as it is not possible to project points onto the ground. Some ENVs slope according to the surfaces of the escalators, in this way it is possible to compute paths between the floors. The same method allows defining the different levels of the elevator and the connections between the floors via lifts.

6.3. A Museum

This building has been a good simulation scene for moveable objects. Figure 5 presents some externals and internal views of the museum and the database attached to the building with the bounding boxes of the objects in green for fixed objects and in blue for moveable objects.

![Figure 5](image)

The database constructed from this scene contains a restaurant with tables, glasses and bottles. Few of them have been defined as moveable objects and during the database creation their visual representation has been extracted from the scene and isolated in different files. The database has been used to initialise a context for an application of virtual perception [3].

6.4. Integration

We have integrated this work into different applications. The major one is a crowd module in the city, several paths are pre-computed and the crowd uses the paths defined by a list of surfaces with entry/exit points associated, to simulate an inhabited town [11][19]. All these paths are loaded before the simulation and the crowd simulator distributes this information to the different groups. Figure 6 and Figure 7 are pictures taken from simulations in the city using the crowd system. In the context of the city, we defined paths for pedestrians and buses. Human crowds walk through the city according to pre-calculated paths (autonomous crowd) and using only areas authorised for pedestrians [19]. Another simulation concerns the bus. The latter sends a message to all agents waiting at the bus stop where it is standing. Using the reacting system of the crowd [16], the virtual humans are able to take the bus: they react when the bus arrives at the bus stop. The bus asks at what station where the human wants to step out, and when this location has been reached, it sends a message to the humans telling them to get off the bus. The humans keep on reaching points on their paths, and the bus continues on its way. The paths for humans taking the bus were calculated using the path creation tool, with the criteria of fastest path in the city.

![Figure 6](image)

Another integration is made during simulations, as defined in [24]. The Informed Environment is a client, other clients are the crowd module (with a guided crowd), a rule based behaviour tool and a smart object application. A server distributes the different messages it receives to the client able to provide information or the service associated to the message. From the point of view of the Informed Environment client, during the simulation, the server can send it requests concerning the location of different places, the location of objects or the location of an agent providing the (x,y,z) parameters. Based on this kind of simple request, the Environment database and the path planning tool provide some paths. The pedestrians are guided by other modules using the Informed Environment to deal with the semantic environment.
Figure 7. Picture of simulation of crowd following pre-computed paths in the city

We have made some simulations of an inhabited station. In this case, we have some "autonomous" pedestrians having initial script including some path and objects definitions. These mobiles move without interaction of other modules during the simulation (only with smart objects if these interactions have been defined in the initial script)[19]. Other pedestrians are named "guided crowd" and they follow order provided by clients or interface using services from all other s clients. The database is used to extract information for path, object during simulation or to create an initial context for virtual perception [3].

7. Conclusion and Future Work

In this paper, we have presented an Informed Environment that creates a database dedicated to urban life simulation. Using a set of manipulation tools, the database permits the integration of what we call "urban knowledge" in order to simulate more realistic behaviours. Moreover, for various types of mobile entities, we can compute paths to move through the city according to area rules. By using data originating from the environment, virtual humans are able to acquire urban behaviour. Future work will concern inclusion in the database of more complex behaviour rules as well as actions to perform in specific places. We also aim at informing, via an intelligent interface, any well segmented scenes after the creation stage. We are also working on a methodology to detect incoherence between the scene visualisation and the attached database, to avoid redundant points in the database. Others tests would avoid multiple objects with the same name (verification in the database when adding new objects). Graphical Interface could help in the verification of spatial constrains and scene modifications.

Acknowledgments

This research was sponsored by the Swiss National Research Foundation. The authors would like to thank Frederic Sidler and Olivier Aune for designing the scenes and their collaborations. We would also like to thank Joaquim Esmerado, Lorna Herda and Ik Soo Lim for their proof readings, Soraia Raupp Musse for her helpful comments, suggestions and collaboration.

References


Figure 8. Hierarchy with fully named objects