

Lifelong Information-driven Exploration to Complete and Refine 4D Spatio-Temporal Maps

João Machado Santos¹, Tomáš Krajník¹, Jaime Pulido Fentanes¹ Tom Duckett¹

Abstract—This paper presents an exploration method that allows mobile robots to build and maintain spatio-temporal models of changing environments. The assumption of a perpetually-changing world adds a temporal dimension to the exploration problem, making spatio-temporal exploration a never-ending, life-long learning process. We address the problem by application of information-theoretic exploration methods to spatio-temporal models that represent the uncertainty of environment states as probabilistic functions of time. This allows to predict the potential information gain to be obtained by observing a particular area at a given time, and consequently, to decide which locations to visit and the best times to go there.

To validate the approach, a mobile robot was deployed continuously over 5 consecutive business days in a busy office environment. The results indicate that the robot's ability to spot environmental changes improved as it refined its knowledge of the world dynamics.

Index Terms—Mapping, Service Robots

I. INTRODUCTION

RECENT improvements in the ability of mobile robots to operate safely in human populated environments have allowed their deployment in households, offices and public buildings such as museums and hospitals. However, the structure of these environments is typically not known a priori, which requires the robots to build their own models of their operational environments. Moreover, natural environments tend to change over time, which means that to achieve long-term autonomous operation, robots must also update their environment models as a part of their daily routine.

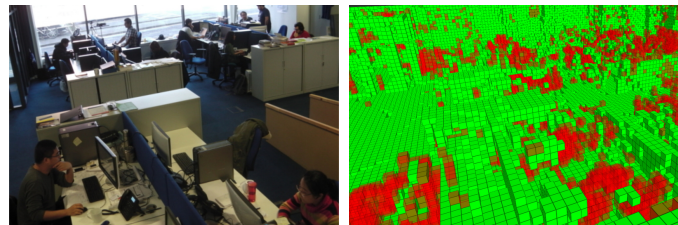
While the problem of acquiring spatial representations of the environment, known as robotic mapping and exploration, has been addressed by many researchers, building and maintaining dynamic spatio-temporal environment representations has been addressed only recently. Several recent works demonstrated that explicit representation of the environment changes improves the performance of mobile robot operation in long-term scenarios [1], [2], [3], [4], [5]. However, these works were concerned with the problem of lifelong mapping, where the environment model is built in a passive way, and not lifelong

Manuscript received: August, 31, 2015; Revised November, 26, 2015; Accepted December, 19, 2015.

This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments. The work has been supported by the EU ICT project 600623 'STRANDS'. We thank the Editor and Reviewers for their constructive comments, which helped us to improve the manuscript.

¹João Machado Santos, Tomáš Krajník, Jaime Pulido Fentanes and Tom Duckett are with Lincoln Centre for Autonomous Systems, University of Lincoln, United Kingdom jsantos@lincoln.ac.uk

Digital Object Identifier (DOI): see top of this page.



(a) L-CAS office view

(b) L-CAS spatio-temporal model

Fig. 1: Spatio-temporal occupancy grid of the Lincoln Centre for Autonomous Systems (L-CAS) office. The static cells are in green and cells that exhibit daily periodicity are in red.

exploration, where a robot decides where and when to gather data in order to complete and refine its environment model.

Our work addresses the problem of acquiring and maintaining a dense spatio-temporal environment model during long-term operation. We show that application of information-theoretic scheduling methods to time-dependent probabilistic environment representations results in a continuously improving exploratory behaviour, which evolves with the knowledge of the environment dynamics. Thus, the method allows the mobile robot to create, maintain and refine its environment models as a part of its daily routine and improve the robot's efficiency in performing other tasks at the same time. This ability is essential for long-term mobile robot operation in changing environments.

The presented work is based on a method that integrates sensory data captured at different locations and times into a dense spatio-temporal model, which represents the uncertainties of environment states as probabilistic functions of time. The probabilistic representation of the environment states allows calculation of the environment's spatio-temporal entropy, which can be used to predict the amount of information that the robot would obtain by observation of a given location at a particular time. Thus, the robot can schedule the times and locations of its observations in order to increase its chances of observing environmental changes, and thus to improve its knowledge of the environment dynamics.

To evaluate the method we compare it to a standard exploration method during 5-day-long simulated and real-world experiments performed in a human-populated environment.

II. RELATED WORK

In order to plan its actions in an intelligent way, a mobile robot needs a model of its operational environment. The

quality of its internal model has a significant impact on the robot's ability to localise itself and navigate to the desired locations. Thus, most of the previous research was aimed at developing efficient environment representations.

Two of the most popular representations are metric and topological maps. Perhaps the most known and used metric map is the occupancy grid [6]. The main drawback of occupancy grids is their low-memory efficiency since they represent large, empty areas of the environment by a large number of empty cells. The aforementioned models represent the environment as a static structure, ignoring the environment dynamics. Consequently, the localization and navigation errors will accumulate over time as the environment changes. Thus, the development of world representations that can model or adapt to the environment dynamics improves the robot performance in changing environments [7], [8].

Some authors have developed approaches that can cope with dynamics without explicitly representing them [2], [4], [9], [10]. Biswas et al. [11] present a novel non-Markov localization algorithm that classifies different types of un-mapped objects according to their dynamics, providing local and global corrections to the environment model and thus more accurate localization than systems relying on a static world assumption. Similarly, [12] presents the concept of Dynamic Pose Graph SLAM, which aims to improve robot localization in changing environments. This method builds and maintains a pose graph of the environment by detecting changes after several measurements taken at the same locations. The pose graph is then edited in order to be more consistent with the current world state.

Other authors [3], [1] have focused on models that explicitly represent the environment changes and try to identify patterns. In [1], the robot's operational environment is modelled at multiple timescales. Churchill and Newman [2] use a vision system to group several distinct observations into 'experiences' of the same spatial locations, which improves long-term mobile robot localisation in outdoor environments. Tipaldi et al. [4] represent the states of the environment components (cells of an occupancy grid) with Hidden Markov Models and show that their method improves the robustness of localization. In [13], each cell in the occupancy grid stores not only the probability of it being occupied, but also the likelihood of the cell to change after a given time.

The aforementioned works focus on solving mobile robot localization in changing environments during long-term scenarios but the mapping task itself is done passively as it does not plan the locations and times to visit. Hence these approaches do not guarantee a complete model of the environment, which is vital for other abilities crucial to a robot such as planning.

Exploration approaches are aimed at guaranteeing completeness of the model by giving the robot the ability to autonomously create an accurate model of its operational environment. These methods typically focus on creating a complete and accurate world model in the shortest possible time. In frontier-based approaches [14], [15], the frontier is defined as the boundary between the known and unknown parts of the environment, and the robot plans its path in order

to remove all the frontiers. While these strategies ensure the map's completeness, they do not consider map quality.

Next Best View strategies optimise several criteria, e.g., the estimated time to reach a given location and the amount of information expected to be gathered there [16]. Typically information gain is calculated as the environment uncertainty reduction achieved by incorporating the measurements from given positions. The reduction of uncertainty is typically calculated by means of entropy [17]. The lower the entropy of the environment model, the more it reflects the actual environment state.

Stachniss et al. [18] presented an information-gain based exploration framework that integrates not only uncertainties of the map, but also the uncertainties of the robot's localization. The method uses a Rao-Blackwellized particle filter to build the map and an entropy reduction method to plan the next location to be visited by the robot. In [19], the authors present a Next Best View approach to build a 3D model of an outdoor scenario while maximising the model's quality and optimising the robot's trajectory.

Most exploration approaches focus on building the initial map and do not deal with its maintenance over time, treating environment dynamics as unwanted noise, which means that the model loses its validity over time. To deal with this a related family of algorithms aims at creating models of the environment that allow them to predict where and when to make observations of specific phenomena within the environment. They achieve this by reasoning about the best times and paths to take. Typically, these algorithms rely on Gaussian Processes [20], [21], [22], which allow the robot to learn patterns in the environment. Other approaches represent the dynamics of the environment states based on the assumption that some of the environment variations observed are caused by routines performed by humans [23]. While the first approach focuses on building a model of sparse environmental phenomena as the main task of the robot, the latter focuses on updating a model of the environment that is used by the robot itself to improve its performance while executing its daily duties. The results [23] showed that this approach could predict the environment changes, allowing the robot to better plan where and when to perform exploration. However, in this approach the topology of the environment was known *a priori*.

Our method builds on the concept of spatio-temporal exploration presented in [23], which builds frequency-enhanced spatio-temporal models from sparse and non-uniform observations and examines the performance of various exploration strategies and dynamic models. However, the prior work in [23] was based on several simplifications that make its real-world use difficult: it assumes that the topology of the environment is known a-priori and it neglects the fact that navigating between different locations requires different time durations. In other words, in [23], the robot simply selects which pre-defined topological locations should be visited at particular times in order to create and maintain local dynamic models on top of an a-priori known topological structure. The work presented here describes a complete exploration pipeline that starts without any a-priori knowledge about robot's environment. The locations to be observed are not

selected from a pre-defined set as in [23], but the robot infers the locations from the 3D structure itself. Thus, it considers not only the information gain obtained by visiting a given location, but also its reachability and the time it takes to navigate there. This results in a life-long exploration system that allows to create and maintain global 4D spatio-temporal representations of real, changing environments without prior knowledge of their topology. This paper also includes substantial new experimental work including extensive ground-truth-based evaluations. Nevertheless, our previous work indicated that the Monte Carlo exploration strategy outperformed other strategies in terms of the resulting model accuracy.

III. EXPLORATION SYSTEM DESCRIPTION

Robotic exploration methods usually consist of two alternating phases: planning and mapping. Considering that the environment is constantly changing, both planning and mapping have to take into account the notion of time. Thus, 3D mapping has to explicitly model the environment dynamics and becomes “4D mapping”. The planning has to determine not only which locations to explore, but also when to perform the exploration. Other activities which form part of the robot’s daily routines may also be scheduled here, since a robot in a real-world application would have to balance its exploration activities with other activities that exploit the current spatio-temporal knowledge.

Our exploration system is composed of five main modules: the *Spatio-Temporal Model* that maintains the environment map, the *Scheduler* that determines the robot activity, the *Planner* that calculates which locations are to be explored, the *Executioner* that acts as a bridge between these modules, and the *Robot*’s navigation and sensing systems. The robot’s activity consists of separate exploration tours during which the robot leaves its charging station, navigates to a set of locations, where it uses its depth camera to observe the environment, and finally docks to its charging station using a precise marker-based localization method described in [24]. In this section, we first provide an overview of the exploration system and then details of its main modules.

A. System overview

The overall system structure and its most important data flows are shown in Figure 2. Every 24 hours at midnight, the *Scheduler* sets up an activity plan for the upcoming day, which is partitioned into several time slots of the same duration. To determine which time slots are to be used for exploration and which ones to use for charging, it uses the *Planner* and the *Spatio-Temporal Map* to estimate how much information would be obtained by performing exploration at each of the time slots. In particular, the *Scheduler* sends the start time of a particular time slot to the *Spatio-Temporal Map* and the number of locations to visit to the *Planner*. The *Spatio-Temporal map* then predicts the probability and entropy of the environment states for the given time and passes the model to the *Planner*. The *Planner* then generates a sequence of candidate locations to visit, queries the *Spatio-Temporal Map* for the expected information gain at those positions and the

Reachability Map for the probability that the robot will be able to navigate to those locations. The *Planner* then selects a number of locations to visit, where the number is given by the *Scheduler*, and reports the overall information gain back to the *Scheduler*. Based on the estimated information gain for each time slot, the *Scheduler* decides which time slots are to be used for exploration and which ones to use for recharging. The schedule-generation process is computationally expensive, mainly because the robot has to calculate the potential information gains across many locations and times. The entire schedule-generation process takes approximately two minutes and is performed during recharging. While the

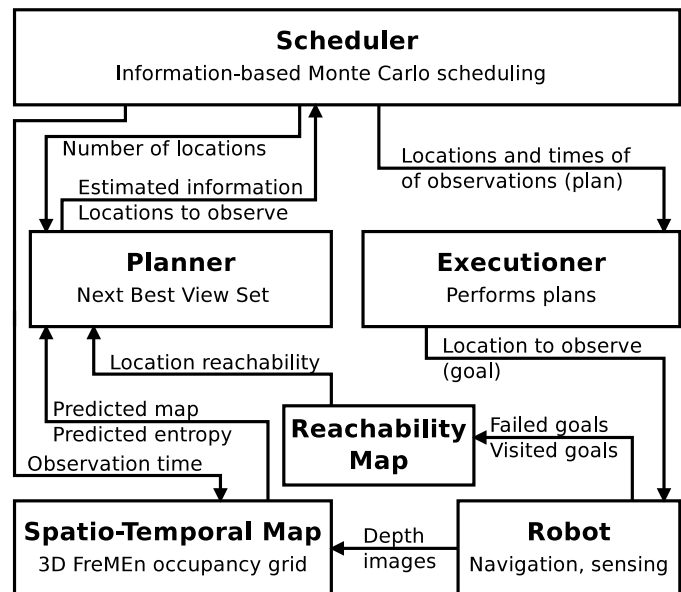


Fig. 2: Exploration system modules and main data flows

generated schedule ensures that the robot will tend to explore the environment when it is more likely to exhibit changes, the plans (sequences of points) generated by the *Planner* during the process of schedule generation might not be suitable at the time of their execution because the environment might change in a way that was not originally predicted.

Thus, at the beginning of each time slot allocated for exploration, the *Scheduler* queries the *Planner* for a new plan. The *Spatio-Temporal Map* predicts a temporary 3D occupancy grid, which is used to estimate the information gain and the *Reachability Map* for the given time. The *Planner* uses this information to decide which locations to visit and the *Executioner* determines their order and passes these goals one-by-one to the *Robot*’s navigation system. The *Robot* monitors whether the required locations (goals) were reached and passes this information to the *Reachability Map*. If a goal is reached successfully, the *Robot* uses its pan-tilt unit and depth camera to update the temporary 3D grid using the method in [25] and marks which cells were observed. After each 3D sweep, the updates made in the temporary 3D grid are propagated to the *Spatio-Temporal Map* using Equation (2).

B. Spatio-Temporal Map

The Spatio-Temporal map used in our work is based on a uniformly-spaced 3D occupancy grid extended by the Frequency Map Enhancement (FreMEn) concept [26]. The authors of FreMEn argue that the states of world models of human populated environments are influenced by human activities that – in a long-term perspective – tend to exhibit regular patterns, which causes some of the states' dynamics to be regular as well. FreMEn attempts to capture and model the periodicities by modeling the states' dynamics by their frequency spectra. In short, FreMEn uses the Fast Fourier algorithm to transform past observations of a given environment state, which is a binary function over time $s(t)$, to the spectral domain $S(\omega)$ and stores the most prominent spectral components of $S(\omega)$ in a sparsely-represented set $P(\omega)$. The inverse Fourier transform of $P(\omega)$ is then interpreted as a probability $p(t)$ of the original state $s(t)$ at time t . Application of FreMEn has been shown to improve mobile robot performance in feature-based localization [3], topological navigation [7] and robot search [8]. However, the Fast Fourier Transform algorithm used in [26] requires that the state $s(t)$ is sampled on a regular basis, which conflicts with the requirements of spatio-temporal exploration.

Thus, we propose a different scheme of transformation between the time domain $s(t)$ and the frequency domain $S(\omega)$. We assume that the spectral representation $P(\omega)$ of a state $s(t)$ consists of a small number of frequencies ω_i , phase shifts φ_i and amplitudes α_i . The probability $p(t)$ of the state $s(t)$ can be calculated as

$$p(t) = \varsigma\left(\alpha_0 + \sum_{i=1}^n \alpha_i \cos(\omega_i t + \varphi_i)\right), \quad (1)$$

where α_0 corresponds to the 'static' probability of the state $s(t)$, n is the number of periodicities modelled and $\varsigma(\cdot)$ ensures that the result of Equation (1) is bounded between 0 and 1. To reflect the fact that we cannot be absolutely certain when predicting a given state, function $\varsigma(\cdot)$ limits the $p(t)$ between 0.05 and 0.95.

To obtain the parameters ω_i, φ_i and α_i from m measurements of the state s taken at times t_k , we first calculate the value of α_0 as an arithmetic mean of all past observations $s(t_k)$. Then we create a set of candidate frequencies Ω , which represent the periodicities of the hidden processes that affect the state $s(t)$. Finally, we establish the amplitudes α_c and phase shifts φ_c as

$$\begin{aligned} \alpha_c &= \left| \sum_{k=1}^m (s(t_k) - \alpha_0) e^{-j2\pi t_k \omega_c} \right|, \\ \varphi_c &= \arg\left(\sum_{k=1}^m (s(t_k) - \alpha_0) e^{-j2\pi t_k \omega_c} \right), \end{aligned} \quad (2)$$

where ω_c are elements of the set Ω .

Then, we order the frequencies ω_c according to their amplitude α_c , select the first n of them and store these as parameters ω_i, φ_i and α_i , which are used in Equation (1). Note that unlike the traditional Fast Fourier Transform used in [26], Equation (2) allows to update the spectral model as new observations of the state $s(t_k)$ are obtained. While faster to calculate and allowing for non-uniform sampling, the proposed representation does not ensure precise reconstruction of the original sequence $s(t)$, but typically results in $\sim 2\%$

reconstruction error. For details, see our previous work on frequency-based representations presented in [26] and [23].

Our *Spatio-Temporal Map* applies the FreMEn concept to occupancies of cells in a 3D occupancy grid. Thus, each cell contains its own set $P(\omega)$ that allows to calculate the probability of the cell's occupancy for any given time. This model is updated by Equation (2) every time the cell is observed and its state $s(t_k)$ is measured.

Both Equations (1) and (2) are derived from the continuous formulation of the Fourier transform in [27], but unlike the classic discrete Fourier transform (DFT), Equations (1) and (2) simply do not assume time-uniform sampling of the state $s(t)$. In the case of uniform sampling with period Δt , i.e. $t_k = k\Delta t$, Equation (2) would become equivalent to the standard DFT.

1) *Temporal model design*: The set Ω of candidate frequencies in Equation (2) defines which periodicities will potentially be captured by our model. The elements of Ω can be chosen arbitrarily, but one should consider that larger Ω enables a finer representation of time at the expense of higher memory consumption of the spatio-temporal model. In our experiments, the set Ω consist of 24 elements ω_i , which are distributed in the same way as in the traditional FFT, i.e. $\omega_i = (24 \times 3600)/i$. This allows to model several periodicities ranging from one day to one hour. To model spatio-temporal dynamics of office-like environments, one could extend the set Ω by adding day-to-week periodicities as in [23]. However, this would require continuous operation of the robot for several weeks, which we hope to achieve during 2016.

The parameter n in Equation (1) determines how many periodicities of Ω are actually considered in the state prediction. Our previous work indicates that a good choice of n is 2, which typically results in modelling week- and day-long periodicities in indoor environments and year- and day-long cycles outdoors. Note that setting n to 0 means that the probability $p(t)$ becomes a constant as in traditional spatial-only representations. Similarly, non-periodic dynamics will cause the coefficients α_i calculated by Equation (2) to be close to 0, which will cause the $p(t)$ to be almost constant as well.

C. Predicting the information gain

Since the aforementioned model can predict the probability of each cell being occupied, it also allows to estimate the amount of information that the robot obtains by observing the particular cell at a given time. The amount of information $I(t)$ obtained by observing a single cell at time t can be calculated as the difference between the cell's a-priori entropy $E(t)$ and a-posteriori E_r entropy, i.e. $I(t) = E(t) - E_r$, which are functions of the cell's occupancy probability before and after the observation. Since the cell's occupancy probability is considered as a function of time and we assume that the robot observes a given cell long enough to determine its state with certainty $p_c = 0.95$ (i.e. the probability of the cell being occupied after the observations becomes either 0.05 or 0.95), the expected information gain at time t is

$$I(t) = -p(t) \log_2 p(t) - (1 - p(t)) \log_2 (1 - p(t)) + p_c \log_2 p_c + (1 - p_c) \log_2 (1 - p_c), \quad (3)$$

where $p(t)$ is the probability of occupancy of a given cell at time t calculated by Equation (2). Using the predicted occupancies and entropies, the *Spatio-Temporal map* allows to estimate the amount of information that the robot will obtain by observing a particular part of the environment at a particular time using its depth camera. Since our robot uses its pan-tilt unit to create a 360° ‘sweep’ of its surroundings, the *Spatio-Temporal Map* implements a function that can estimate the obtained information given the robot position and the time of observation.

D. Reachability map

Although the ability of the robot to reach individual locations of the environment can be inferred by the *Planner* from the environment’s spatio-temporal representation, some locations might not be reachable due to factors that are not included in the spatio-temporal model, such as transparent obstacles or objects with dimensions smaller than the spatio-temporal grid resolution. To reflect that, the exploration system maintains a *Reachability Map*, which is a 2D (50×50 cm) grid with cells that contain the robot’s success rate over the last five attempts to reach that particular location. This information is taken into account when the exploration plans are calculated.

E. Locations to observe

We assume that moving to and observing one location takes approximately two minutes. Taking into account the time needed to dock to and leave the charging station, the robot can visit 6 locations in a 15-minute time slot.

To determine which locations are to be visited during a given time slot, the *Planner* first generates a uniform 2D grid of candidate positions $x_i, y_i \in \mathcal{C}$ that cover the operational environment. Then, it sends these positions to the *Reachability Map*, which returns the probability that the robot will be able to reach these positions, i.e. the *Planner* will obtain a reachability $p_r(x_i, y_i)$ for each candidate location (x_i, y_i) . If a position (x_i, y_i) is reachable, i.e. $p_r(x_i, y_i) > 0$, the *Planner* forwards the position (x_i, y_i) to the *Spatio-Temporal Map*, which uses the predicted 3D grid to estimate which cells are likely to be observable by the robot’s depth camera from the position (x_i, y_i) . The *Spatio-Temporal Map* sums the information gain of these cells using Equation (3) and reports it to the *Planner* as $I_c(x_i, y_i)$. This allows the *Planner* to create an evaluation $E(x_i, y_i)$ of each candidate location as

$$E(x_i, y_i) = p_r(x_i, y_i)I(x_i, y_i). \quad (4)$$

Once Equation (4) has been calculated for every (x_i, y_i) , the *Planner* starts to generate the locations to visit. First, the *Planner* finds the global maximum $E_{max}(x_j, y_j)$ of $E(x_i, y_i)$, adds (x_j, y_j) and E_{max} to the set of goals \mathcal{G} and sets $E(x_j, y_j)$ to 0. To take into account the fact that the cells observable from (x_j, y_j) are also visible from neighbouring locations but observations at locations close to (x_j, y_j) would not provide the same expected information, the values of $E(x_i, y_i)$ in the vicinity of (x_j, y_j) are decreased proportionally to their proximity to (x_j, y_j) . The aforementioned two steps, i.e. maxima search and suppression of the information gain

estimates at the neighbouring locations, are repeated until the number of goals in the set \mathcal{G} equals the number of locations requested by the *Scheduler*. Then, the *Planner* calculates the sum E_G of information gains $E_{max}(x_j, y_j)$ in \mathcal{G} and reports the value of E_G to the *Scheduler* along with the locations in \mathcal{G} .

F. Generating the schedule

Once the *Scheduler* obtains the summarised information gain E_G for every time slot using the aforementioned procedure, it uses a Monte-Carlo-based method to determine which time slots to use for exploration and which ones to use for charging. Thus, the probability that a given time slot will be selected for exploration is proportional to its expected information gain E_G . The generated schedule is then saved and the *Scheduler* is deactivated until the start of the next time slot.

At the beginning of each time slot, the *Scheduler* checks whether the time slot was allocated for exploration and eventually queries the *Planner* for an up-to-date plan for the given time. Then, it forwards the set of locations to observe to the *Executioner*.

G. Plan execution

The *Executioner* module is responsible for carrying out the plan provided by the *Scheduler*. At first, the *Executioner* uses a 2-opt method [28] to establish a sequence in which the planned locations should be visited. Then, it ensures that the robot leaves its charging station, follows the given path while taking measurement at the given locations and returns back to recharge. If the *Executioner* fails to reach a given location, which is typically caused by the location being blocked, it first waits for the location be cleared. If the location remains unreachable, the *Executioner* simply proceeds with the following location in the plan. After each run, the *Executioner* reports the successes or failures in reaching the planned locations to the *Planner*, which updates the *Reachability map*. This causes the robot to avoid areas that are more likely to be blocked. However, the amount of obtainable information for the neighbouring cells is likely to be high, causing the robot to perform observations in nearby locations in the next exploration run.

H. The robot

The platform used in this paper is a SCITOS-G5 mobile robot equipped with RGB-D cameras and a laser rangefinder. The robot’s navigation system is based on open-source, freely available software developed during the STRANDS project [29], which extends the navigation stack of the Robot Operating System (ROS). The sensor that was used for 4D mapping presented in this paper was the Asus Xtion RGB-D camera, which was mounted on a pan-tilt unit placed on top of the robot’s head. Using this pan-tilt unit, the robot created 360°×90° 3D sweeps with a 4 m radius at locations it was supposed to observe.

IV. EXPERIMENTAL EVALUATION

The most popular metrics used to evaluate static exploration methods are the completeness of the robot's environment model and the time or travel distance required to complete the exploration. Since spatio-temporal exploration is a continuous process, these metrics are not applicable. Another quality metric lies in comparison of the ground truth with the robot's internal environment model.

Thus, we propose to evaluate the exploration algorithm by two different metrics. The first metric is based on the amount of changes observed by the exploration algorithm. This metric is calculated directly as the number of cells that change their states through direct observation during an exploration tour. One would expect that a better spatio-temporal exploration algorithm would be able to observe more changes, because it can use its predictive capabilities to determine which areas are more likely to change and direct the robot's attention to these locations.

The second metric is based on the accuracy of the created model, which is calculated by comparing the model with the ground truth. However, obtaining a complete ground truth would require continuous observation of all environment locations, which would necessitate an extensive infrastructure. To overcome this limitation, we built a simulated environment, where ground truth can be obtained relatively easily and performed the ground-truth comparison in the simulated environment. In the real-world experiment, the ground-truth comparison is performed on a limited set of regions around the researchers' workplaces.

A. Experiment description

To evaluate our spatio-temporal exploration algorithm we compare it against a method that considers a static environment model. This 'Spatial-only' exploration method is equivalent to state-of-the-art information-theoretic next-best-view exploration methods, such as [16].

To compare these two methods, a robot platform running the system described in Section III was deployed in both a simulated and real-world office for 5 business days.

Every midnight, the *Scheduler* generated a schedule for the following day. This schedule was composed of 15-minute-long time slots, of which 48 were exclusively allocated for the spatial-only (SO) and 48 for the spatio-temporal (ST) exploration algorithm. Since each method had to use half of its allocated time slots to replenish the robot's batteries, the robot performed 24 exploration tours guided by the spatio-temporal method and 24 tours guided by the spatial-only method per day.

Both methods operated as described in Section III. The only difference between them was that the ST method used the predicted (by the *Spatio-Temporal Model*) map while the SO method used the last obtained map. We hypothesize that the use of a predicted map should allow the *Scheduler* to determine when it is more likely to obtain more information and schedule more exploration tours at the times when the office is more likely to be occupied. Moreover, the *Planner* should be able to predict which areas of the environment are

likely to change at a particular time and take this into account when generating the locations to explore.

B. Real-world experiment

The real-world experiment was performed in an open-plan office of the Lincoln Centre for Autonomous Systems (L-CAS). The office consists of a kitchenette, a lounge area and 20 working places that are occupied by students and postdoctoral researchers. During the experiment, two ceiling cameras were used to capture a time-lapse video of the office dynamics, which allowed not only for a location-based ground truth comparison, but also to build a database of the office dynamics.

After five days of exploration, we calculated the amount of changed cells that were observed by the two aforementioned strategies during the individual exploration tours. The Figure 3

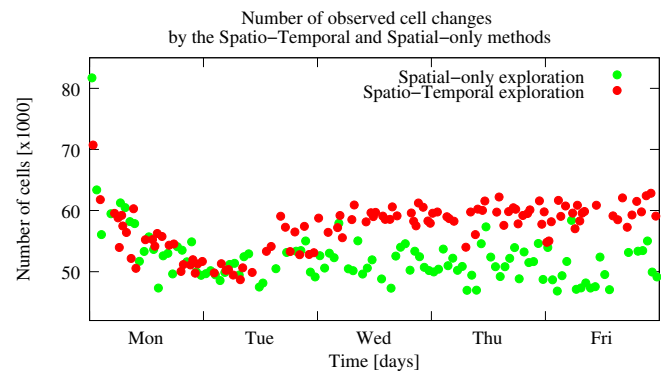


Fig. 3: The number of observed occupancy changes by the Spatio-Temporal versus the Spatial-Only exploration methods.

shows that at the start of the exploration process, the number of cells that changed their state was high, but gradually decreased as the environment structure became known. After the first day, the amount of changes observed by both methods tended to stabilize around a value given by noise and the environment dynamics. During the second day the Spatio-Temporal method would start to identify the daily routines and the *Planner* would guide the robot to locations that are more likely to exhibit changes – see Figures 1 and 4 for the spatio-temporal map obtained after the first two days of the experiment. After the second day, the Spatio-Temporal method would allocate more exploration tours to the afternoon, when the office is more likely to be populated. In fact, there were 30% more tours scheduled for the afternoon than for the morning.

In the last three days of the experiment, the Spatio-Temporal method observed more changes than the Spatial-Only one, due to its ability to identify the locations and times of environmental change. In other words, the Spatio-Temporal exploration method could plan better **where** and **when** to explore.

To establish the accuracy of the models created, we selected six working locations in the office, see Figure 4, and manually established the presence of people at these locations over time. Then, we used the Spatio-Temporal models built by the two exploration strategies to predict the overall occupancy

of these areas (see to Figure 4) for every hour of the five-day experiment. Then, we compared these occupancies to the ground truth provided by hand-annotated people presence. This allowed us to calculate the error of each model in the

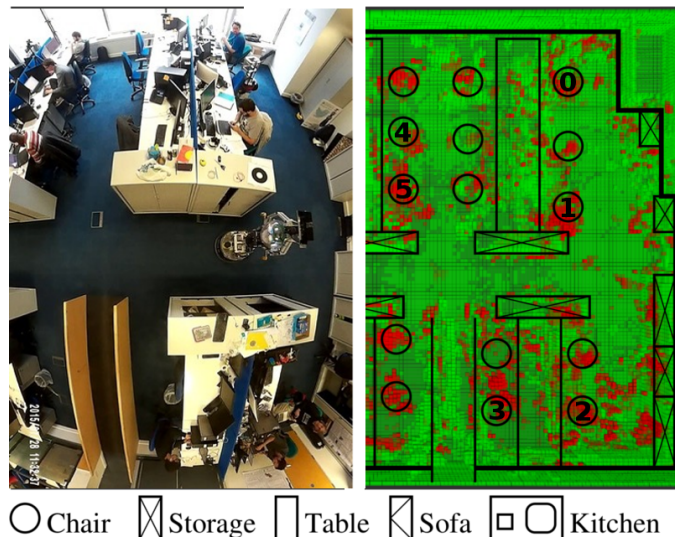


Fig. 4: The layout, the spatio-temporal occupancy grid and top camera view of the Witham Wharf office. The static cells are in green and the cells that exhibit daily periodicity are in red. The locations for ground-truth evaluation are marked with numbers.

same way as in [23], i.e. as an average deviation from the ground truth during the experiment. Table I indicates that part

Model type	Location						Avg	StD
	0	1	2	3	4	5		
SO	28	23	43	23	21	29	28	8
ST	20	23	25	19	17	14	20	4

TABLE I: Overall error of the environment model [%]

of the dynamics of these locations can be explained by periodic processes related to human activity. The researchers working at these six places had diverse working habits, which caused the error rates to vary across the individual locations. Performing a paired t -test indicates that the error of the ‘Spatio-Temporal’ environment model is significantly lower than the error of the ‘Spatial-Only’ method.

C. Simulated experiment

To speed up testing and to allow for a more representative ground-truth comparison, we created a 3D MORSE-based simulation of our office.

Moreover, we created a software component that allows to reconfigure the simulated environment on the fly. Using the data gathered for five days by two ceiling cameras, we created a database that contains the positions and presence of twenty dynamic objects over time. Combination of the reconfigurable simulator with the aforementioned database allowed us to create a realistic simulation that reflects the real-world dynamics.

Thus, our simulator does not only reflect the environment static structure, but also simulates dynamic elements, such as people, chairs, laptops and doors (see Figure 5). The experiment was performed in the same way as in the real environment. The number of changed cells captured by both the Spatio-Temporal and Spatial-Only algorithms followed a similar pattern as in the real-world experiment. The main advantage of the



Fig. 5: Snapshot of the simulated environment.

simulation was the possibility to obtain ground truth that spans the entire space and time of the experiment. The ground truth for a single time slot was obtained by configuring the simulation for a particular time and letting the robot perform its 3D sweeps at several locations in order to obtain a complete overview of the environment. This was repeated for every time slot of the experiment, obtaining 480 static 3D grids that represent the environment’s evolution over time. The error of

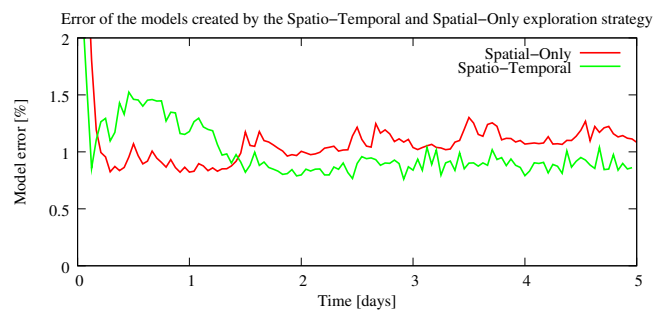


Fig. 6: The ratio of incorrectly estimated cells for the Spatial-Only and Spatio-Temporal strategies.

a particular model at a given time is calculated as the number of cells whose states differ from the ground truth divided by the total number of observed cells.

To compare the performance of the SO and ST models, we calculated their errors for each time slot and performed a t -test of the error values for each experiment day. The results of the t -tests indicate that during the first day, the ST model performed significantly worse than the SO one. During the second day, the ST model started learning the periodic patterns, which improved its performance, and the ST and SO model errors were not significantly different. The t -tests of the third, fourth and fifth day show that during these days, the ST model error was significantly lower than the SO one. These

Copyright Notice

This is a submitted version of the paper:

João Machado Santos, Tomáš Krajník, Jaime Pulido Fentanes, Tom Duckett: *Lifelong Information-driven Exploration to Complete and Refine 4D Spatio-Temporal Maps*, In IEEE Robotics and Automation Letters.

The full version of the article is be available on IEEE Xplore or on request. For questions or requests, email jsantos@lincoln.ac.uk.

Copyright 2377-3766 ©2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. We are happy to grant permission to reprint these images, just send us a note telling us how you plan to use it. You should also request permission from the copyright holder, IEEE, at the copyrights@ieee.org address listed above.