From geometry to spatial reasoning : automatic structuring of 3D virtual environments

Carl-Johan Jorgensen, Fabrice Lamarche

Université de Rennes 1
IRISA, UMR CNRS 6074
Université Européenne de Bretagne
carl-johan.jorgensen@etudiant.univ-rennes1.fr
flamarch@irisa.fr

Abstract. *In this paper, we address the problem of automatically creating a meaningful spatial representation of 3D virtual environments, suitable for spatial reasoning. We propose a spatial analysis technique that distinguishes indoor, outdoor and covered parts of the environment. It also separates buildings into floors linked by stairs and represent floors as rooms linked by doorsteps. On this basis, we compute a natural hierarchical representation of the environment. We also demonstrate that this representation can be used to handle multi-criterion queries relating to spatial reasoning including zone selection and path planning.*

Keywords: spatial analysis, spatial reasoning, path planning

1 Introduction

Many application fields such as video games, virtual reality or virtual training, tends to immerse the user in virtual environments automatically crowded with autonomous virtual humans. In order to improve the user's experience, these humanoids must show realistic behaviors. One of the most important behaviors is the ability to navigate inside a virtual environment as this skill is continuously used. Human navigation is influenced by the nature of the environment [1, 2] and eventually by external factors. For instance, if a pedestrian wants to go from home to the nearest bakery in a rainy day, he may want to follow a covered path. However, he will not walk through his neighbors' houses under the pretext it is the shortest and better-covered path. To reproduce such a behavior, a virtual human should be aware of buildings locations and be able to plan paths that make a compromise between the travelled distance and the use of covered zones. This kind of information should be available when using informed virtual environments. However, most of 3D environments are modeled with modelers that do not provide an information system. In such a case, manually adding information to the geometry can be a long and tedious process.

In this article, we propose an original method that aims at automatically extracting a meaningful representation from 3D environments. This method relies

on a spatial analysis process that distinguishes indoor, outdoor and covered environments. It also identifies buildings and their internal structure. Floors linked by stairs are identified and floors are represented as rooms connected through doorsteps. We show how this representation can be used to handle zone selection and multi-criteria path planning in order to enhance the quality and credibility of generated paths. In the next section, we present related work. Then, the spatial analysis and spatial reasoning processes are explained and discussed in the result section.

2 Related work

To populate a virtual environment, modelling the navigation behaviour is crucial. This behaviour relies on the ability of planning a path inside a complex environment which itself relies on an adequate representation of the environment structure. Path planning and environment representation have been widely studied in robotics where navigation is a necessary task to achieve [3]. In this field, two main techniques can be distinguished: the roadmap and the cell decomposition approaches.

The roadmap approach captures the connectivity of the free space thanks to sets of standardized paths [4–6]. This kind of approach focuses on creating a data structure enabling path planning but does not explicitly model the borders of obstacles which is a prerequisite of our approach. The cell decomposition approach represents the free space with a set of cells. Those decomposition approaches can be either approximate or exact. Approximate cell decompositions uses predefined cells shapes (uniform grids, quad trees, circles) whose union is strictly included in the free space [7, 8]. Exact cell decompositions exactly cover the free space. Among other techniques [3], Constrained Delaunay Triangulations (CDT) have been used in last ten years to compute an exact cell decomposition of virtual environments [9, 10] and has been slightly modified to identify bottlenecks (most constrained part of the environment) [11]. Its good properties in terms of navigation queries [12] and path finding [13] have been demonstrated.

Based on constrained Delaunay triangulations, several techniques have been proposed to produce a hierarchical abstraction of the environment topology. The main idea is to qualify cells or sets of cells (regions) given the number of accesses (dead ends, corridors, crossings) [11, 14]. On this basis and eventually on the basis of the geometric shape of the identified regions, a topological abstraction is automatically computed and mainly used to increase path planning performances. Such a kind of abstraction is based on the geometric properties of the identified regions but does not rely on standard notions such as indoor / outdoor locations or rooms, corridors, floors or stairs for instance.

To handle more complex cases and increase the realism of simulations, the notion of informed environment has been introduced. The idea is to associate a data structure to regions of the environment. This data structure stores information dedicated to the behaviour [7]. This approach has been used to model populated cities [1] and to propose semantic abstractions of a city structure [15].

More recently, this approach has been used in [16] to propose a hierarchical decomposition of multilayered environments taking into account notions of floors and stairs for instance. It has also been used in the field of geographic information systems to provide an informed and hierarchical representation of virtual geographic environments that is used to speed up and enhance the quality of computed paths [17]. One main aspect of informed environments is the fact that they need to be modelled with specific tools or manually informed on the basis of the original geometry.

The aim of our approach is to automatically generate a meaningful and hierarchical representation of virtual environments that is suitable for path planning and more generally for spatial reasoning. It relies on an analysis of the 3D environment structure to automatically generate an abstraction of the environment including notions such as indoor / outdoor locations, identification of floors and stairs in indoor environments and a topological decomposition including notions of rooms and corridors that relate to the real nature of the environment. In those terms, it generates an augmented representation that is closer to the notion of informed environment than previously proposed approaches in the field of automated environment analysis.

3 Spatial analysis of geometric environments

In our everyday life, we unconsciously differentiate buildings and outdoor environments. Inside buildings, we identify floors, stairs, rooms or corridors for instance. The aim of the proposed method is to automatically create a representation that includes those concepts. To do so, we analyze a 3D environment geometry in which all furniture has been removed. We start by computing a spatial subdivision of this geometry which is well suited for spatial analysis. Then, this spatial subdivision is analyzed to compute zones that can be identified (indoor, outdoor, rooms, stairs...). Once this analysis is achieved, a hierarchical representation, grouping rooms into floors, floors linked by stairs into buildings is computed.

3.1 Prismatic subdivision and low level topology

The prismatic subdivision has been introduced in [18] to analyze the structure of 3D environments provided has a set of 3D triangles. Its aim is to organize a set of 3D polygons in order to capture ground connectivity and identify floor and ceiling constraints. It represents the environment by a set of vertical 3D prisms dividing the input database into layers of 3D triangular cells. The figure 1 depicts the computation process, for more details we invite the reader to read [18]. The figure 2 shows the prismatic subdivision of a simple environment that will be used in the following to demonstrate the results of our spatial analysis.

Based on this spatial subdivision, with respect to some characters characteristics (minimum navigable height, maximal navigable slope and maximum surmountable height), a low level topological graph is constructed. Each node

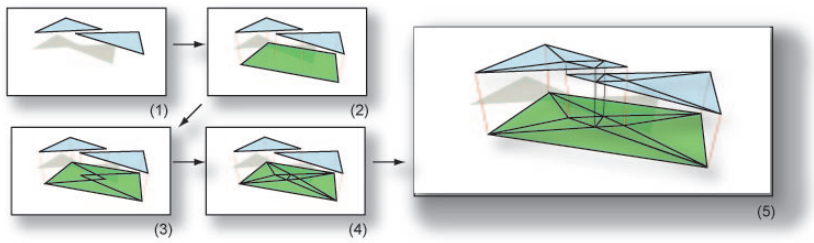

Fig. 1. Prismatic subdivision computation.

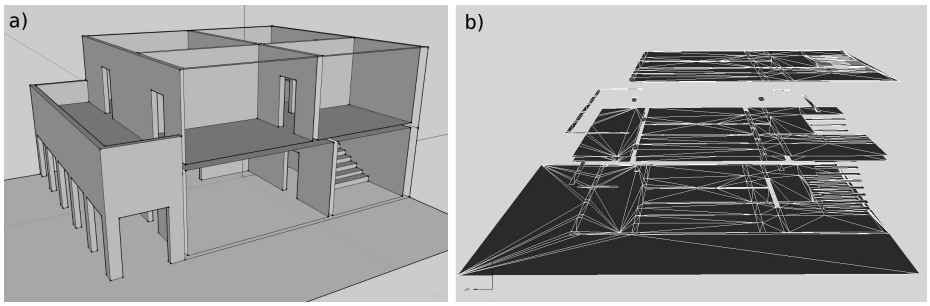

Fig. 2. a) A simple example environment (wall and ceiling masked for more visibility) b) Its prismatic decomposition as generated by TopoPlan [18]

of this graph is a navigable 3D triangular cell (in terms of minimum navigable height and maximum navigable slope) of the subdivision and each edge is an accessibility relation between cells. An edge is created between two nodes if the cells are located into adjacent prisms and if the height difference between those cells is lesser than the maximum step height. Edges are tagged *continuous* if the height difference is lesser than a given threshold or *step* otherwise. The prismatic spatial subdivision and its associated low level topological graph constitute the input of our spatial analysis algorithm which is described in the following section.

On the basis of the low level topological graph, the 3D triangular cells of the spatial subdivision can have three type of borders: *step*, *obstacle* and *surmountable*. By extension, borders of zones (groups of connected cells) can be classified into those three categories.

3.2 Meaningful zones computation

The prismatic spatial subdivision and its associated low level topological graph constitute the input of our spatial analysis algorithm. Based on those data structure, we propose a four step algorithm which aims at (1) differentiating interior and exterior zones, (2) extracting floors and stairs and (3) decomposing floors into rooms and doorsteps.

Covered and uncovered regions. The first step of our algorithm identifies covered and uncovered regions of the environment. This step uses the prismatic spatial subdivision to tag 3D triangular cells as *covered* if the cell is surmounted by another cell belonging to the same prism and *uncovered* otherwise. Based on this process, two set of zones (group of mutually accessible cells) are extracted: $\mathcal{Z}_c$, the set of covered zones and $\mathcal{Z}_u$, the set of uncovered zones. The result of this decomposition on our example environment is depicted Fig. 3(a).

Floors and stair steps. The second step consists of dividing each zone of $\mathcal{Z}_c \cup \mathcal{Z}_u$ at any *step* border. Those step borders are identified with the low level topological graph computed in the previous section. This produces two new sets of zones : $\mathcal{Z}'_c$ and $\mathcal{Z}'_u$. In covered zones, this process extracts floors and stair steps. In uncovered zones, this process can differentiate roads and sidewalks for instance. Zones mainly having *step* borders are tagged *step*. The figure 3(b) depicts the result of this process on our example environment.

Topological maps. For each zone of $\mathcal{Z}'_c \cup \mathcal{Z}'_u$, we compute a topological map. This map is computed using the 2D spatial subdivision technique presented in [11]. A constrained Delaunay triangulation is computed on the projection onto the XY plane of the zones borders. This triangulation keeps the nature of the zone borders namely *obstacle* or *step* border. This Delaunay triangulation is then modified by computing the bottlenecks [11] and triangles are merged into convex polygons if all bottlenecks remains present as a border of convex polygons. Keeping the identified bottlenecks is important as they are likely to identify doorsteps borders. The result of this processing applied to our example is depicted in Figure 3(c).

Room decomposition. Each covered zone identified as a *floor* is assumed to be an interior zone belonging to a house or a building. A building floor or house floor is composed of rooms and corridors separated by doorsteps. To achieve room decomposition, we thus need to identify doorsteps. To identify those doorsteps, we define a 'door likelihood' function that is computed for each convex cell c of the topological map. Let $S(c)$ be the surface of cell c, $H(c)$ be the average ceiling height of cell c, $N(c)$ be the set of neighboring cells, B(c) be the set of *free* borders of cell c and $L(s)$ be the length of border s. The 'door likelihood' function is computed thanks to three criteria:

- $C_1(c) = \frac{\sum_{c' \in N(c)}(S(c'))}{S(c)}$. A door is a small zone between bigger ones.
- $C_2(c) = \sum_{c' \in N(c)}(\parallel H(c') - H(c) \parallel)$. A door's ceiling is often lower than the one of surrounding rooms.
- $C_3(c) = \frac{1}{\sum_{b \in B(c)}(L(b))}$. A door is bordered with narrow bottlenecks.

On the basis of those three criteria, the 'door likelihood' function (DL) is defined as follow: $DL(c) = C_1(c) * (1 + C_2(c)) * C_3(c)$. This function tends to return low values for cells belonging to rooms and high values for cells defining doorsteps. To separate doors and rooms, we compute the mean value of the DL function applied to each cell of the topological map. Cells having a DL value greater than the mean value are tagged *door*, other ones are tagged *room*. The

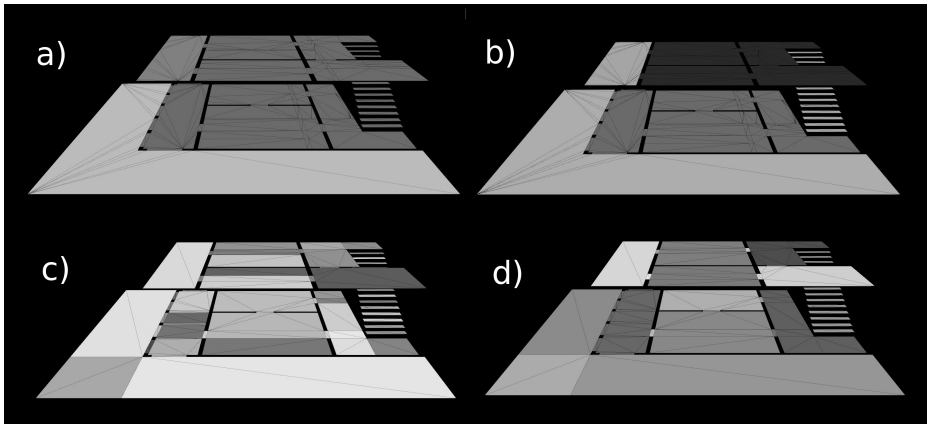

Fig. 3. Decomposition steps : a) buildings decomposition, b) floors decomposition, c) convex cells decomposition d) rooms decomposition

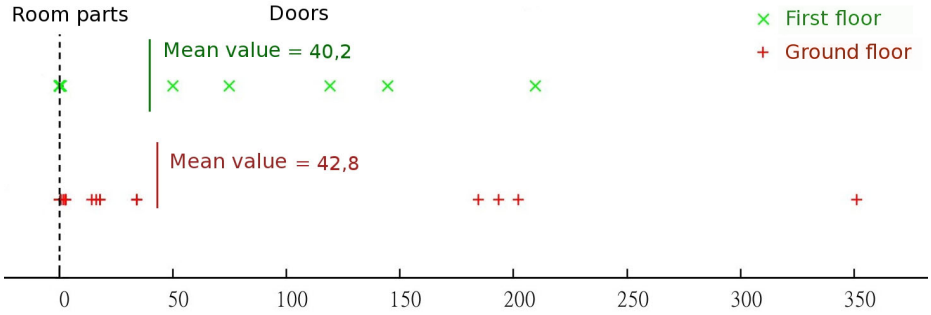

Fig. 4. Repartition of door likelihood values for covered convex zones

figure 4 shows the values computed for the two floors of our house example and the resulting environment decomposition is depicted in figure 3(d).

3.3 Informed hierarchical topological graph

In the previous section we described our environment analysis and tagging. Based on this analysis, a set of topological maps has been extracted and cells of those topological maps are tagged in several ways: covered, uncovered, room, door and step. On this basis, we create an informed hierarchical topological graph. Nodes of this graph are cells of the topological maps or group of cells. Edges represent accessibility between cells or zones. The hierarchy associated to the graph contains four levels leading to a natural description of the environment structure:

– The level four contains all cells of the topological maps (nodes) and their accessibility relations (edges).

- The level three groups mutually accessible cells having the same tags and belonging to the same topological map. Therefore, in this level, each step of a stair, each room, each doorstep is identified by a unique node. At this level, we are also able to distinguish covered exterior zone and indoor environments. A room is retagged *covered exterior* if its borders are mainly tagged *surmountable* or *step* and if it is mainly connected to uncovered zones. More generally, at this level, a set of rules can be defined and used to better qualify zones' nature. For instance, a rule can be used to qualify a corridor as a room surrounded by at least three doors.
- The level two groups steps into stairs, rooms and doors into floors.
- The level one groups floors and stairs into buildings, uncovered zones into exterior zones and covered exterior zones.

We can observe that levels three, two and one lead to a natural representation of an environment. It automatically provides a meaningful representation that (1) distinguishes indoor and outdoor environments, (2) decomposes building into floors linked by stairs, (3) each floor being itself decomposed into room linked by doorsteps. as we will see in the next section, this representation can be used for enhancing path planning quality. For instance, we can plan paths favoring the navigation in outdoor environments, while avoiding entering building if not needed. The hierarchical representation can also be used to enhance path planning performances by identifying high level paths that can be refined when needed.

4 Multi-criteria path planning

Navigation is subject to multiple constraints of various importances. For instance, one would want to minimize a path length while avoiding entering an unknown building and favoring navigation in covered regions if it is raining. A path corresponding to such a request is a compromise between several heterogeneous criteria. In this section, we propose a multi-critera path planning algorithm that makes intensive use of our environment representation. This algorithm uses a zone matching function that evaluates the adequacy of a zone given a multi-criteria request.

4.1 Zone matching

In many cases, finding a zone or a set of zones that exactly satisfy all criteria included in a request is not possible. Therefore, the first step toward multi-criteria path planning is to evaluate the distance between a request which characterize an ideal zone and a zone extracted from the environment. Criteria used in the request can be of different nature: maximizing a given value, favoring zones having some given properties. To evaluate the satisfaction of a criterion, we use several kinds of functions that returns values in [0;1] interval, 0 meaning the criterion is not satisfied and 1 meaning the criterion is satisfied:

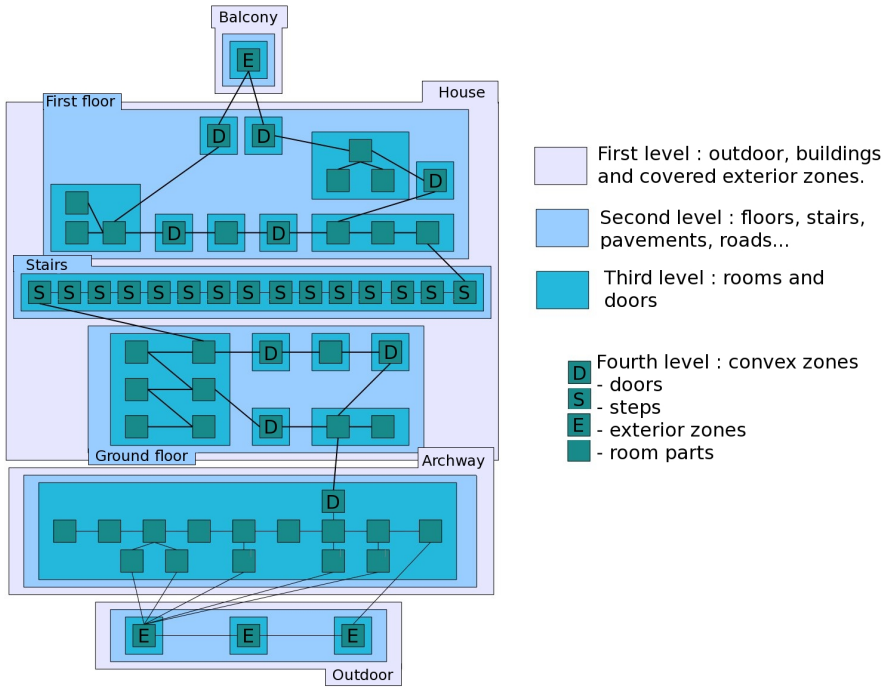

Fig. 5. Picturing of generated topological graph for our example environment

- For criteria relating to zones' nature (covered, does not belongs to a building for instance) we use a step function returning 1 if the criterion is satisfied and 0 otherwise (Cf. Fig. 6(a)).
- For a criteria concerning comparison between numerical values (zone's surface should be over 9 square meters for instance) we also use a step function (Cf. Fig. 6(b)).
- For criteria concerning values that should be minimized or maximized (ceiling height should be the highest possible) we use a linear function which is normalized using the maximum possible value in the current environment (Cf. Fig. 6(c)).
- For criteria concerning values that should be closed to a given value (zone's surface should be around 12 square meters) we use a gaussian-shaped function returning a maximum of 1 for the wished value (Cf. Fig. 6(d)).

In our system, a request is a conjunction of criteria. To take into account the importance of a given criterion, a weight w_c is associated to each criterion c. Given that the valuation of criterion c on a zone z is evaluated by the function $V_c(z)$, we define a function $D(z, R)$ giving the distance between the considered zone and the ideal zone described by the request R as follow:

$$D(z, R) = \sum_{c \in R} ((1 - V_c(z)) * w_c)$$

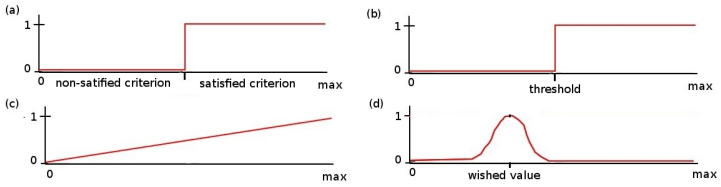

Fig. 6. Normalizing functions for a) criteria on zones nature, b) numeric values to compare to a threshold, c) numeric value to maximise, d) wished value to get near of

This function returns a value in interval $[0; \sum_{c \in R} w_c]$, 0 being the value associated to a zone perfectly satisfying the request. This function can be used for several requests purposes. For instance, it can be used to select a set of zones that maximize similarities with the ideal zone described by the request. For instance, one could want to select the nearest covered zone in order to avoid rain. Moreover, as we will see in the next section, it can be used during a path planning process to bias generated paths.

4.2 Multi-criteria path planning

The aim of our path planning process is to find a path going through zones that tend to match a multi-criteria request. In the previous section, we explained how we defined a distance function $D(z, R)$ evaluating the distance between the ideal zone described by the request R and the currently explored zone z. This distance function is used in a cost function that evaluates the cost of a path. Let $C(P, R, I_d)$ be the function evaluating the cost of path P in function of planning request R and giving importance $I_d \in [1; \infty]$ to the path length criterion. Let $L(P, z)$ be the length of the path P traversing the zone z. The function $C(P, R, I_d)$ is defined as follow:

$$C(P, R, I_d) = \sum_{z \in P} (L(z, P) * (I_d + D(z, R)))$$

In this function, the length of the path going through a zone z is multiplied by the sum of two factors: I_d and D(z,R). As D(z,R) is minimal for zones matching request R, paths traversing zones that match request R are encouraged. On the other hand, the I_d factor tends to select shortest paths for high values or paths matching request R for small values.

To plan a path inside our virtual environment, we use a Dijkstra algorithm. This algorithm is ran on the level four of our informed hierarchical topological graph and $C(P, R, I_d)$ is used to evaluate the cost of going through a cell z of the topological maps. Currently, as we have no hypothesis on the requests nature, the algorithm does not use the hierarchical representation to increase path planning performances but only to increase the quality of generated paths.

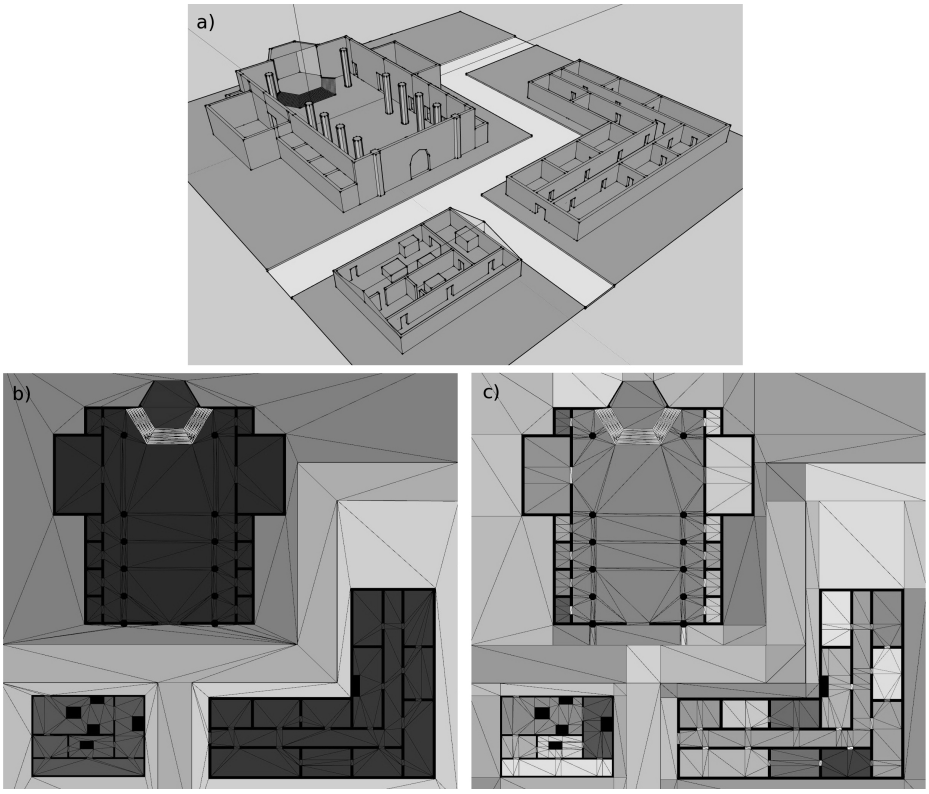

Fig. 7. a) A more complex example environment (roofs masked for better visibility) b) Its computed floor decomposition c) its computed room decomposition

5 Results

We tested our spatial analysis technique on a complex example depicted Fig. 7(a). The environment contains a road with sidewalks, a church and two houses. The church has heterogeneous rooms size and doorstep dimensions. It also contains pillars in the body of the church and unusual stairs leading to the pulpit. The building on the right of the figure contains a long and narrow corridor exhibiting numerous bottlenecks. The house on the left contains doorsteps of different width and height as well as a step roof and obstacles in some rooms. As presented in Fig. 7(b) and (c), buildings have been identified as well as stairs, doorsteps and rooms. This demonstrates the robustness of our technique which has been able to identify all relevant information despite potential interferences induced by obstacles, pillars or irregular doorsteps width and height.

To test path planning and related queries, we set up an environment with covered and uncovered exterior zones. This environment, depicted in Fig. 8, is used to test the behavior of our algorithm in the rainy day example. The planning

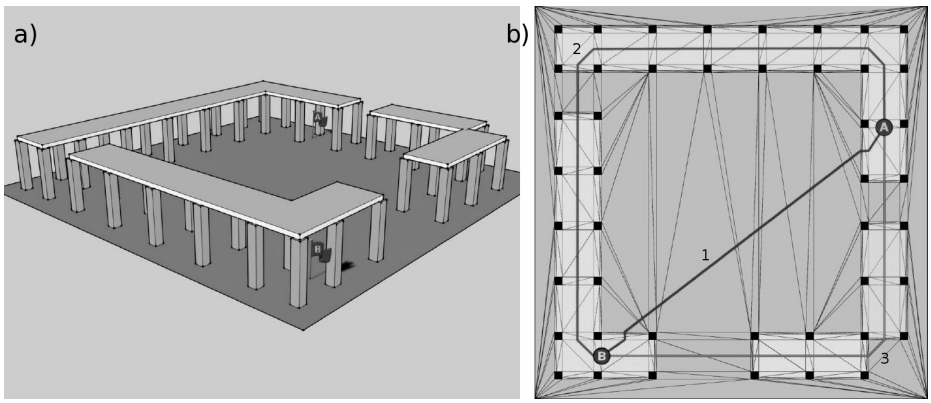

Fig. 8. Environment used for spatial reasoning test and path calculated in it out of three different requests.

request is set up by giving a constraint on navigating in covered zones (to avoid the rain) and changing weights associated to the covered and path length criteria. Computed paths are presented Fig. 8(b). Path 1 has been generated by giving a weight of 1 to the covered criterion and a weight of 2 to the path length. As expected, the shortest path is used. Path 2 has been generated by giving a weight of 2 to the covered criterion and 1 to the distance. In such a case, our character uses the longest path but this path minimizes the exposition to rain. Finally, path 3 has been generated by giving an equal value of 1 to the covered criterion and the path length. Our character navigates mostly in covered zones and is also exposed to rain. This example demonstrate that our algorithm can help characters to exhibit more natural behaviors by taking into account the environment nature.

6 Conclusion

We proposed an original method extracting an abstract, meaningful and hierarchical representation out of a 3D geometric environment. Our technique uses a spatial analysis algorithm that automatically detects the real structure (in terms of rooms, doorsteps, floors, stairs...) of the environment described by a 3D database. We showed that this structuring can be used to increase path planning quality and credibility through the use of multi-criteria requests. Our spatial analysis can be extended by adding specific rules to enhance the tagging process by taking into account some architectural standards for instance. However, we believe that an automatic process cannot deduce all information. For instance, the function of a room (kitchen, bedroom...) cannot be determined without analyzing its associated furniture. Nevertheless, if more precise information is needed, our method could be used to assist manual tagging by

pre-identifying specific zones and thus decreasing the effort needed to manually inform virtual environments.

References

1. Thomas, G., Donikian, S.: Modeling virtual cities dedicated to behavioral animation. Computer Graphics Forum (Proc. of Eurographics) **19** (2000)
2. Turner, A., Doxa, M., O'Sullivan, D., Penn, A.: From isovists to visibility graphs : a methodology for the analysis of architectural space. Planning and Design **28** (2001)
3. Latombe, J.: Robot motion planning. Kluwer Academic Publishers, Boston, MA (1991)
4. Arikan, O., Chenney, S., Forsyth, D.A.: Efficient multi-agent path planning. In: Computer Animation and Simulation 01. (2001)
5. Bayazit, O.B., Lien, J.M., Amato, N.M.: Roadmap-based flocking for complex environments. In: Pacific Conference on Computer Graphics and Applications. (2002)
6. Geraerts, R., Overmars, M.H.: Creating high-quality roadmaps for motion planning in virtual environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (2006)
7. Shao, W., Terzopoulos, D.: Environmental modeling for autonomous virtual pedestrians. In: Digital Human Modeling for Design and Engineering Symposium. (2005)
8. Pettre, J., Laumond, J.P., Thalmann, D.: A navigation graph for real-time crowd animation on multilayered and uneven terrain. In: V-Crowds. (2006)
9. Kallmann, M., Bieri, H., Thalmann, D.: Fully dynamic constrained delaunay triangulations. Geometric Modelling for Scientific Visualization (2003)
10. Mekni, M.: Hierarchical path planning for situated agents in informed virtual geographic environments. In: 3rd International Conference on simulation Tools and Techniques (SIMUTools). (2010)
11. Lamarche, F., Donikian, S.: Crowd of virtual humans : a new approach for real time navigation in complex and structured environments. Computer Graphics Forum (Proc. of Eurographics) **23**(3) (2004)
12. Kallmann, M.: Navigation queries from triangular meshes. In: The Third International Conference on Motion in Games (MIG). (2010)
13. Demyen, D., buro, M.: Efficient triangulation-based pathfinding. In: National Conference on Artificial Intelligence - AAAI. (2006)
14. Paris, S., Donikian, S., Bonvalet, N.: Environmental abstraction and path planning techniques for realistic crowd simulation. In: Computer Agents and Social Agents. (2006)
15. Farenc, N., Boulic, R., D.Thalmann: Informed environement dedicated to the simulation of virtual humans in urban context. Computer Graphics Forum (Proc. of Eurographics) **18** (1999)
16. Jiang, H., Xu, W., Mao, T., Li, C., Xia, S., Wang, Z.: A semantic environment model for crowd simulation in multilayered complex environment. In: the 16th ACM Symposium on Virtual Reality Software and Technology. (2009)
17. Mekni, M.: Automated Generation of Geometrically-Precise and Semantically-Informed Virtual Geographic Environments Populated with Spatially-Reasoning Agents. PhD thesis, Université Laval (2010)
18. Lamarche, F.: Topoplan : a topological path planner for real time human navigation under floor and ceiling constraints. Computer Graphics Forum **28**(2) (2009)