A topological map for scheduled navigation in a hospital environment

Cristina Urdiales[1], Alberto Poncela[1], Roberta Annicchiarico[2], F. Rizzi[2], Francisco Sandoval[1] and C. Caltagirone[23]

[1] Dpto Tecnologia Electronica, ETSI Telecomunicacion, Campus de Teatinos s/n, University of Malaga, 29071 Malaga, Spain
cristina@dte.uma.es
[2] Fondazione IRCCS Santa Lucia, Via Ardeatina 306, 00179 Roma, Italy
[3] Universita Tor Vergata, Roma, Italy

Abstract. We present a new scheduler for navigation in structured environments. To achieve efficient navigation planning, a model of the environment supporting fast processing is required. We hierachically extract a topological map from a plot by using a new 3D data structure where nodes correspond to significant places in the environment. Paths linking this nodes are also hierarchically calculated. A TSP algorithm uses this topological map to efficiently schedule navigation tasks.

1 Introduction

Hospital Management is a hard task due to the complexity of the organization, the costly infrastructure, the specialized services offered to different patients and the need for prompt reaction to emergencies. Artificial Intelligence planning and scheduling methods can offer substantial support to the management of the hospital, and help raising the standards of service. Management, as pointed out in [4], is particularly important in a Rehabilitation hospital for a person with disabilities who - due to different pathologies - is no longer able to independently provide to his own self-care, and who needs the support of a second person to perform even the simplest every-day activities, referred to as basic Activities of Daily Living (ADLs).

To develop their work, the hospital member staff has to practice its daily assistance activity operating in wide spaces, thought to let proper movements to patients moving on wheel chairs, never forgetting their different planned schedule. In fact, during the whole day, the patients run through different peculiar activities, such as attending to the gymnasium for motory rehabilitation or into other laboratories for different recovery activities (logopedics, cognitive therapy, phoniatrics). Each floor has its own gymnasium and laboratories for his own rehabilitation goals. Preparing scheduling for physicians and nurses in the Rehabilitation department is a complex task, which requires taking into account a large number of rules, related to various aspects: the number of patients, the actual location of the patient related to the bedroom, the gymnasium, the priority of a patient compared to another, etc.

Rudomín, J. Vázquez-Salceda, J. Díaz de León (Eds.). e-Health: Application of Computing Science in medicine and Health

A typical problem in such an environment where a staff member must accomplish several tasks at different places is in which order those places must be visited to work in an efficient way. Naturally, time and experience provide a valuable knowledge to the staff to organize their activity but unexpected emergencies or priorization make it difficult to follow the same pattern on a regular basis. In these cases, it would be useful to have a scheduling mechanism capable of providing the best plan to achieve the most optimal path through all required places as fast as possible. Also, the introduction of new mobile technologies like autonomous wheelchairs makes it necessary to develop such a mechanism to avoid constant reprogramming.

In its simplest formulation, the aforementioned scheduling problem can be briefed as, given a set of places to visit and the spatial relationship among them, finding in which order they must be visited to achieve the maximum possible efficiency. This is obviously equivalent to the well known Travelling Salesperson Problem (TSP). It must be noted, though, that the TSP does not provide information about how the agent, human or not, may reach each place from the previous one. The intention of reaching a goal point from a departure location while avoiding obstacles in the way is known as path planning. Path planning is required if the agent is mechanical so that it can be fed with the chain of movements required to navigate in the environment. Even if the agent is a human, it is also recommendable to solve navigation in case s/he is not completely familiar with the environment, it is in a hurry or an unexpected change, like a closed door, happens.

In this paper, a fast scheduling method for an efficient navigation in a dynamic structured environment is going to be presented. Whenever an environment is known, it is useful to use a model of such an environment for efficient problem solving. However, it must be noted that the TSP is a well known NP-complete problem and path planning is also very dependant on the size of the problem instance. Hence, an automatic map building method to extract a compact model of the environment from any available representation like a plot is presented in section 2. Using this representation, a TSP based scheduling method plus a path planning technique are presented in section 3. A real test environment is described in section 4. Experiments and results are presented in section 5. Finally, section 6 presents conclusions and future work.

2 A spatial representation for navigation

Spatial representations have been traditionally used for efficient navigation. Such representations must present free and occupied space as well as the relationship between the represented places so that paths can be inferred. The most classic representation is, obviously, the printed map. However, these representations have evolutionated to allow its processing by computers. Traditional spatial representations include metric and topological approaches.

Metric models of the environment rely on explicitly reproducing the geometry of the environment by using, for example, segment models [5], vertex models

[8], convex poligon models [3] or grids [12]. The main advantage of these representations is that they implicitly provide information about the relations among different places. Also, they can be extracted in a mildly fast way from a conventional printed map and some of them can also be easily generated by a mobile platform equipped with range sensors. Their main drawback is that their data volume may depend strongly on the complexity of the environment and, hence, the processing time of algorithms relying on them can not be predicted. Also, if maps are generated by a mobile platform, mechanical and sensor errors make it difficult to preserve accuracy in medium and large scale space.

Alternatively, topological approaches represent the environment by means of graphs, where each node is a place and the weights of the links joining two nodes are related to factors like the distance between them or the difficulty to travel from one to the other. There are many methods to create topological maps. Early algorithms relied on processing digitalized printed maps. Objects, modelled by polyhedra, were used to split space into a limited number of regions corresponding to rooms, doors or any other significant element. More recent approaches focused on building such maps by inserting a node each time a distant is travelled or each time a significant geometric beacon (e.g. corridor intersections) is detected in the environment. This process can be performed either a priori by a human observer or by a more or less complex automatic algorithm. Topological maps are reported to be more directly suited to problem solving [11] and more resistant to errors in mapping information. However, they are difficult to build when no layout information is known a priori and the information they provide about the relationship among nodes is usually vague. Some approaches combine metrical and topological representations. Zelinski [21] relies on partitioning a metric grid by means of quadtrees, but the resulting topological representation strongly depend on the environment layout and it is typically very suboptimal. Arleo *et al.* [1] model obstacle boundaries by means of straight lines, but they can not deal with irregular obstacles or wall which are not parallel or orthogonal. Thrun *et al* [17] split a metrical grid into regions by means of Voronoi diagrams, but it has been reported [9] that these maps, as well as other self organizing ones like the colored Kohonen map or the growing neural gas provide unintuitive tesellations. Kraetzschmart *et al* [9] extract walls from a grid by using wall histograms, but they acknowledge that their method does not provide suitable topologies to navigate in free space.

2.1 Data structure generation

We propose a new method to extract a topological map from a digital print of a map. The proposed technique is based on the split and merge paradigm but rather than an uniform grid or a quadtree a hierarchical data structure is used to support the node extraction process. Our previous approaches [18] to this process relied on the adaptive relinking paradigm [2], but in those cases it was not granted that resulting nodes were related to connected regions. To overcome this drawback, a new structure known as uncomplete pyramid is proposed. Given a digital 2D map of the environment where free and occupied nodes are printed in different colors, say white and black respectively, this map becomes level 0 of the proposed structure, which is generated as follows:

1. Hierarchical structure generation. The map becomes the base of a pyramidal structure. Each level l of this pyramid is a reduced map with 1/4 of the cells of the level immediately below. Each pyramid cell (x,y,l) has five associated parameters:
   - Homogeneity, $H(x,y,l)$. $H(x,y,l)$ is set to 1 if the four cells immediately underneath have the same occupancy probability and their homogeneity values are equal to 1. Otherwise, it is set to 0.
   - Occupancy probability, $P(x,y,l)$. If the cell is homogeneous, $P(x,y,l)$ is equal to the occupancy probability value of any of the four cells immediately underneath. If the cell is not homogeneous, the value of $P(x,y,l)$ is set to a fixed value (c_{NH}).
   - Area, $A(x,y,l)$. It is equal to the addition of the areas of the four cells immediately underneath.
   - Parent link, $(X,Y)_{(x,y,l)}$. If $H(x,y,l)$ is equal to 1, the values of parent link of the four cells immediately underneath are set to (x,y). Otherwise, these four parent links are set to a null value.
   - Centroid, $C(x,y,l)$. It is the centre of mass of the base region associated to (x,y,l).

   After the generation step, remaining nodes present an homogeneity value equal to 1 and provide a quadtree-like discretization. Thus, the complexity of this decomposition is not directly related to the world complexity but to the position of the obstacles.
2. Homogeneous cells fusion. In this step, the algorithm tries to link cells whose parent link values are null. Basically, these cells, (x,y,l), are linked to parents of neighbours cells, $(x_p,y_p,l+1)$, if the following conditions are true:
   - $H(x,y,l) = 1$ & $H(x_p,y_p,l+1) = 1$
   - $P(x,y,l) = P(x_p,y_p,l+1)$
   - $\|C(x,y,l) - C(x_p,y_p,l+1)\|_2 < DistMax$,

   being $DistMax$ a threshold that fixes the maximum dispersion of the regions at the base.
3. Homogeneous cells classification. Two neighbour cells, (x_1,y_1,l) and (x_2,y_2,l), are fused if the following conditions are true:
   - $(X,Y)_{(x_1,y_1,l)} = NULL$
   - $(X,Y)_{(x_2,y_2,l)} = NULL$
   - $H(x_1,y_1,l) = 1$ & $H(x_2,y_2,l) = 1$
   - $P(x_1,y_1,l) = P(x_2,y_2,l)$
   - $\|C(x_1,y_1,l) - C(x_2,y_2,l)\|_2 < DistMax$

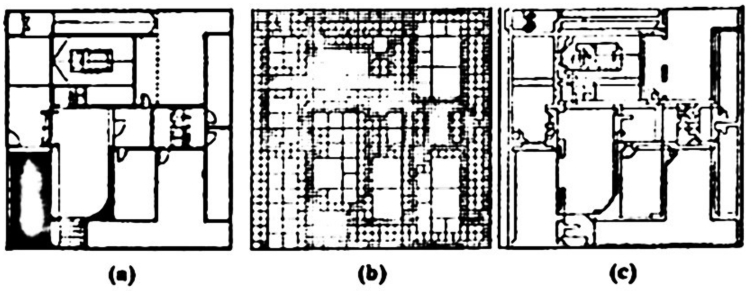
When this top-down relinking step is finished, regions linked to a node no longer depend on the layout of the environment. It can be observed that this process only depends on threshold $DistMax$, which is used to select the maximum size of a node. For low $DistMaxs$, nodes tend to correspond to large areas. Otherwise, they are related to small areas. Despite this threshold, complicated areas packed with obstacles tend to be related to several nodes rather than to a single one. This is useful to plan paths between pair of nodes because planning is hierarchically decomposed into shorter and easier paths.

It can be noted that this process relies on the split and merge paradigm, but it is significantly faster than conventional 2D split and merge approaches because of the hierarchical nature of the data structure. Fig. 1 shows how nodes are arranged during the bottom-up split stage (Fig. 1.b) and after the top-down merge stage (Fig. 1.c). It can be observed that resulting regions roughly correspond to rooms. Smaller regions are related to occupied areas and, hence, removed from the structure, where only free nodes are allowed.

2.2 Topological map generation

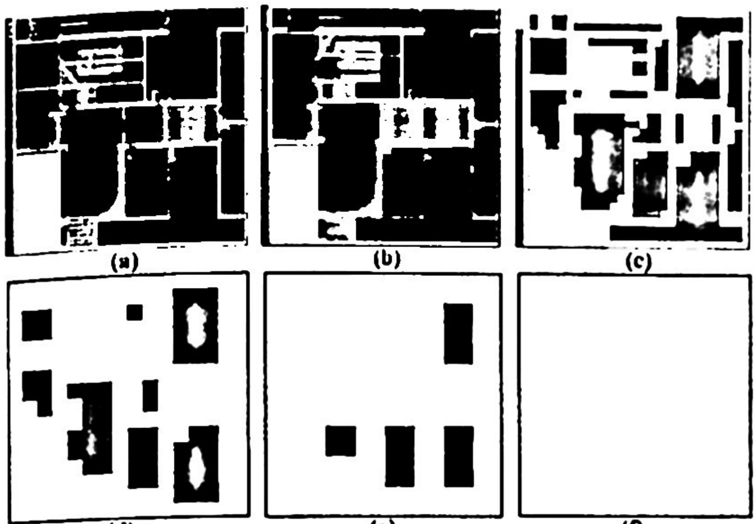
Since homogeneous regions linked to nodes in the proposed structure are connected and roughly correspond to structures like rooms and corridors, such nodes can be used to build a graph. Figs. 2.a-f show levels 1 to 6 of the generated pyramid for the base in Fig. 1.a. Free nodes in those levels are painted in black and the white gaps in the levels correspond to areas where there are no nodes. It can be observed in Fig. 2.f that no nodes are available in levels 6 and above. Only the largest regions in the map are defined at level 5 (Fig. 2.e). Then, the garage and the rooms on the left side of the map are defined at level 4 (Fig. 2.d). Smaller rooms and corridors are mostly defined at level 3 (Fig. 2.c), while minor details appear in lower levels. In order to build a topological map using this structure, nodes in the graph correspond to parentless nodes in the 3D structure. A parentless node in this structure corresponds to a region which is not included in any larger region. For example, the three rooms in the bottom of the map are defined at level 5 (Fig. 2.e), even though they also appear in levels 0 to 4.

Despite the correct distribution of free space, the proposed representation is not valid to build a topological map because obstacle areas corresponding to doors are painted in black in the original map and, hence, assumed to be obstacles. Thus, in the topological representation it would be assumed that a person can not move between two rooms connected by a door because there is no free path between them. Naturally, the map can be manually preprocessed to avoid this problem, but it is easier to use simple image preprocessing to achieve the same results. Fig. 3.a shows the original input map after a gaussian blur is applied. While large obstacles are still clearly dark, thin lines tend to blur to a light grey instead. Then, if a histogram stretching for contrast enhancement is



(a) (b) (c)

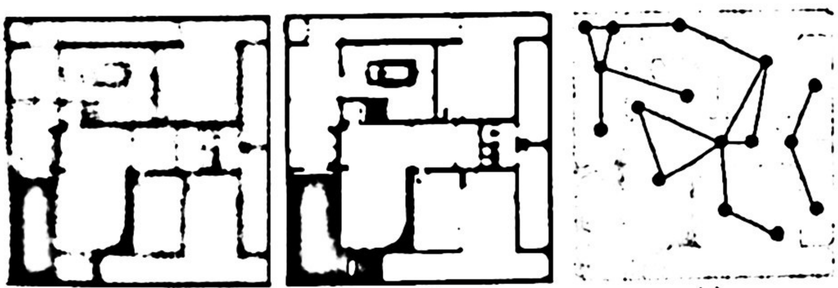
Fig. 1. Structure generation: a) original map; b) hierarchical split; b) hierarchical merge.



(a) (b) (c)
(d) (e) (f)

Fig. 2. Map structure: a-f) levels 1 to 6 of the map.

applied, dark areas go black and light areas go white (Fig. 3.b). If the topological map is extracted from Fig. 3.b, doors and thin lines are removed from the base of the structure and connected rooms become linked nodes (Fig. 3.c). It is necessary to note, though, that if a wall is represented by means of a thin line for any reason, it is removed from the map as well, as can be appreciated in the bottom part of Fig. 3.c. Similarly, regions presenting a high density of lines, like the stairs on the right of Fig. 3.c, are also identified as obstacles by the process. This problem can only be solved by correcting the map after generation or by removing such areas by means of texture detection techniques.



(a) (b) (c)

Fig. 3. Map post processing and graph calculation: a) gaussian blur; b) histogram stretching; c) graph calculation.

When the set of nodes in the topological map is available, it is also necessary to build links between them. It can be observed that the proposed data structure implicitly preserves connectivity information: two nodes related to regions in contact at the base must be linked. In order to calculate the weight of the links, the euclidean distance between the centroids of the regions linked to each pair of nodes are used. This might not be representative if regions could present completely random shapes and, hence, the distance to travel from one to another depended largely on the departure and arrival points, but in the proposed structure regions presenting strange shapes are decomposed into several nodes rather than linked to a single one. It can be noticed that, given a printed map, a person could build a topological map similar to the proposed one fairly fast. The average time to extract a topological map using the proposed technique is 0.45 s. using a conventional Pentium III. However, processing time is not the main advantage of the proposed technique. Its main advantage is that the obtained map is grounded to the metrical map and implicitly adapted to its geometry. Thus, it is extremely easy and fast to calculate paths at low level using the proposed structure, as explained in the next section.

3 Scheduling and navigation

Once a model of the environment is available, scheduling can be performed by simply computing in which order nodes need to be visited. To obtain a complete plan, it is also necessary to know where the mobile is at a given instant, so that the path starts at that position. If the mobile is a human, it can give its approximate position to the system. Otherwise, such a position must be estimated either by active landmark localization (e.g. [7]) or, if the mobile is mechanical, by odometry based localization methods (e.g. [15]). In any case, when nodes to visit are selected, scheduling can be formulated as a TSP: given a finite set of nodes $N = c_1, c_2, ...c_n$ and a distance $d(c_i, c_j)$ for each pair of nodes, the TSP consists of searching for a tour of minimum length to visit all nodes once. N does not include all nodes in the topological map, but only those corresponding to places that need to be visited. Sometimes, the order to visit some places can not be altered (e.g. go to a room and take a patient to the gym). In these cases, we deal with a simple path planning algorithm. In other cases, several rooms must be visited before performing a certain tasks (e.g. collect the medical record of a patient and some equipment before a visit). In these cases, the TSP is used to get the correct visiting order. For n places to visit there are $\frac{(n-1)!}{2}$ tours and, hence, the complexity of the TSP grows very fast with n. However, an exact solution can be calculated when n is low. In our case, n is usually low, so the TSP can be solved exhaustively and time responses at topological level, lower than a second, are more than suitable to operate in real time. Thus, plans can be altered even when an agent is moving. It is necessary to note that, at least in emergency situations, we deal with more than one person. In those cases, the algorithm pre-plans the best routes for all of them so that all important places are visited once: initially the TSP is ran for all persons to visit every room. Using these solutions, we check who reaches any of the goals first, say person P_1. After this, the plan of P_1 is preserved, but the TSP is run for the other persons after removing the goal reached by P_1. It is important to note that this is done a priori, when no one has started to move yet. This operation is performed as many times as necessary until everyone reaches the final goal.

Thus, it is granted that each node is visited only once and tasks are divided into the whole staff or a part of it.

In cases where the mobile is a robotic agent or a person who is not familiar with the environment providing the order in which places should be visited might not be enough. In these cases, it might also be interesting to provide the best path to move between the current position and the next goal at any time. It is important to note that, to solve this problem, the current position of the mobile in the environment is required and it must be either calculated by any of the localization procedures described above or provided by the agent when requested. An efficient path to a given goal can be calculated as follows:

- Calculate which node in the graph the departure point is linked to. This can be done in a straight way because such an information is preserved in the pyramid link structure.
- Calculate a path at topological level between nodes c_i and c_j by means of the well known A* path planning algorithm, being c_i the node related to the current position of the agent and c_j the next destination node in the list returned by the scheduler. This algorithm returns a node path.
- Propagate the node path to base level using the link structure. At this level, the node path becomes a path region which includes the departure and arrival points.
- Using a potential fields approach [19], calculate an unique path at metric level between the departure and arrival points.

Fig. 4 shows an example of the proposed process. Fig. 4.a presents a simple map where a topological graph is over imposed. In this map, a pair of departure and goal points are marked with a "D" and a "G", respectively. After the A* algorithm is used, the path consists of the three nodes marked with a black circle, where the first one is related to "D" and the second one to "G". Fig. 4.b shows the path region after the path is propagated to base level. The potential field approach basically consists of considering obstacles as repulsors and goals as attractors. The direction to follow to the goal at each point of the trajectory is a combination of all forces involved and the final path to the goal is a line of minimum energy. In our case, the path must be included inside the path region, so the boundaries of the path region are considered repulsors and the goal is an attractor. The final path is printed in white in Fig. 4.b and the gradient of the potential field is represented in shades of grey.

4 Test environment

In this work, the mobile agent is a member of the staff of a hospital environment, more specifically the Fondazione IRCCS Santa Lucia, in Rome. The department organization needs the presence of 6 physicians that provide the attendance to 50 patients, 25 beds each 3 physicians, and a chief, plus 6 nurses, divided as well in two halves, and including also a matron and assistant. During the first hours, until 2:00 p.m., the medical staff is almost complete, with the exception

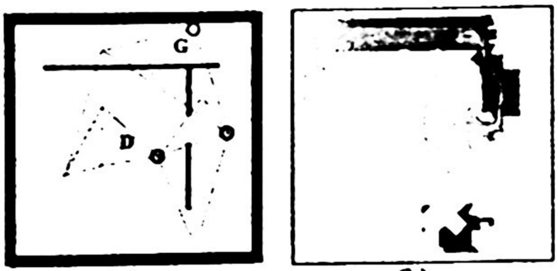


Fig. 4. Hierarchical path planning algorithm

for eventual night services and vacations. From 2:00 to 8:00 p.m. there's only one physician in each department. Finally, during the night there is only one physician in the whole hospital, that includes six departments. Obviously, physicians and nurses dues differ and, of course, every plan is changed in case they have to face an emergency. It can be noted that the squad is not wide and, hence, a better organization and optimization of the resources is required.

In the Fondazione IRCCS Santa Lucia, a standart day consists of the following. Nurses each morning have got to attend the patients considering their needs (wash, change clothes and "catheters", administrating therapy, get vital parameters as blood pressure, body temperature and cardiac frequency, cure pressure ulcers) and, in the end, get the patients up and ready to go to the gymnasium. Of course, this activity must follow a schedule, and the first patients to be ready are the ones first beginning the motory therapy. Physicians visit the patients every morning, after they have been informed about eventual problems occurred during the night, and if this ever happens, this leads to a change to the normal plan. During their visit, physicians can change the therapy and ask for eventual specialist consult for each patient. Every week, each patient has got to be observed by the physician during his training in the gymnasium. Around 12:00, nurses feed patients who are unable to do it self-dependently and administrate drugs. After 2.00 p.m., as aforementioned, there is only one physician in the whole department. He checks blood analysis, eventually change therapies and deals eventual emergencies. At 4:00 p.m. nurses administrate drugs. At 6:00 p.m. nurses feed patients who are unable to do it self-dependently and administrate drugs at 8:00 p.m. In the end, they must attend the patients one more time (wash, change clothes and catheters, administrate therapy, get vital parameters as blood pressure, body temperature and cardiac frequency, care pressure ulcers) and, after this, get the patients ready to go to bed for the night. The last check is at 10:00 p.m.. Then, during the night, they go through an ordinary control monitoring.

When an emergency occurs, the schedules of the whole staff are completely changed. The whole physicians staff comes to the patient room to attend the patient and, to attend its duty, needs to:

- Get there as soon as possible
- Have at least two nurses with him

- Have the emergency trolley and the defribillator near
- Have the oxygen bottle
- Have bandage and everything needed to face eventual injury
- Have a electrocardiograph, sphyigmometer and saturimeter
- Have the patient medical record containing all his information, because he might not be familiar with that specific patient if he/she is not his personal physician during the afternoon or the night attendance.

Each of the aforementioned elements required by the physician to face the emergency are located in different places, and can even be moved during the day. Specifically, there is only one emergency trolley and defibrillator in each department, located in one of the two medical rooms (usually predefined) unless it's been recently used in the other half of the department. The oxygen bottle is placed in the medical room as well. Bandages are placed in both medical rooms. Sphyigmometer is placed in both rooms. There is only one electrocardiograph and saturimeter in each department, located in one of the two medical rooms (usually predefined) unless it's been recently used in the other half of the department. The medical records are placed in the physician's room during the morning and in the nurses room during the afternoon and during the night.

Naturally, such a complex system works well when the medical and the nurses staff has worked together for a long time, when everything is placed in the right place, and when there are no critic situations such as double emergencies. It is also inmediate to realize that the staff can go through different troubles due to "ambient obstacles" and to the fact that physicians and nurses may not know very well where to move rather than how to move. All this leads to a waste of resources that could be better used and distributed during an ordinary day and to less speed and efficiency in the attendance to a patient during an emergency. This delay, even if it does not worse his prognosis, certainly makes patients suffer and provoke anxiety. In the end, a proper use of the stuff grants the patients a better care.

5 Experiments and results

This section includes several results of the proposed scheduling strategy. First, a topological map of the hospital is extracted from its plot as proposed (Fig. 5). This process takes 0.45 s. approximately. There are several important rooms in this plant: Medical Rooms 1 and 2 (MR1 and MR2), Nurses Rooms 1 and 2 (NR1 and NR2), the Physician Room (PR), the gym, the dining room (DR) and rooms 301 to 327 (R301-R327). As previously commented, some equipment is located only in one of the medical rooms, in these tests MR1.

During an ordinary day, the matron decides in which order the patients must visit the gym. This influences the order in which a given nurse, which has different patients assigned, must attend them and take them there. Leaving factors not related to navigation aside, in order to provide a suitable gym scheduling, the matron must take into account where the patient assigned to a nurse are to determine in which order they must be visited. In a first test, we have chosen

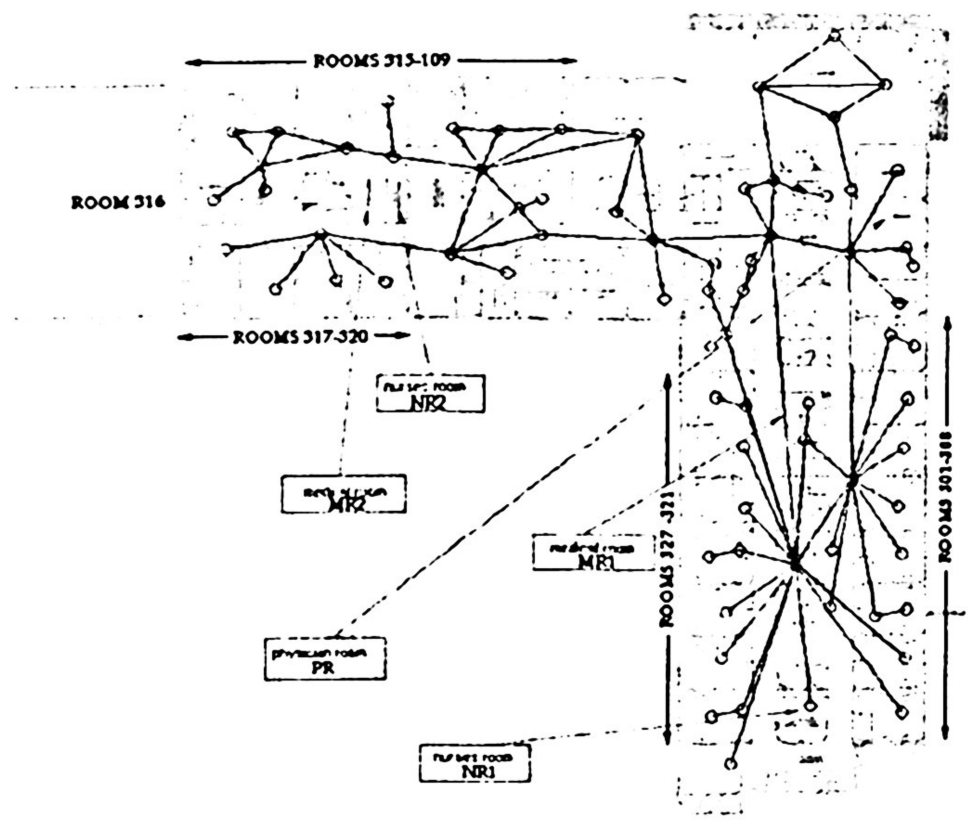


Fig. 5. Original map with its topological map overimposed

nurse who is assigned to patients in rooms R301, R308, R323 and R327. We assume that, originally, the nurse will be at NR1. In our experiment, she must go to each room, attend the patient there, take him to the gym, wait for him there, bring him back to his room and proceed to the next patient. Obviously, these basic actions may change to suit the structure of the hospital organization. After the TSP is used, the visiting order is R327, R323, R308 and R301. When the nurse reaches each room, she attends the patient and then, she receives the fastest route to the gym and back. Fig. 6.a shows a path to go the gym and back from R301. In this case, it would probably have been more optimal to use the east corridor, but heuristical planning algorithms do not grant the optimal solution. Similarly, Fig. 6.b shows a path to move from R308 to DR and back. In this case, the path does seem to be the shortest one. Finally, Fig. 6.c shows a monitoring visiting order for the night, where the proposed scheduling sequence used. Again, in this case paths seem to be qualitatively adequate.

It must be observed that the problem becomes more complex when there are more rooms to monitor per nurse and when additional criteria (e.g. time spent with a patient, nurses assisting other patients while one is busy in the gym, etc) need to be considered. In this paper, though, we only regard navigation factors. However, one of these factors is how to react when a planned path is blocked by unexpected factor. Fig. 7 shows a typical example of one of those

(a) (b) (c)

Fig. 6. Path planning during an ordinary day: a) path from R301 to the gym and back; b) path from R308 to DR and back; c) path to monitor R327, R323, R308 and R301 efficiently.

replanning problems. In the original topological map, the link marked in Fig. 7.a joining nodes in both corridors in the east wing of the plant existed. Then, a path running from one wing of the plant to the other went through that area (Fig. 7.b). However, in a new experiment, the link vanishes due to an ambient obstacle and both corridors are no longer connected at the wing junction. The recalculated topological map is very similar to the previous one, as expected, minus the aforementioned link. Thus, in this case, an alternative path joining both wings in an efficient way does necessarily run through the gym (Fig. 7.c). Figs. 7.b and c show both paths returned by the potential fields overimposed to the path regions returned by the high level planner. It is important to note that paths can be calculated as fast as 0.05 s average over the used map. Thus, as soon as the obstacle in the way is detected, an alternative path can be provided.

A more interesting test is when an emergency situation at night is presented. In this situation, there are only two nurses at R307 (Nurse1) and R324 (Nurse2) and a doctor (Doctor) at R314. Then, an emergency is detected at R312. The emergency trolley, defibrillator and other unique equipment is supposed to be at MR1. Medical records are at NR2 at night. The rest of the equipment is located both at MR1 and MR2. Thus, a complete route for a single person would include its position, medical room 1 (MR1), nurses room 2 (NR2) and room 312. Naturally, the TSP can return an exact solution with so few nodes. Results, presented in table 1, were to be expected. All distances are expresed in resolution units. For each run of the algorithm, places in the plans finally expected to be visited by a given member of the staff are bolded. Nurse1 is closer to MR1 and would reach it first (Run 1), so Doctor and Nurse2 do not even try that node. When we recalculate routes taking this fact into account (Run 2), it is obvious that the doctor is closer to NR2, where the records are. Thus, Nurse1 and Nurse2 do not try to visit NR2 and go to room 312 straightly. The final paths for the members of the staff are NR2, 312, MR1,312 and 312 for the doctor and nurses 1 and 2, respectively (Fig. 8). It can be checked that each room is visited only once, that at the end the doctor and 2 nurses are at room 312

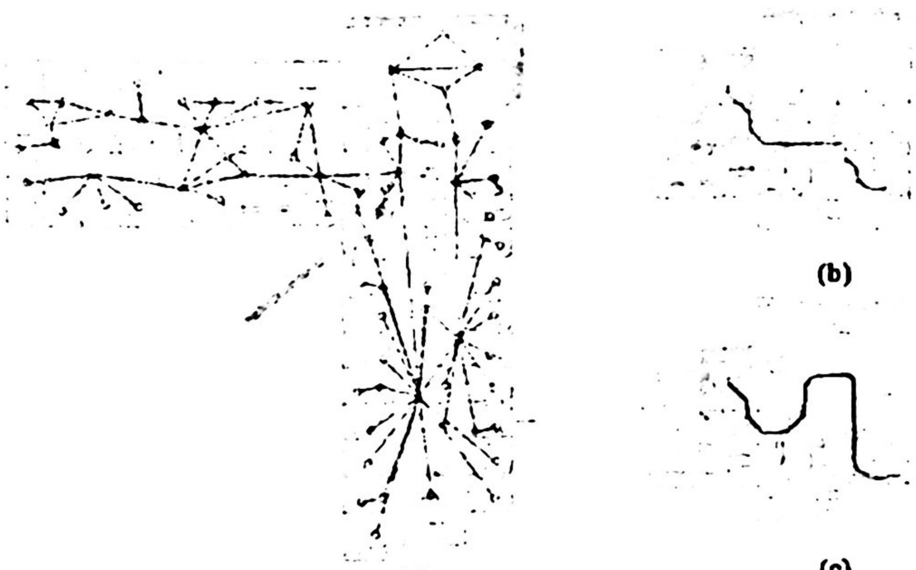


Fig. 7. Original map with its topological map overimposed

R312), as required and that the lengths of their paths, if not the best because of the heuristical nature of the path planning algorithm, are clearly optimized. is interesting to note that the length of the path for Nurse1 does not change much whether she has to visit NR2 or not. This happens because the fastest path to reach R312 runs through NR2 (Fig. 8.b). Test 2 deals with an additional problem: the equipment supposed to be in MR1 has in fact been moved to MR2 and, hence, when Nurse1 arrives there, she asks for a change of plans. The staff has started to move and Doctor and Nurse 2 are on their way to NR2 and R312 respectively. It can be observed in the table that they are redirected. Now the Doctor gathers all equipment, first in MR2 and then in NR2, and both nurses go straightly to R312. In this test, both nurses save the time of visiting MR2, which visited by the doctor instead. It needs to be noted that in this case, all people has already moved 86 units, required by Nurse1 to reach MR1. Hence, the real final distances are 310, 699 and 635 respectively. As expected, the doctor path is longer than before, because he has to change his plans and move to MR2 as well. However, the nurses' paths are quite similar because they are not supposed to go to MR2 and the rest of the plan, going to R312, matches their previous ones.

It can be observed that, for such a reduced number of nodes and using no high level constraints, the proposed problem is relatively easy to solve. However, the proposed data structure and planning algorithms are also suitable for larger problem instances and, so, they can provide the basis to solve complex problems as well.

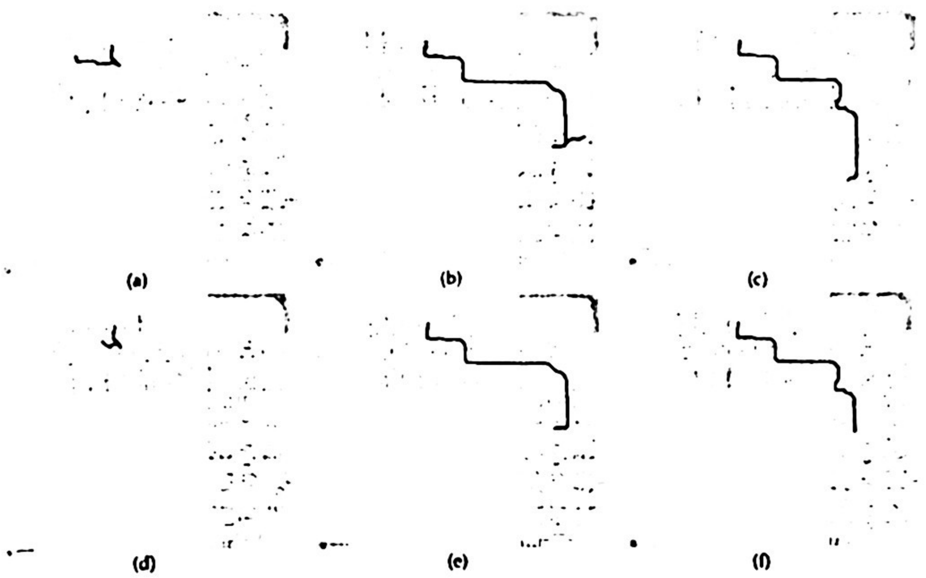


Fig. 8. Paths to attend emergency in room 312 for Dortor, Nurse1 and Nurse 2 in a-c) Test 1; and d-f) Test 2.

6 Conclusions and future work

This paper has presented a new data structure to extract a topological map from a hospital plant so that scheduling can be performed in a fast way. We have also presented a path planning technique to provide an efficient route to reach the goals in the schedule, so that members of the staff less familiarized with the environment can optimize their work. Both the topological map generation and the calculation of the paths are fast enough to adapt to unexpected obstacles and situations. Tests have been performed in a plant of the Fondazione IRCCS Santa Lucia, in Rome. Results have proven that fast scheduling can be achieved, that people can be efficiently coordinated in emergencies and that paths provided by the scheduler are realistic and efficient as well. However, it is assumed that a centralized scheduling system capable of communicating with each member of the staff is available. Under this premise, a person does not need to know where the others are, but simply follow the guidelines provided by the scheduler that include which places to visit, in which order and how to reach every one of them.

Future work will focus on improving interaction with other users for a better job coordination during an ordinary day. Besides, more environment intelligence is required to: i) determine if some rooms (e.g. medical or nurse rooms) can not be crossed to shorten a path; ii) determine if there is room for large objects (e.g. wheelchairs) to move through a place; iii) include in our calculations how difficult it is to cross a place (e.g. when it is crowded) to provide alternative routes; iv) take into account how many members of the staff are required to move the equipment so that rooms can be visited by several people rather than one (e.g. nurse1:move to medical room 1 and take the emergency trolley; nurse2:move to medical room 1 and take the defibrillator). Also, the proposed method will be adapted to an autonomous agent to build an intelligent wheelchair.

	Test 1	
	Run 1	Total Distance
Doctor	MR1 (675), NR2 (496), R312 (113)	1284
Nurse1	MR1 (86), NR2 (496), R312 (113))	695
Nurse2	MR1 (120), NR2 (496), R312 (113)	729
	Run 2	Total Distance
Doctor	NR2 (180), R312 (113)	293
Nurse1	MR1 (86), NR2 (496), R312 (113)	695
Nurse2	NR2 (522), R312 (113)	635
	Final Path	Total Distance
Doctor	NR2 (180) R312 (113)	293
Nurse1	MR1 (86) R312 (608)	694
Nurse2	R312 (634)	634
	Test 2	
	Run 2'	Total Distance
Doctor	MR2 (22), NR2 (89), R312(113)	224
Nurse1	NR2 (496), MR2 (89), R312 (57)	642
Nurse2	NR2 (433), MR2 (89) , R312 (57)	568
	Run 3'	Total Distance
Doctor	MR2 (22), NR2 (89), R312(113)	224
Nurse1	NR2 (496), R312 (113)	609
Nurse2	NR2 (433), R312 (113)	546
	Final Path	Total Distance
Doctor	MR2 (22), NR2 (89), R312(113)	224
Nurse1	R312 (608)	608
Nurse2	R312 (545)	545

Table 1. Places visited by the hospital staff plus distances during the described emergency when the equipment is where it was expected (Test 1) and otherwise (Test 2)

Acknowledgments

This work has been partially supported by the Spanish Ministerio de Ciencia y Tecnologia (MCYT) and FEDER funds, project No. TIC2001-1758. We would like to thank the Fondazione IRCCS Santa Lucia - Roma to let us use the hospital map.

References

1. Arleo, A., Milln, J.R. and Floreano, D., "Efficient learning of variable-resolution cognitive maps for autonomous indoor navigation", *IEEE Trans. on Robotics and Automation*, 15 (6), pp. 990-1000, 1999.
2. Burt, P., Hong, T. y Rosenfeld, A., "Segmentation and estimation on image region properties through cooperative hierarchical computation", *IEEE Trans. on Systems, Man and Cybernetics*, 11 (12), pp. 802-809, 1981.

3. Chatila, R. and Laumond, J., "Position referencing and consistent world modelling for mobile robots", *IEEE Proc. Int. Conf. Robotics and Automation*, pp. 138-170, 1985
4. Cortes, U., Annicchiarico, R., Vazquez-Salceda, J., Urdiales, C., Caamero, L., Lopez, M. and C. Caltagirone, "Assistive technologies for the disabled and for the new generation of senior citizens: The e-Tools architecture", *AI-Communications*, 16(3), pp. 193-207, 2003
5. Crowley, J., "Navigation for an intelligent mobile robot", *IEEE Journal on Robotics and Automation*, RA-1(1), pp. 31-41, 1985
6. Fabrizzi, E. and Saffiotti, A. "Extracting topology-based maps from gridmaps", *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*,pp. 2972-2978, 2000
7. Kleeman, L., "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning", *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'92)*, 3, pp. 2582-2587, Nice-Francia, 1992.
8. Koch, E., "Simulation of path planning for a system with vision and map updating", *IEEE Proc. Int. Conf. in Robotics and Automation (ICRA)*, pp. 146-160, 1985
9. Kraetzschmar, G., Sablatnog, S., Enderle, S. and Palm, G., "Application of neurosymbolic integration for environment modelling in mobile robots", in Wermter and Sun (eds) *Hybrid Neural Systems*, Springer, Berlin, 2000.
10. Kuipers, B.J. and Byun, Y., "A robust qualitative method for robot spatial learning", Proceedings of American Association for Artificial Intelligence (AAAI-88), pp. 774-779, Minneapolis, 1988.
11. Kuipers, B.J. and Byun, Y.T., "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representation", *J. Robot. Auton. Syst.*, 8, pp. 47-63, 1991.
12. Moravec, H. P., "Sensor fusion in certainty grids for mobile robots", *AI Magazine*, 9, pp. 61-74, 1988
13. Pearl, J., *Heuristics*, Addison Wesley: New York 1984.
14. Prescott, T.J., "Spatial representation for navigation in animats", *Adaptive Behaviour*, 4, pp. 85-123, 1996.
15. Schiele, B. y Crowley, J., "A comparison of position estimation techniques using occupancy grids", *Robotics and Autonomous Systems*, 12, pp. 163-171, 1994.
16. Sutton, R.S., "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming", *Proceedings of the Seventh International Conference on Machine Learning*, pp. 216-224, June 1990.
17. Thrun, S., Bucken, A., Burgard, W., Fox, D., Frohlinghaus, T., Hennig, D., Hofmann, T., Krell, M., and Schimdt, T., *Map learning and high-speed navigation in RHINO*, MIT/AAAI Press, Cambridge-MA, 1998.
18. Urdiales, C., Bandera, A., Arrebola, F. and Sandoval, F., "Multi-level path planning algorithm for autonomous robots", *Electronics Letters*, 342, pp. 223-224, 1998
19. Warren, C.W. "Global path planning using artificial potential fields", *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 316-321, 1989.
20. Yamauchi, B., Shultz, A. and Adams, W., "Integrating exploration and localization for mobile robots", *Adaptive Behaviour*, 7(2), 2000.
21. Zelinsky, A., "A mobile robot navigation exploration algorithm", *IEEE Trans. on Robotics and Automation*, 8, pp. 707-717, 1992.