

This is the published version of a paper presented at 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012; Vilamoura, Algarve; 7 October 2012 through 12 October 2012.

Citation for the original published paper:

Aydemir, A., Jensfelt, P., Folkesson, J. (2012)

What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments.

In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 4675-4682). IEEE

IEEE International Conference on Intelligent Robots and Systems

http://dx.doi.org/10.1109/IROS.2012.6386110

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-111534

# What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments

Alper Aydemir

Patric Jensfelt

John Folkesson

Abstract-Many robotics tasks require the robot to predict what lies in the unexplored part of the environment. Although much work focuses on building autonomous robots that operate indoors, indoor environments are neither well understood nor analyzed enough in the literature. In this paper, we propose and compare two methods for predicting both the topology and the categories of rooms given a partial map. The methods are motivated by the analysis of two large annotated floor plan data sets corresponding to the buildings of the MIT and KTH campuses. In particular, utilizing graph theory, we discover that local complexity remains unchanged for growing global complexity in real-world indoor environments, a property which we exploit. In total, we analyze 197 buildings, 940 floors and over 38,000 real-world rooms. Such a large set of indoor places has not been investigated before in the previous work. We provide extensive experimental results and show the degree of transferability of spatial knowledge between two geographically distinct locations. We also contribute the KTH data set and the software tools to with it.

#### I. INTRODUCTION

Predicting what lies in the unexplored part of the environment constitutes a crucial part of a robotic system. Various robotics tasks such as rescue missions in unknown buildings or task driven exploration requires the robot to reason about unknown space. The quality of such reasoning depends on the accuracy of the robot's predictions about the unexplored part of the environment.

To illustrate this, imagine a mobile office helper robot, on the first day of its operation in an unknown office building and tasked with cleaning the kitchen. The robot needs to plan the set of actions that lead to task completion. In this case, it first needs to find a kitchen by means of exploration. One sensible plan would make use of corridors as connector rooms in most indoor environments. The robot's expectation of finding a kitchen by exploring more of a corridor should be higher than finding a kitchen connected to one of the many other room types it detects during exploration.

Another example is a search-and-rescue robot, entering an unknown office building in the search of survivors. In this case, the robot's goal is to find the maximum amount of survivors in the minimum possible time [1]. The robot, without a complete map of the environment, would need to employ spatial reasoning to estimate what lies in the unknown part of space, prioritizing search actions, which regions to look for survivors in and how best to reach them.

Such highly needed autonomy requires a general understanding of how indoor environments are typically config-

The author is with the Centre for Autonomous Systems at the Royal Institute of Technology (KTH), Stockholm, SE-100 44, Sweden. [aydemir,patric, johnf]@csc.kth.se

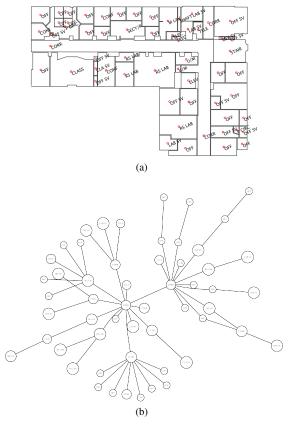


Fig. 1: Two representations of a building floor. (a) shows the bird-eye view of the geometric layout and (b) the corresponding graph where each room corresponds to a node in the graph and edges between nodes refer to a direct traversable path. Nodes in the graph have labels indicating the category of that room such as kitchen, office, corridor.

ured, currently lacking in state-of-the-art robot systems. Even though much work focuses on building autonomous robots that operate indoors, indoor environments are neither well understood nor analyzed enough. In most systems where structural and semantic spatial information is useful, the models of large scale indoor environments are hard-coded and therefore limited [2], [3], [4].

Indoor environments are organized in inter-connected rooms<sup>1</sup>. A widely used representation for this is an undirected graph, where each room corresponds to a node

<sup>1</sup>Rooms may or may not be separated by physical boundaries, e.g. a kitchen consisting of a sink area and dining area, as recently shown by Sjöö [5]. We make no distinction between the two and refer both as rooms.

in the graph and edges between nodes refers to a direct path between rooms. Figure 1 shows a bird-eye view of an building floor plan represented as a undirected graph. Furthermore, each room can have a label indicating the category of that room, e.g. kitchen, office, corridor, restroom. This type of representation is often called a topological map of the environment.

In this paper we adopt a data-driven approach for building models of indoor environments and predicting what lies ahead in the topology of the environment. We make use of two large floor plan data sets from the MIT and KTH campuses, with each floor represented as a graph as introduced above. The whole corpus of real world indoor spaces consists of 197 buildings, 940 floors and over 38,000 rooms. To the best of our knowledge no such work exists on datadriven models of indoor maps on this scale. An important property we discover is that, in floor plan graphs local complexity remains nearly constant for increasing global complexity, as we will show, a property observed in other natural graph networks as well. Using this insight, we present two methods for predicting extensions to partial topological maps including room category labels. Finally, we present extensive experimental results on the performance of both methods. In particular, we show how well the models of floor plan graphs learned from the MIT data set transfer to the KTH data set.

## A. Contributions

Our contributions in this paper are three fold:

- We report statistical properties of indoor environments by investigating the two floor plan data sets from different parts of the world. This data-driven approach to understanding man-made environments at the topological map level is new and not found in previous work.
   We present various properties of indoor environments with the aim that such analysis will improve future designs of robots that operate indoors.
- Inspired by this analysis, we present two methods for predicting indoor topologies given a partial map of the environment. We evaluate both algorithms' performance with extensive experiments.
- We make the KTH campus data set, our annotation tool and software library developed during this work publicly available at http://www.cas.kth.se/floorplans.

# B. Outline

In section II, we begin with the problem formulation and notation used throughout this paper. We then present a straight forward algorithm in section III and discuss its shortcomings and how can we address them. This requires an in depth analysis of the data set, as we present in Sections IV and V. Inspired by this analysis, in section VI we present our main method. In section VII we provide a extensive experimental results on both methods and in particular how the models learned from one data set transfer to the other.

#### II. PRELIMINARIES AND PROBLEM FORMULATION

We represent each floor plan as an undirected graph. Each node in a graph is assigned a label from an ordered, finite alphabet. The nodes correspond to rooms, the labels are room categories and an edge between two nodes means that there exists a traversable direct path connecting them. An example floor plan graph is shown in Figure 1.

A graph is then a three-tuple  $G=(V,E,\alpha)$  where V is a finite node set,  $E\subseteq V\times V$  is a finite edge set and  $\alpha:V\to\mathcal{L}$  is a node label mapping. Let  $\mathcal{G}$  be the set of all graphs that can be formed using the label alphabet  $\mathcal{L}$ .

A graph database  $\mathcal{D} = \{G_1, ..., G_n\}$  is a set of graphs. Given a graph  $G \in \mathcal{G}$  and a graph database  $\mathcal{D}$ , we define the projected database  $\mathcal{D}_G$ , as the subset of  $\mathcal{D}$ , for which G is a sub-graph of each element, i.e.  $\mathcal{D}_G = \{\forall G' \in \mathcal{D} : G \subseteq G'\}$ . The number of elements of the projected database is called the frequency of the graph G in the graph database  $\mathcal{D}$  and is denoted by  $freq(G) = |\mathcal{D}_G|$ .

We may now define the *support* of the graph G as:

$$supp(G) = \frac{freq(G)}{|\mathcal{D}|} \tag{1}$$

A graph G will be called a *frequent subgraph* in  $\mathcal{D}$  if  $supp(G) \geq \tau$  where  $\tau$  is some minimum support threshold,  $0 < \tau < 1$ .

Finally, we define an *edit operation* to a graph as a node or an edge addition <sup>2</sup>

## A. Problem formulation

Given an incomplete floor plan graph, predict the most likely next room category together with where it is connected to in the current graph or the most likely new path between the two rooms.

In formal terms, let  $G_p$  be a partial floor plan graph for which the full graph is G. We want to find a certain discrete probability distribution which determines the probability of an edit operation (a node or an edge addition) to  $G_p$ . Once this discrete probability distribution is acquired, it is possible to attain the most probable next floor plan graph  $G_p'$ . This graph is the result of applying the most probable edit operation upon  $G_p$ .

## III. METHOD I - COUNT BASED PREDICTION

Following the problem formulation, we present the following algorithm. Given an initial input graph  $G_p$ , we first compute its projected database  $\mathcal{D}_{G_p}$ . This means that for each element  $G' \in \mathcal{D}_{G_p}$  it holds that  $G_p \subseteq G'$ . For each edit operation e applicable to  $G_p$  with the resulting graph  $G'_p$ , we calculate  $supp(G'_p)$  as described in Section II. Finally, the edit operation whose resulting graph has the highest support is taken as the most likely node or edge addition to  $G_p$ . This method is akin to a hidden Markov Model (HMM) formulation where the state of the model is the graph itself and actions are edit operations.

<sup>&</sup>lt;sup>2</sup>An edge addition should connect two existing nodes in the graph. Likewise, a new node is added with an edge to an existing node.

As we will show in the evaluation section this algorithm performs well for small graph sizes. However, it is limited in the sense that it considers whole graphs at once. Therefore, subtle changes to the input graph can result in drastic changes in the predictions. As an example, if a given input partial graph contains a node with an uncommon or non-existing label in  $\mathcal{D}$ , the algorithm will only consider those graphs which include this rare node label and disregarding others for building  $\mathcal{D}_{G_p}$ . Worse yet, in a robotics scenario, it's not unreasonable that certain rooms remain uncategorized or that the system may be uncertain about the category of a room. In the latter case, method I would report zero probabilities for any possible edit operation. This leads to poor generalization and has the undesired effect of discarding good candidates for prediction.

Therefore we expect that an algorithm that instead exploits the *local structure* in floor plan graphs rather than the graph itself would fare better. To develop a better method, we first analyze the properties of floor plan graphs with the goal of finding exploitable structure in the data.

#### IV. DATA SET

Each floor plan in the data sets is stored as an XML file, consisting of a set of rooms. Each room has the following:

- The 2D layout represented as line segments and the room's centroid coordinates.
- A set of doorways<sup>3</sup> that indicates a direct traversable path between this room and others it is connected to, if any.
- A category label, e.g. office, corridor, lavatory.

The MIT campus floor plan data set is originally created by [6] to assist campus travelers by building a machine readable representation of the campus buildings. We propose to use the data set in a mobile robot context as described in this paper. This data set covers over 160 buildings, 775 floors and nearly 32,000 rooms with 91 categories.

We've created the KTH campus data set by annotating the architectural floor plan drawings of the KTH campus in order to evaluate our algorithms on a wider set of indoors data and make it available to the community. We have adopted the MIT data set XML format with the addition that in the KTH data set, a line segment of a room's layout can be of three types: a wall, a window or a doorway to another room. In the KTH data set, we have annotated 37 buildings, 165 floors and 6248 rooms. This results in a very rich representation of large scale indoor environments that has not been used in the literature.

We have made the KTH data set, the annotation tool and an easy to use C++ library for loading and manipulating floor plan databases available at http://www.cas.kth.se/floorplans. Next, we will study various properties of both data sets.

## V. ANALYSIS OF THE DATA SET

As discussed earlier, each floor plan can be represented naturally as a graph. The mathematical properties of graphs

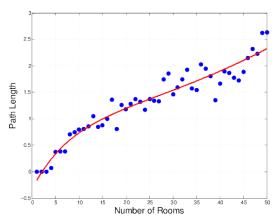


Fig. 2: Characteristic path length of indoor topological maps for increasing graph sizes.

are very well researched. We first start by looking at the global properties of floor plans and then examine the local structure.

# A. Global properties

[7] introduced the property *small-world* to describe the type of graphs in which most nodes can be reached from every other in a small number of steps. Naturally occurring graphs such as social networks between humans and the Internet are shown to be small-world. It is yet to be shown if indoor environments fall into this category. The significance of this in a mobile robot context is that, if floor plans possess small-world characteristics, methods that reason about space can be improved by exploiting its various implications.

The clustering coefficient together with the characteristic path length of a graph are used to determine the small-worldness. Graphs that have a high clustering coefficient and low characteristic path compared to that of a random graph<sup>4</sup> can be called small-world graphs. Next we explain both concepts in more detail and present our results.

1) Characteristic Path Length: We look at the average number of nodes (i.e. rooms) a robot needs to visit in order to travel from one room to another, computed over all pairs of rooms. This is called the characteristic path length of a graph. This value gives insights on how connected indoor environments are, it also provides the expected amount of rooms to traverse in order to move from one room to another.

The average characteristic path length for increasing graph sizes (i.e. number of rooms in the environment) is shown in Figure 2. To elaborate on this, note that for graph sizes between 10 and 25 this value is around 1. What his tells is that, one has to go through on average one another room to be able to reach anywhere on this floor from anywhere else. Our insights are well in line with this result, this intermediate room usually corresponds to a corridor. However the characteristic path length increases roughly linearly with the graph size.

<sup>&</sup>lt;sup>3</sup>A doorway can be either a physical door or a conceptual boundary between two spaces, i.e. printer area in a corridor.

<sup>&</sup>lt;sup>4</sup>In a random graph, the number of edges that a node has is determined randomly [8].

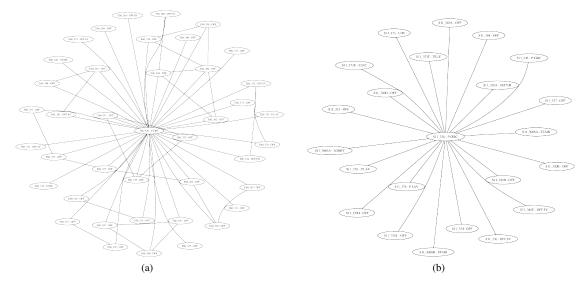


Fig. 3: Two example floor plan graphs showing different clustering coefficients of 0.56 for (a) and 0 for (b). Since certain nodes in (a) have neighbors that are connected to each other, the value for (a) is non-zero, however no such node exists in (b), as a result of this, if the central node is removed, the average path length for the graph in (b) would be infinite.

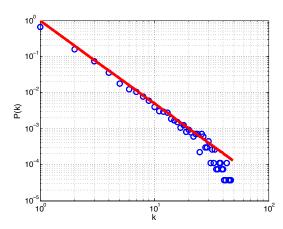


Fig. 4: Degree distribution of the data set in logarithmic scale. The slope of the line gives the  $\gamma$  value discussed in Section V-B.1

2) Clustering Coefficient: Widely used in graph analysis, the clustering coefficient, C, indicates whether nodes in a graph are dispersed from each other or form densely connected subgroups. [9] gives a comprehensive overview of the results in previous work where the clustering coefficient for several man-made graph networks is computed. Figure 3 shows two graphs from the data set which have different clustering coefficients.

In [7], the authors calculate the clustering coefficient of the United States power grid and the neural network of the worm C. elegans among others and found that  $C_{powergrid} = 0.08$  and  $C_{c.elegans} = 0.28$  respectively, compared to  $C_{random} = 0.00027$  and  $C_{random} = 0.005$  for random graphs of the same sizes. We have computed the average clustering coefficient of our database and found it to be  $C_{floorplan} = 0.08$ , i.e. same as the power grid network. Since our floor plan graphs are in similar size and sparsity compared to a power

grid network and much less dense than a biological neural network graph, it is natural to obtain a value closer to that of the power grid data set. Another possible conclusion is that similar efficiency concerns are valid for efficient travel of people in the indoors and electricity across the grid.

#### B. Local properties

1) Scale-free graphs: An important local property that is shown to remain invariant in many natural graphs despite the high complexity is the scale-free property. In scale-free graphs, the probability that a node has k edges, is defined as:

$$P(k) \propto k^{-\gamma} \tag{2}$$

It has been shown that this seemingly simple property holds for very complex graphs such as the Internet, scientific citations, power grids and social network edges between humans [10], [9]. In effect, this result indicates that across different natural graph data sets, the local connectivity is free of scale and remains unchanged. Figure 4 shows the edge distribution of our data sets on logarithmic scale. We have computed  $\gamma$  as 2.073 for graphs of indoor environments compared to 2.1 for the World Wide Web and 2.4 for the United States power grid which are known to possess scale-free characteristics.

2) Single node statistics: Next, we have looked into single node characteristics of floor plan graphs. Figure 5 shows the average number of edges per room category, for the 15 most frequent categories in both data sets. Here, we can see that corridors mainly act as connector nodes while the rest of the room categories have single edges mostly.

Figure 6 shows most common 30 room categories for the data sets. Here we can see that the general distribution for both data sets are similar with some regional differences. As an example, toilets are much more numerous in the KTH

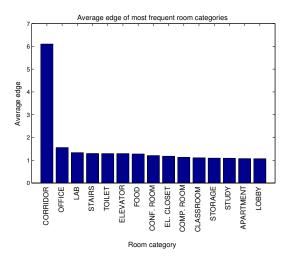


Fig. 5: Average edges for most frequent room categories in both data sets.

data set, the reason being is that at KTH, each toilet stall has its own walls and door opening to the floor.

#### C. Discussion

From this analysis we can see that the topologies of these indoor environments are strongly structured. In particular, a relatively high clustering coefficient indicates that nodes in floor plan graphs tend to form tightly connected subgraphs. In addition, low characteristic path length despite increasing graph sizes together with the edge distribution shown in Figure 4 point out that certain rooms have a very high number of direct pathways to other rooms, acting as connector rooms.

With the goal of exploiting this structure in the data, we can argue that a floor plan consists of several functional parts, each of which amounts to a small collection of rooms. As an example, the subgraph  $\{office-corridor-stair\}$  is useful for navigation between floors in a building and whereas  $\{office-corridor-lavatory\}$  fulfills another necessity in indoor environments. The idea of combining functional parts is also supported by the scale-free property of floor plan graphs as shown previously which displays invariance of local structure for increasing graph sizes. Therefore, we have developed our methods making use of such frequently occurring subgraphs, introduced formally in Section II.

## VI. METHOD II - EXPLOITING FUNCTIONAL SUBGRAPHS

Following the lessons learned from looking at the structure of floor plans, we extract frequently occurring subgraphs from the data set. They constitute building blocks of the topology of indoor environments. Extracting frequent subgraphs of any size from a graph data set is known to be an NP-Complete problem. We use the gSpan algorithm [11], which is a fast algorithm for extracting frequently occurring subgraph patterns of any size in a graph data set. This provides us with the set of frequent subgraphs extracted from our database,  $\mathcal{S}$ , used in the first step of the method.

The main steps of this method is given in the following:

- 1) Split the input graph into smaller, overlapping subgraphs, forming a set C.
- 2) For each subgraph in C, determine the probability of every possible edit operation.
- Combine the results of the estimates of the edit operations for each subgraph into a final solution for the whole input graph.

Step 1: The goal of this step is to divide the input graph  $G_p$  into a set of overlapping connected subgraphs  $\mathcal{C}=\{C_1,\ldots,C_m\}$  such that  $\forall i\exists j,C_i\cap C_j\neq\emptyset$ . The selection of subgraphs  $C_i$  plays an important role in the prediction quality. Our hypothesis is that indoor topologies consists of multiple functional smaller parts and we should try to exploit these. We do this by, when possible, picking subgraphs from  $\mathcal S$  and expand them as viable predictions. Therefore, in the first step of the method, we determine which of the frequent subgraphs from  $\mathcal S$  that are present in the input graph, and extract the largest possible such frequent subgraphs set. The procedure for computing  $\mathcal C$  is given in Algorithm 1.

Step 2: In this step, we aim to calculate the probability of all possible edit operations for each element of  $\mathcal{C}$ . Let  $\mathcal{D}_{C_i}$  be the projected database of  $C_i \in \mathcal{C}$ . Let  $\mathcal{X}_{\mathcal{C}_i}$  be the set of graphs which are one edit operation away from  $C_i$ . We then define, for  $x \in \mathcal{X}_{C_i}$ ,  $\phi(x,C_i) = |\{x \subseteq G' \in \mathcal{D}_{C_i}\}|$ . This is the number of times we have observed a specific edit operation upon  $C_i$  among all the graphs. The most likely edit operation to perform on  $C_i$  is then given from  $\mathop{\mathrm{arg}} \max \phi(x,C_i)$ . This  $\underset{x \in \mathcal{X}_{C_i}}{\operatorname{constant}}$ 

procedure is given in detail in algorithm 3.

Step 3: The most likely edit operation for each of the subgraphs  $C_1, ..., C_m$ , lead to new graphs  $C'_1, ..., C'_m$  respectively. We must select one of these edit operations to get the final prediction outcome. For any selection  $C'_j$  made, the resulting prediction will be  $G'_p = \bigcup_{i \in [1,m] \setminus \{j\}} C_i \cup C'_j$ . We select the edit operation which has the highest support from the graph database. That is,  $\underset{C_i, i \in [1,m]}{\arg\max} \phi(C_i, C'_i)$ .

Given the function  $\phi: \mathcal{G} \times \mathcal{G} \to \mathbb{N}$ , it is possible to arrive at an estimate of the discrete probability distribution of the different edit operations upon  $G_p$ :

$$P(G_p' = x) = \frac{\phi(x, C_j)}{\sum_{y \in \mathcal{X}_{C_i}} \phi(y, C_j)}$$
(3)

 $C_j$  here refers to the selected subgraph and is chosen as detailed above.

# VII. EXPERIMENTS

As stated in the problem definition, given a partial graph we want to estimate the probability distribution over possible edit operations, which correspond to predictions about what lies ahead in the unexplored part of the environment. We have tested the accuracy of predictions for various graph sizes of the input partial graph. The test procedure is as follows. First we pick a random graph G from the data set and given m, the desired size of the input graph to be included in the test set, we select m connected nodes from G randomly. The m

## Algorithm 1 Graph splitting

Input:

•  $G_p$ , the current partial graph

## Output:

```
partial graph

1: P \leftarrow \emptyset

2: for s \in \mathcal{S} do

3: if s \subseteq G_p \land (\neg \exists s' \in \mathcal{S}, s \subseteq s', s' \subseteq G_p) then

4: P \leftarrow P \cup \{s\}

5: end if

6: end for
```

•  $\mathcal{C} = \{C_1, ..., C_m\}$ , the overlapping subgraphs of the

 $\{P \text{ now contains those frequent subgraphs which are contained in the partial graph } G_p$ . They are also the largest possible frequent subgraphs.  $\}$ 

```
7: sort(P) by graph size, descending.
 8: \mathcal{C} \leftarrow \{\text{FindCommonFreqSubgraph}(P, G_p, \emptyset)\}
    while |G_p| \neq |\bigcup_{i=1}^n C_i| do
         Found \leftarrow 0
10:
        for all c \in \mathcal{C} \wedge Found = 0 do
11:
            c' \leftarrow \text{FindCommonFreqSubgraph}(P, c, \mathcal{C})
12:
            if c' \neq \emptyset then
13:
               \mathcal{C} \leftarrow \mathcal{C} \cup c'
14:
               Found \leftarrow 1
15:
16:
               break
           end if
17:
        end for
18:
19:
        if Found = 0 then
            D_g \leftarrow G_p \setminus \bigcup_{i=1}^n C_i
20:
            Add the following
                                                 vertex
21:
            \bigcup_{v \in V(D_q)} N(v, G_p) \setminus D_q
            Add the edges (from the edge set of G_p) which
22:
            correspond to the vertex additions above.
            \mathcal{C} \leftarrow \mathcal{C} \cup \operatorname{GetComponents}(D_a)
23:
           return C
24:
         end if
25:
26: end while
27: return \mathcal{C}
```

connected nodes and their edges result in subgraph  $G_p \subset G$ .  $G_p$  is added to the test set and this procedure is repeated until the desired amount of test graphs is acquired. Then, for each graph in the test set, we compute the edit operation with the highest probability using both methods. Applying this to  $G_p$  results in a new graph  $G_p'$ . In order to evaluate the performance, we count every time the new resulting graph  $G_p'$  is included in the actual true graph  $G_p'$  in same place as  $G_p$ <sup>5</sup>. Note that we train and test on different subsets of the data, so  $G_p$  will never be part of the training data.

We have evaluated both methods on three experimental setups. In the first two experiments, we have trained and tested our algorithm on single data sets. We have used approximately 80% of the data sets for training and the

# Algorithm 2 FindCommonFreqSubgraph

This function will attempt to find another frequent subgraph from the set P that has some vertex in common with some graph  $C_i$  (the already established subgraphs of  $G_p$ ). Input:

- P, the sorted sequence of frequent subgraphs that are present in the partial graph
- G, a graph which the result should have some vertex in common with, this is always some  $C_i$  except for the initial execution.
- $C = \{C_1, ..., C_n\}$ , the thus far added overlapping subgraphs of the partial graph

## Output:

p, the largest frequent subgraph present in the partial graph that has at least one vertex in common with G (if found). p is also removed from the set P. If no such p could be found, it returns the empty graph Ø.

```
1: for all p \in P do

2: if \operatorname{HasVertexInCommon}(G, p) \land p \not\subseteq \bigcup_{i=1}^n C_i then

3: P \leftarrow P \setminus \{p\}

4: return p

5: end if

6: end for

7: return \emptyset
```

# Algorithm 3 Find most likely graph edit operation

Input

- G, a "small" graph, one subgraph from the output of the graph splitting.
- $\mathcal{D}$ , the graph database

#### Output:

 $\bullet$  G', the graph which is the result of performing the optimal edit operation upon G

```
\begin{array}{l} \mbox{for } x \in \mathcal{D} \mbox{ do} \\ \mbox{if } G \subseteq x \mbox{ then} \\ \mbox{for } G' \in B(G,1) \land G' \subseteq x \mbox{ do} \\ \mbox{ \{Every $G'$ corresponds to some valid edit operation upon $G$ (that is, both $G$ and $G'$ are contained in this specific graph $x$).} \\ \mbox{ } \phi(x,G) \leftarrow \phi(x,G) + 1 \\ \mbox{ end for } \\ \mbox{end if } \\ \mbox{end for } \\ \mbox{return } \mbox{ } \underset{x \in B(G,1)}{\operatorname{max}} \phi(x,G) \\ \mbox{ } \\ \mbox{ } x \in B(G,1) \\ \end{array}
```

 $<sup>^5{\</sup>rm This}$  is to avoid counting the cases where  $G'_p$  appears somewhere  ${\it else}$  in G as true positives.

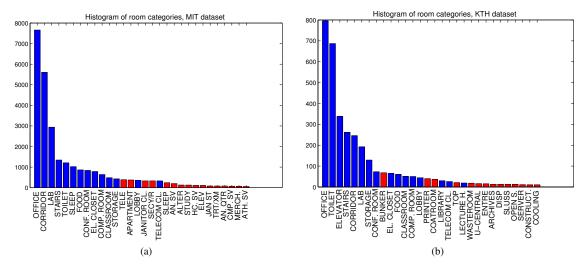


Fig. 6: Histograms of 30 most frequent categories of the KTH and MIT data sets. The blue color indicates categories that exists on both data sets, red is unmatched categories. Rooms with matched categories constitute 74% and 81% of all rooms in the MIT and KTH data sets respectively.

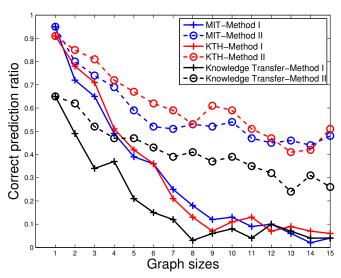


Fig. 7: The results of the three experimental setups. The solid lines correspond to method I and dashed lines correspond to method II. MIT, KTH and knowledge transfer experiments are color coded blue, red and black respectively.

remaining 20% for testing. In the third experiment, we have used the MIT data set for training and KTH data set for testing in order to investigate how transferable is the spatial indoor knowledge between two data sets from distinct places in the world. For all experiments, we have sampled the data set for 30.000 partial graphs in the test set with no duplicates.

Figure 7 shows the results where solid lines denote method I's results and dashed lines correspond to method II's outcome. Furthermore, blue and red lines correspond to the first two experiments where the algorithms are trained and tested on MIT and KTH data sets separately and black lines correspond to the third experiment where we investigate how well the spatial models trained on the MIT data set transfer

to the KTH data set, we call this the knowledge transfer experiment.

In all cases, we can see that method II performs better compared to method I. Looking more closely, for small graph sizes the difference in performance is minimal since there are a limited number of possible edit operations applicable to input graphs. However, as the graph size increases, i.e. as the graph complexity increases, we can see that method I experiences a sharp drop in performance (blue and red solid lines in Figure 7). Earlier, as a result of our analysis in Section V-C, we hypothesized that indoor topologies consist mostly of smaller, functional parts. We think that method I's performance suffers because identifying such local structure becomes more crucial to make correct predictions as the graph size increases. Method I considers the partial graph monolithically as a whole disregarding any local structure. In comparison, method II identifies frequently occurring parts of graphs and extends those. This makes the latter approach more robust to the artifacts introduced by increasing graph complexity. As an example, if the input graph contains a room with a rarely occurring category, method I would only consider graphs that includes this specific input graph in them, thus ruling out plausible candidates for prediction. This leads to poor generalization of the built spatial model for indoor environment. As opposed to this, method II works on the parts of the graph that are known to be commonplace in the data set and extends those.

A further observation is that although the method II's performance drops sharply for graph sizes 0-5, the rate of the drop slows down to a great extent for sizes 5-15. Based on our analysis of *scale-free* properties of indoor floor plan graphs laid out in Section V-B.1, this behavior may be the result of local complexity in indoor topologies remaining relatively unchanged despite increasing global complexity. By using frequently occurring subgraphs, method II exploits this invariance in local structure in contrast to method I and

maintains its correct prediction rate.

In the third experiment, we have evaluated how transferable the knowledge about indoor topologies between the two data sets despite geographical and cultural biases introduced. First, we analyzed if such a transfer is possible in general. For this reason we have plotted the histogram of room categories of both data sets, and checked how many of each data sets categories overlap with each other. Fig 6 displays both histograms where matched categories are shown in blue. The matched categories on both data sets constitute 77% of all rooms in the MIT data set and 81% in the KTH data set.

As stated earlier, we trained our algorithms on the MIT data set and testing on the KTH floor plans. The results are shown as the black lines in Figure 7. We can see that the performance suffers compared to single data set results. One reason for this is that the set of room categories in both data sets are not exactly the same. The algorithm has no way of predicting unmatched categories in the KTH data set. However despite the apparent unmatched categories and other biases, the results indicated that the models learned from the MIT data set describes the floor plan graphs of the KTH data set to a certain extent. Here again, method II performs better compared to method I. An interesting result is that, similar to single data set experiments, the performance of method II also degrades gracefully for increasing graph sizes compared to method I. We think this is again is due to method II taking advantage of local structure in graphs.

Each prediction takes on average 0.21 seconds to compute on a computer with a single core 2.6 GHz CPU and 8GB of RAM with the implementation not taking advantage of the highly parallelizable characteristic of the algorithms. The frequent subgraphs generated by gSpan algorithm is run offline and cached to file. This takes 62 and 23 seconds for the MIT and KTH data sets respectively, with a frequency threshold of 5%.

# VIII. CONCLUSION AND DISCUSSION

In this paper we have analyzed indoor environments by using two large floor plan data sets. By utilizing tools from graph theory, we have investigated certain statistical properties of indoor environments. An important take away from this analysis is that, like many natural graphs, the local structure in indoor topologies remains largely unchanged despite growing global complexity. We have argued that, the reason for this phenomenon is that indoor environments are organized in functional subunits, therefore even for high overall global complexity, local structure of floor plan graphs remain stable.

We have used our analysis to develop a method that predicts what lies in the unexplored part of the environment exploiting the structure in the data. We have compared it to a more straight forward count based method. Prediction capabilities are important for various robotic tasks where the environment is partially known or the robot has uncertain measurements about the environment. We have provided extensive quantitative results on both methods and data sets, also evaluating how well the spatial knowledge can be transferred between two data sets.

We make our data set and tools to create more data sets like it publicly available at http://www.cas.kth.se/floorplans. We believe this work will enable others in the field to build on our work more efficiently.

In the future, we plan to incorporate more properties to graphs such as geometric properties and objects in order to make more informed predictions. Furthermore, we would like to explore the idea of using machine learning techniques for structured spaces [12], since our output space does not constitute a continuous space of a certain dimension but rather a space of graphs. Exploring different domains such as regular homes and compare their properties to what we have shown in this paper is also an interesting research venue.

## IX. ACKNOWLEDGMENTS

The authors thank Emily Whiting, Seth Teller and the RVSN group at MIT CSAIL for their helpful pointer to the MIT data set. This work was supported by SSF through its Centre for Autonomous Systems and the EU FP7 project CogX.

#### REFERENCES

- A. Davids, "Urban search and rescue robots: from tragedy to technology," *Intelligent Systems, IEEE*, vol. 17, no. 2, pp. 81 –83, march-april 2002
- [2] A. Aydemir, M. Göbelbecker, A. Pronobis, K. Sjöö, and P. Jensfelt, "Plan-based object search and exploration using semantic spatial knowledge in the real world," in *Proc. of the European Conference on Mobile Robotics (ECMR'11)*, Örebro, Sweden, Sept. 2011.
- [3] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in ICRA'09: Proceedings of the 2009 IEEE international conference on Robotics and Automation. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4116–4121.
- [4] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 522–537, June 2008.
- [5] K. Sjöö, "Semantic map segmentation using function-based energy maximization," *IEEE International Conference on Robotics and Au*tomation (ICRA), 2012.
- [6] E. Whiting, "Geometric, topological and semantic analysis of multibuilding floor plan data," 2006.
- [7] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, June 1998.
- [8] P. Erdős and A. Rényi, "On random graphs. I," *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.
- [9] R. Albert and A. Barabási, "Statistical mechanics of complex networks," Reviews of modern physics, vol. 74, no. 1, p. 47, 2002.
- [10] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [11] X. Yan and J. Han, "gspan: Graph-based substructure pattern mining," in *Proceedings of the 2002 IEEE International Conference on Data Mining*, ser. ICDM '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 721–.
- [12] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, *Predicting Structured Data (Neural Information Processing)*. The MIT Press, 2007.