HAL
open science

Building topological models for navigation in large scale environments

Dominique van Zwynsvoorde, Thierry Simeon, Rachid Alami

To cite this version:

Dominique van Zwynsvoorde, Thierry Simeon, Rachid Alami. Building topological models for navigation in large scale environments. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, May 2001, Seoul, South Korea. 10.1109/ROBOT.2001.933283 . hal-01979376

HAL Id: hal-01979376

https://laas.hal.science/hal-01979376v1

Submitted on 17 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Building topological models for navigation in large scale environments

D. Van Zwynsvoorde, T. Siméon, R. Alami
LAAS-CNRS
7, Avenue du Colonel Roche, 31077 Toulouse Cedex 4 - France

Abstract

In mobile robotics, pure geometric representations may not be well suited for navigation in large scale environments. New models combine topological and metrical information to give compact and efficient representations.

In this paper, we briefly remind the outlines of the construction of our Voronoï-like graph and give further details about its implementation in a real environment. Several trials made around our lab demonstrate the ability of our modeling scheme to recognize topological features of the environment and use them to detect loops and relocalize the robot position.

1 Introduction

This paper concerns the construction of a model of an *a priori* unknown, large scale environment to be later used in localization and navigation. Both issues become difficult in large environments with pure geometric models [1, 2]. We have chosen to construct incrementally a topological representation from the data obtained by a laser range finder.

Many topological approaches have already been proposed [7, 9, 10, 11, 12]. Graph-like representations are more compact than geometric maps and allow fast route planning. However they have to face the issue of *Place recognition* which is likely to be hard in indoor non-specific large scale environments. Kuipers [9] and Choset [7] designed exploration algorithms that make the robot move in the environment until it can decide it has reached a *Distinctive Place* or a *Meet Point*. In [7] the robot traces the edges of the Voronoï graph and create a new *Meet Point* at each Voronoï vertex. In [10] the system stores a complete metrical representation of each detected *Island of Reliability*. Thrun [11] constructs a topological graph over a metrical map learnt by the robot to plan paths through adjacent regions of the environment.

In our work, we construct the graph directly from sensor data but the model is not strongly linked to the movements of the robot. The aim of the model is to capture the stable topological structures of the environment so that the robot can identify them in navigation tasks to localize itself or even plan motion strategies [3, 4]. The paper first gives an overview of the approach and then focuses on the validation of the algorithms with real data.

2 A Voronoï-like graph model

The main idea of our approach is to construct a topological model of an indoor environment from some range sensor data using an incremental algorithm that first computes a local topological model of the region perceived by the sensor[1] and then updates the global model by merging local and global graphs. This two-step scheme is repeated at each successive position of the robot while it moves around the environment. This approach is summarized in figure 1.

2.1 Local step

The basic structure is a graph obtained from sensor data by computing the Voronoï diagram of a star-shaped visibility generalized polygon formed by the segmented laser data along with artificial segments (escape-lines) modeling visibility constraints. Figure 2-a- shows a typical example of such a graph in a simulated environment. The algorithm presented in [5] traces incrementally the edges of the graph by analyzing the geometry of the polygon in the vicinity of the current position of a virtual tracing point.

Because of the limited angular range of the sensor, the graph may be splitted into several subgraphs as in fig 2-b. From this rough voronoï-based graph we

[1] This is possible because data provided by the LRF are dense, relatively accurate and have a large range (> 10m) compared to ultrasonic sensors.

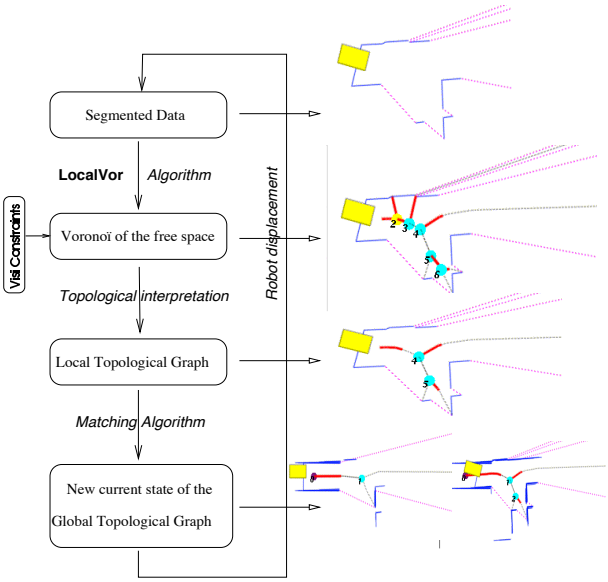
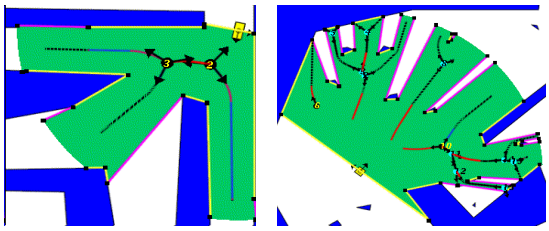


Figure 1: Overview of the Approach

keep useful features that model navigable parts of the environment.

2.2 Incremental construction

The principle of our model learning is an incremental scheme in which we try to connect each local graph to the current state of the global model. The first point is to identify to which part of the global model, each of the subgraphs of the local one will be connected. Then, departing from the selected position, identify what is common, "better" or new in the local model. "Better" means that every node in both local end global graphs may have *temporary* edges constructed based on escape-lines. Thus, as the robot moves, one node is likely to move inside an area according to the evolution of the escape-lines (see [6] for details) and evolves from *temporary* to *definitive* (constructed over real segments only). Figure 3 gives an example of the global step of the approach. From sub-figures -a- to -b all nodes are updated and node 9 turns from *temporary* to *definitive*. In sub-figure -c-, node 11 is inserted between nodes 9 and 10. Finally, node 12 is appended to the graph.



- a - - b -

Figure 2: local graphs: -a- simple, -b- splitted one

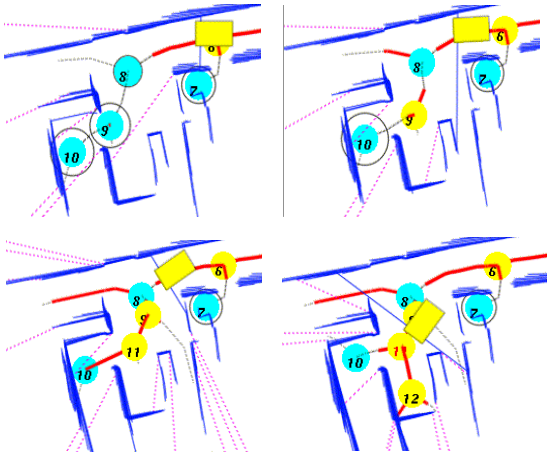


Figure 3: Short scenario of incremental construction

3 Adaptation to real environments

3.1 Connection of the local graph

As seen before, the local graph consists in several subgraphs each of which we try to connect to one edge of the global model. But even in an error-free simulated environment, simply "retracting " the extremity point of the local subgraph L_i onto the global graph may lead to misleading solutions because of visibility limitations, as illustrated by figure 4.

We proposed a score-based connection to verify if the local edge and the candidate of the global model ensure an acceptable continuity. The scores evaluate the collinearity between vectors V_g, V_l, and V_c (which connects local and global extremities) denoted in figure 4-a-. In the part -b- of the figure we see that a simple retraction of extremity D onto the global graph leads to an erroneous connection. Instead, we only process the connection if the total score is above a configurable threshold.

As a result of this procedure, the system could fail to connect any of the following subgraphs. The adopted solution is to initiate a new subgraph of the global model as soon as:

- the robot has covered a limit distance (configurable) without connection, or

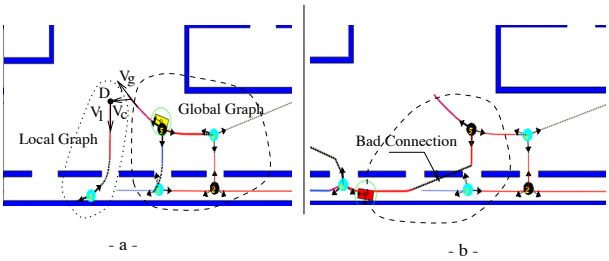


Figure 4: Connection between local and global graphs

- no match has succeeded for the past $MaxFailed$ local graphs (also configurable)

3.2 Connecting global subgraphs

With the goal to use the constructed model to navigate, we expect to have a connected graph. Consequently we will add a *virtual* link connecting edges of global subgraphs which have been sequentially reached by the robot. This *virtual* link means that the system could not model the environment with sufficient certainty but the region between the two so connected subgraphs is navigable by the robot. This may happen when the robot traverses large or poorly structured spaces like in fig 5.

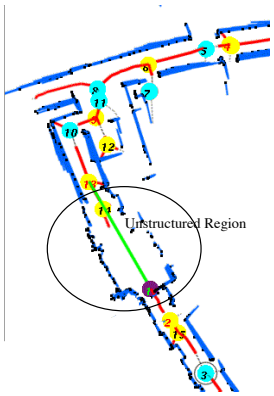


Figure 5: Virtual Link: One virtual link connects two subgraphs separated by a poorly structured region

3.3 Discarding weak subgraphs

To limit the number of global subgraphs we discard local subgraphs that do not seem relevant enough. This concerns local subgraphs formed of one single short edge that cannot be connected to the edge being traced by the robot, but also subgraphs that only contain very "temporary" nodes located almost at the range limit of the sensor.

3.4 Robustness of the model

Figure 7 shows several trials made by manually moving the robot around the corridors of the lab (see a map in figure 6). It shows that well structured parts of the environment can be reliably modeled and produce very similar graphs over the different trials, even when paths are different around the environment, as can be noted comparing subfigure -a- with the others. This means that once one graph has been learnt, it can be used by the robot for localization or navigation, because *strong* topological nodes will be recognized, such as those enclosed in rectangular and elliptic windows in fig 7. In sub-figure 7-b, the loop has not been connected and we can see that the graphs inside the two rectangular windows are very similar. See also figure 13 that situates the graph in the real environment.

One should notice that these trials have been realized without using any "intelligent" exploration algorithm that could increase the quality of the graph construction.

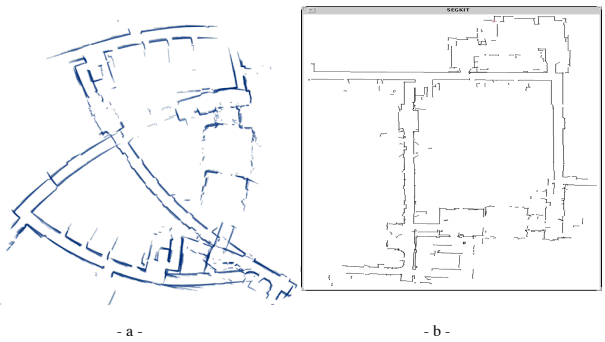


Figure 6: The real environment: -a- as perceived by the robot, -b- the real map. Dimensions are about $50 \times 30m$

4 Loop detection

So far, we have described our approach for incrementally constructing a global model of a real environment. One difficult issue of topological modeling arises when dealing with cyclic environments. Detecting loops in a topological graph requires some *Place Recognition* capability to allow the system to decide whenever the robot is re-traversing an area that has already been modeled.

Our loop detection scheme manages hypotheses formulated from candidate nodes isolated from the whole set of already created nodes. Although it is not a

novel technique, its use in the context of local topological graphs allows making hypotheses without the need for the robot to be located on the node, as for the *rehearsal procedure* of Kuipers's work [9], also found in Choset's approach [7].

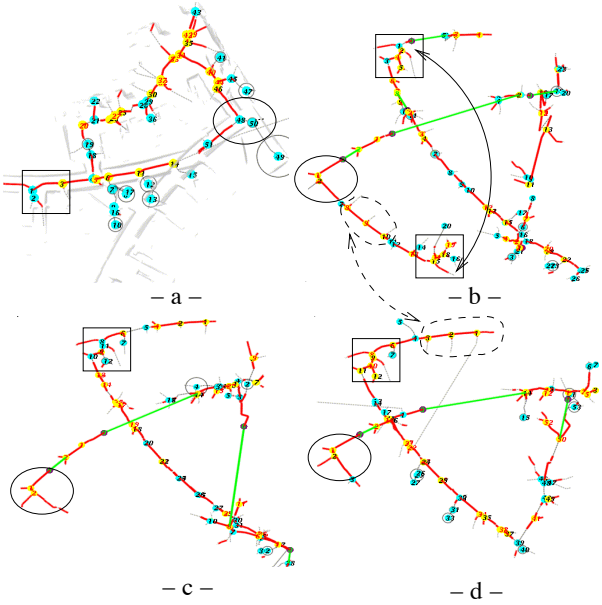


Figure 7: Graphs of different trials

4.1 Formalization of a hypothesis

In the following, newly discovered nodes will be called *local*, whereas nodes previously modeled are referred to as *global*.

For each newly discovered node Nl during the incremental step of the construction, we generate a list of candidate nodes $\mathcal{C}and(Nl) = \{Ng_i\}$ selected from the nodes of the global graph such that each Ng_i topologically "looks like" Nl. The description in terms of qualitative and quantitative features of the nodes (see [5]), along with neighboring topology comparison is used to generate such a list.

For each element of $\mathcal{C}and(Nl_i)$ we search all the $\mathcal{C}and(Nl_j)$ where $i \neq i$, for a possible sequence connecting both candidates by a path similar to the one connecting Nl_i to Nl_j.

4.2 Insertion of a new node

A hypothesis thus results in a "local" part and a "global" one. The local part connects at least two recently discovered nodes whereas the global one connects the corresponding candidates in the global graph. When considering the candidates of a new local node Nl, we examine each of the existing hypotheses to detect if the candidate node Ng_i corresponds to one node in the path of the hypothesis that was not yet participating to it (for which no candidate had been found so far). If the considered candidate does not belong to the path of the hypothesis, it may connect to it, increasing the length of the hypothesis. In each of these situations, the hypothesis is further confirmed and the couple (Nl, Ng_i) is set to form part of the hypothesis as a *detected* node (this is denoted by an arrow connecting the candidate nodes in figure 8).

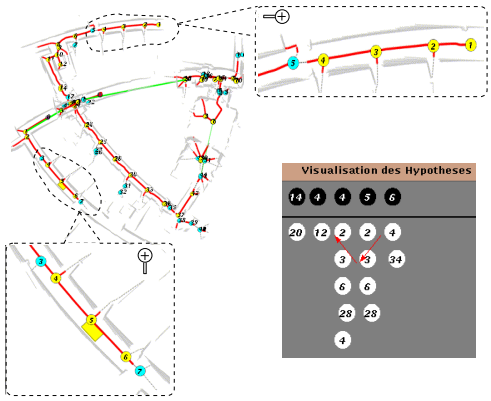


Figure 8: Valid hypothesis: A three-node length hypothesis could be generated: Nodes 4-5-6 may correspond to nodes 2-3-4

4.3 Invalidating a hypothesis

On the contrary, further exploration may allow invalidating a hypothesis. Comparing the local and the global paths we can detect wrong hypotheses in which the last *detected* node of the hypothesis cannot be connected to a node such as the newly modeled node Nl.

4.4 Closing the loop

As identical (or very similar) topological structures may be present in an indoor environment, we must capture a significant part of them to be able to decide whether one loop has been detected or not. However, it does not exist any terminating condition that allows detecting loop in any indoor environment, unless we incorporate more information into the graph description, such as the presence of posters or plants in the vicinity of a node,or elementary rules of building constriction for example.

If more than a limited number of nodes are *detected* in an hypothesis, we decide to close the loop. To do so,

the algorithm (but not the robot) traces back the local and global parts of the hypothesis simultaneously until the end of one of the paths or any topological difference is found. There, the local part of the hypothesis is erased and the pending edge is connected to the correspondent global edge. Compare figures 8, 9. In figure 8 nodes numbered 4, 5, 6 are recognized to be nodes 2, 3, 4. After closing the loop we obtain the graph of figure 9.

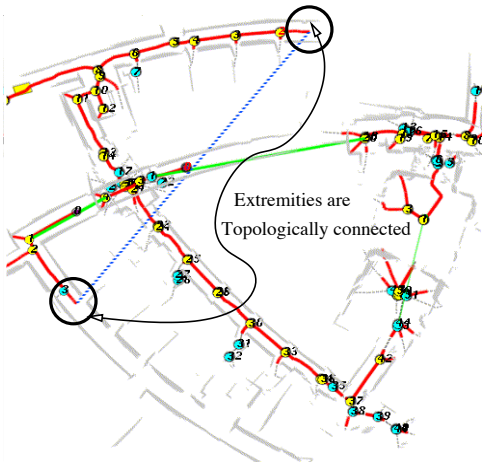


Figure 9: Loop example

5 Using the graph to limit pose estimation errors

We have shown that our learning scheme allows modeling cyclic environments. However, the displays illustrate well that the graph appears quite distorted because of odometric errors. Although it is not dramatic for forward movements, a problem of topological consistency arises when the robot intentionally retraces a previously modeled region. Figure 10 shows an example where the robot turns back to its initial point. The strong odometric error accumulated during the manoeuver causes a loss of connection and generates a new subgraph modeling the back-way part of the exploration.

To avoid this phenomenon, we use the graph to limit pose estimation errors. Two situations can be outlined: i) in forward motions, ii) re-discovering a node:

"Continuous" relocalization

We can use the geometric description of edges to detect an important orientation discrepancy between the local edge and the corresponding one in the global model. Comparing their respective averaged direction, we get a value $\hat{e} = \theta_l - \theta_g$ that we can use to relocalize the robot and correct the local graph, before using it for the global update.

If the local graph contains one *definitive* node that is already present in the global graph, then we can estimate a value for $\hat{e}$ by comparing the departing angles of the edges emanating from each of the two nodes. Again, we must consider values for $\hat{e}$ over a lower limit conditioned by segmentation errors. Indeed, *definitive* nodes are not present in many successive local models because of visibility constraints. Hence, as odometry does not drift so much along a short trajectory, values of $\hat{e}$ could be rather small and not due to odometric error.

Figure 10: Effect of Relocalization

Re-discovering a node

The second situation is when closing a loop or intentionally turning back-way. Once we have estimated the orientation error $\hat{e}$, we must compute position errors (D_x, D_y), related to a local node Nl and the global one Ng: to the model:

$$(Real)\begin{cases} X_{Nl} &= X_r + D_x + l * cos(\theta_l + \theta_N) \\ Y_{Nl} &= Y_r + D_y + l * sin(\theta_l + \theta_N) \end{cases}$$

$$(Theorical)\begin{cases} X_{Ng} &= X_{th} + l * cos(\theta_l - \hat{e} + \theta_N) \\ Y_{Ng} &= Y_{th} + l * sin(\theta_l - \hat{e} + \theta_N) \end{cases}$$

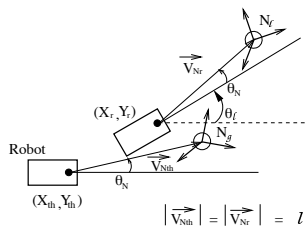


Figure 11: Pose estimation error

Which leads, given $\delta = \theta_l + \theta_N$, to:

$$\begin{cases} D_x &= X_{Nl} - X_{Ng} + l * (cos(\delta) - cos(\delta - \hat{e})) \\ D_y &= Y_{Nl} - Y_{Ng} + l * (sin(\delta) - sin(\delta - \hat{e})) \end{cases}$$

Figure 12 shows a graph from the same environment obtained with continuous relocalization. One can ob-

serve that the graph is less distorted than the rough sensor data but still is not completely corrected. This is not the principal aim of our work, but it can help reducing the number of candidate nodes.

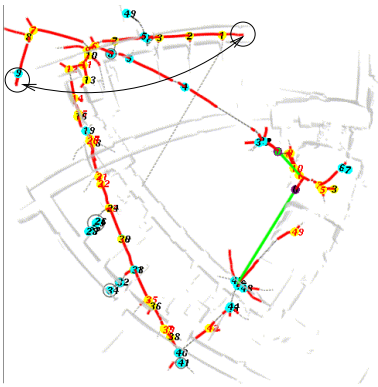


Figure 12: Continuous relocalization

6 Future work

Obviously, this model does not provide any exact geometric representation of the environment. However it can certainly be used to give a qualitative information about where the robot is. In particular it could give a way of answering the question of initial localization of the robot. Letting the robot somewhere in the environment, the same hypotheses generation procedure as used for loop detection could be deployed to recognize part of the previously learnt graph and so give one or more possible location.

One other promising use of the graph concerns the generation of sensor-based motion plan. Given two specified points on the graph (typically hand-supplied by the user), it is straight-forward to find a topological path that link the two points. One step toward sensor-based motions is to analyze the information contained in the model along the topological found path to produce a sequence of semantic portions such as CORRIDOR, HALL, CORNER OR BLIND-MOTION. The next step would be to assign to each type of portion one or several possible sensor-based motion strategy like WALL-FOLLOWING, TURN-CORNER, GO-TO-POINT, OR GO-UNTIL-SEE-NODE,... or even use the Voronoï itself to navigate [13], then find terminating conditions that ensure the feasibility of the plan.

References

[1] J.L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *IEEE International Conference on Robotics and Automation, Scottsdale (USA)*, 1989.

[2] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. In *IEEE Trans. on Robotics and Automation*, 7(3):376-382, 1991.

[3] M.Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Intelligent Autonomous System*, Karlruhe (Germany), 1995.

[4] B. Bouilly, T.Simeon , R. Alami. A numerical technique for planning motion strategies for a mobile robot in presence of uncertainties. In *IEEE International Conference on Robotics and Automation*, Nagoya (Japon), 1995.

[5] D.Van Zwynsvoorde, T.Simeon, R.Alami. An algorithm for incrementally constructing a topological graph for long range navigation, (in french), in *RFIA 2000*, Paris, (France), 2000.

[6] D.Van Zwynsvoorde, T. Simeon, R. Alami. Incremental Topological Modeling using Local Voronoï-like Graphs. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.*

[7] K. Nagatani, H. Choset, S.Thrun. Towards Exact Localization without Explicit Localization with the Generalized Voronoï Graph.*In IEEE ICRA*, Leuven (Belgium), 1998.

[8] K. Nagatani, H. Choset. Toward Robust Sensor Based Exploration by Constructing Reduced Generalized Voronoï Graph. In *IROS*, Kyongju (Korea), 1999.

[9] B. Kuipers, Y.T. Byan. A robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal of Robotics and Autonomous Systems*, 8:47-63, 1991.

[10] S. Simhon, G. Dudek. A global topological Map formed by Local Metric Maps. In *IROS*, Victoria (Canada), 1998.

[11] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *In AI Journal 99(1)*, 1998, pp. 21-71.

[12] R. Simmons, S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In *IJCAI*, 1995.

[13] A. Victorino, P. Rives, J.J Borrelly. Localization and Map Building using a Sensor-Based Control Strategy. *IROS 2000.*

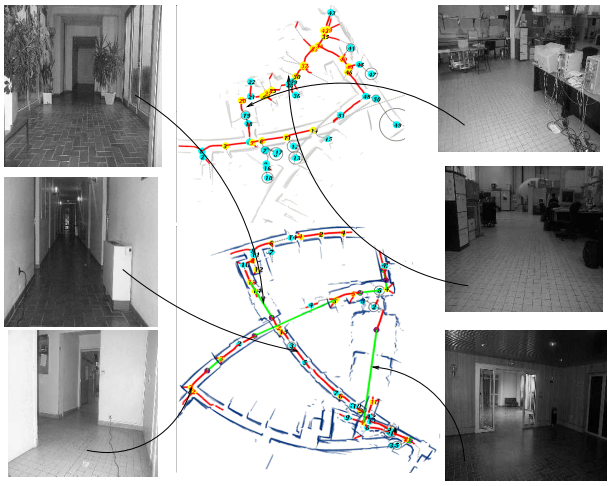


Figure 13: Graphs of real environments