2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Optimal Placement of Passive Sensors for Robot Localisation

Fabiano Zenatti, Daniele Fontanelli, Luigi Palopoli, David Macii, Payam Nazemzadeh

Abstract— **We consider the problem of self-localisation for a mobile robot in an environment with a requested level of accuracy. The robot moves in a known environment following typical trajectories, which can be characterised in statistical terms. One of the main drivers of this paper is its application to assistive robots guiding senior or impaired users in shopping centres or in other public spaces. To localise itself the robot uses onboard sensors such as encoders and inertial platforms. The level of noise in these sensors and the lack of absolute measurements determines a steady growth of the uncertainty on its position. To alleviate the problem, we assume the presence of a number of visual markers deployed in the environment. Whenever the robot comes across one of these sensors, the uncertainty on its position is reset. In the paper, we show a methodology to minimise the number of these sensors and to select their position so that the uncertainty is never worse than a given target threshold with an assigned probability.**

I. Introduction

In its general terms the problem we address in this paper is described as follows. A set of robots of the same kind move in a known indoor environment and their motion is observed and recorded. Such robots rely on a localisation mechanism combining their on-board sensors with occasional readings of landmarks deployed in the environment at known positions. The on-board sensing system comprises wheel encoders and an IMU (i.e., accelerometers and gyroscopes). As each robot moves in the environment, the dead-reckoning integration only based on on-board sensors accumulates uncertainty on the robot absolute position [1], [2], which is reduced every time the robot detects these landmarks. Since the landmark infrastructure is expensive to deploy and maintain, we have strong motivations to find a solution to the following problem: *find the minimum number of visual markers and identify their optimal positioning so that the uncertainty accumulated by the robots remains below a threshold with a given probability*.

Several instances of this problem occur in different application domains. The specific motivation of our paper has to be sought in the European ACANTO research project [3]. The goal of this project is to develop a robotic walking assistant (the *FriWalk*) to help older adults with mild mobility problems navigate and use complex public environments. Key to a successful fulfilment of the robot tasks is its accurate localisation within the environment [4], [5]. Since the robot is chiefly targeted at indoor applications, GPS localisation is hardly a viable solution. For this reason localisation is implemented using a combination of sensors on board of the robot with external markers such as QR codes [6]. For the sake of completeness, we give in the paper a description of the ACANTO specific case study, but the techniques developed here could easily be applied to other robotic systems (e.g., mobile robots used for transportation in an industrial compound). We will use the term "target" to refer to the robot that we need to localise.

An overview of the approach used in this paper is now proposed. We start from a collection of data on the trajectories followed by the target in the environment. Using this information, we derive a stochastic abstraction that reproduces the trajectories with a reasonable approximation. A stochastic abstraction is, in this context, a discrete–time Markov chain, in which states are associated with the "typical" locations visited by the target, and transitions describe the motion between two locations resulting from the observed trajectories and are annotated with a probability accounting for the frequency with which the transition is observed. The use of this type of stochastic abstractions is frequent in the literature [7], [8], [9]. In our case, a transition is also associated with a level of uncertainty, which accounts for the loss of localisation accuracy incurred in moving between start and end locations without using landmark information. In the paper, we show how to cast the problem of finding the minimum number of QR markers and their optimal position into the framework of mixed binary linear programming (MBLP). An important theoretical result of the paper is that the problem can be addressed with a MBLP of limited size even in the presence of cyclic paths.

A. Related work

The problem of sensor placement for localisation has been studied in the literature in the last two decades with particular attention placed on active sensing systems [10], [11]. While some of the solutions proposed consider the active sensors as moving agents in the environment [12], [13], the majority of the solutions refer to a static deployment.

The placement of the markers in an environment in a framework similar to the one presented was first addressed by Nazemzadeh et al. [6], [14], with a sub-optimal solution derived for the placement of markers on fixed positions of a regular grid. The objective of the placement was simply to find the minimal distance between two nodes in the grid that guarantees a specified level of accuracy. Very similar are the

This paper has received funding from the European Unions Horizon 2020 Research and Innovation Programme - Societal Challenge 1 (DG CONNECT/H) under grant agreement n° 643644 "ACANTO - A CyberphysicAl social NeTwOrk using robot friends".

D. Fontanelli and D. Macii are with the Department of Industrial Engineering (DII), University of Trento, Via Sommarive 5, Trento, Italy daniele.fontanelli@unitn.it. L. Palopoli, P. Nazemzadeh and F. Zenatti are with the Department of Information Engineering and Computer Science (DISI), University of Trento, Via Sommarive 5, Trento, Italy palopoli@disi.unitn.it

ideas proposed by Beinhofer et al. [15], who use linearised dynamics to find the minimal distance between the markers that satisfies the application requirements with a good confidence. In this paper we take into explicit consideration the statistic profile of the motion of the target in the environment. This prevents us from wasting a QR marker on an area that is never (or very infrequently) visited. Moreover, our use of a discrete stochastic abstraction decouples the problem of choosing the candidate location (which are not constrained to be on a grid) for the nodes from the optimisation algorithm.

An appealing direction is human-aware robot planning [16], [17], [18]. Indeed, the use of graph abstractions is very frequent in algorithmic solutions for people tracking in large indoor environments. The environment is broken down into points of interest (POI) using empirical observations and learning algorithms. The POI are usually modeled as nodes on the graph, where the arc represent the typical paths joining two points of interest. The nodes are often called *sub-goals*, with the goal being represented by the long-term target position which is reached through a sequence of subgoals. A description of this kind, where the graph topology is from statistical data of pedestrian motion is presented by Kuipers et al.[19]: the topological nodes represent physical point where humans take decisions. In a different view, more grounded to robot localisation, identifies decision points as places where humans usually stop [7], [8]. A stochastic abstraction similar to the one adopted in this paper is proposed by Burgard et al. [9], who model the navigation behaviour of humans using a probability distribution over the trajectories and generates a graph of nodes with associated probabilities (essentially a discrete-time Markov chain). A similar idea is by Kirchner et al. [20], who champion the use of observed position trace and environmental data into a probabilistic framework for path planning. We have just named a few examples of derivation of stochastic abstractions that can be readily treated by our optimisation framework.

A more direct link with the problem of optimal sensor placement can be found in the work of Lin et al. [21], who apply active sensing to the problem of intruder detection. Contrary to our work, the authors do not consider neither the problem of uncertainty growth, typical of self localisation with dead–reckoning, nor the probability of detecting the target. Another related solution is proposed by Jourdan et al. [22] that tackles the optimal sensor placement for ranging sensors. Even though the problem is related to the one proposed here, the objective of their solution is to find the best sensor locations in order for the target to be always in view from at least two sensors.

II. Problem Description

Robot localisation usually relies on the combination of two types of complementary techniques: dead reckoning (for ego motion) and measurements of absolute position and heading (from external sources or detecting environmental markers). The rationale underlying the proposed approach is related to the features (and the limitations) of different kinds of indoor localisation techniques available in the ACANTO project context, nevertheless the basic concepts are of general applicability. The dead reckoning solutions are fast and generally require low-cost sensors, thus enabling high-rate position tracking without the need for infrastructures in the environment. However, the initial values of heading and position must be known, because they are unobservable. Moreover, the positioning uncertainty tends to increase as the travelled distance grows. The problems above can be addressed if accurate and absolute heading and position measurements are used to adjust occasionally the results provided by dead reckoning techniques. However, such absolute measurements require some external system or infrastructure able to estimate the distance/orientation between the robot and a set of known reference points. Since the GPS signals can be hardly received inside buildings and the wireless ranging techniques suffer from poor accuracy indoors, alternative solutions with a similar function, but on a local scale, have to be devised. General essential requirements for the proper combination of dead reckoning and absolute positioning techniques are: low deployment costs, scalability, robustness to unexpected events (e.g. obstacles or other moving targets) and, of course, adequate accuracy for the intended application. For the aims of ACANTO, positioning uncertainty must be kept below a stringent target value (i.e. some tens of cm) with a high level of confidence. To achieve this goal, using sensors for absolute position and heading estimation with a limited reading range can be advantageous, since this choice prevents to incur ambiguous conditions and interferences due to the simultaneous detection of multiple reference points. Also, a limited reading range provides an upper bound to distance measurement uncertainty. For such reasons, the absolute positioning approach proposed in ACANTO relies on both odometry and a vision system detecting one of the Quick Response (QR) codes stuck on the floor (or on the ceiling) of the chosen environment. A qualitative overview of the proposed approach is sketched in Fig. 1. This solution is effective for a variety of reasons. First of all, each QR code can be easily and univocally associated with a triple of values (x, y, θ) denoting the planar coordinates of the detected landmark on a given map and its orientation with respect to a known direction. Second, several standard and well-established algorithms, libraries and software tools exist to detect QR codes and the distance between the camera and one of them. Third, the detection range (even if strongly dependent on the size and the resolution of the codes) can hardly exceed 2 m if the side of the QR code is in the order of a few tens of cm. Third, as discussed in a different paper [23] QR codes can be arranged in geometric configurations such that a robot comes always in sight of a marker when moving between nearby positions. When multiple markers are in sight, we can easily select the one which is closest to the robot. Fifth, the cost of a QR code sticker is extremely low. Nonetheless, deploying many of them in large and wide rooms can be expensive, time-consuming and with a significant visual impact. Therefore, a problem that spontaneously arises from this scenario is the following: how and where the QR codes should be placed in order to minimize

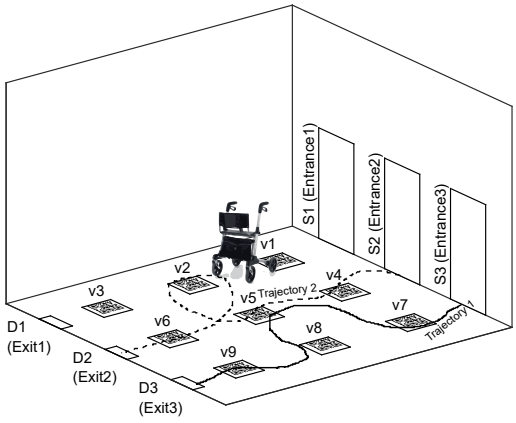

Fig. 1. Qualitative overview of the localisation technique based on the use of odometry and QR codes detected by a camera.

the total number of landmarks, while meeting the wanted accuracy boundaries? In this paper, this optimisation problem is formalised through a stochastic discrete abstraction. The same abstraction can be also easily applied or adapted to similar localisation problems where other passive devices (e.g. RFID tags [6]) are used to adjust robot's position estimated through dead reckoning techniques. The core of the proposed abstraction relies on an oriented graph including three sets of nodes, i.e. the *source nodes* $\mathcal{S}=\{S_1,\ldots,S_L\}$, the *intermediate nodes* $\mathcal{V}=\{V_1,\ldots,V_N\}$ and the *destination nodes* $\mathcal{D}=\{D_1,\ldots,D_G\}$. The general properties of such sets of nodes are shortly summarised below:

- The source and destination nodes can have just outgoing or incoming edges, respectively;
- Position and heading at both source and destination nodes are assumed to be known or measured with negligible uncertainty;
- At time 0, the robot is assumed to be located in any one of the source nodes with a given probability;
- Every destination node is reachable from at least one source node;
- Each intermediate node $V_i \in \mathcal{V}$ represents the position of one of the possible landmarks to be deployed in the environment. Therefore, a binary variable $b(V_i)$ is associated with each node. This variable will be set equal to 1 if a device is located in V_i or 0 otherwise (further details on the use and expression of $b(V_i)$ will be reported in Section III).
- A node V_j will be referred to as a *successor* of node V_i for $i,j=1,\ldots,N$ and $i\neq j$, if the probability $\rho_{i,j}$ that the V_j is visited immediately after V_i (namely in one step) is larger than zero.

As a consequence of the assumptions above, the resulting system can be regarded as a discrete–time absorbing Markov chain. Even though the step duration in this case is not constant (as it depends on the actual path of the robot) this is not a issue for the solution of the optimisation problem considered. Note that the set of graph edges $\mathcal{E}=\{e_1,\ldots,e_L\}$ includes all the possible transitions between pairs of successor nodes V_j and V_i with probability $\rho_{i,j}>0$, for $i,j=1,\ldots,N$ and $i\neq j$. In the destination nodes instead just self-transitions are possible (i.e. their probability is equal to 1), since they are absorbing states. Of course, the values of $\rho_{i,j}$ depend on various factors, i.e. the model describing the motion of the robot in the environment, the reading range of the vision system and the distance between the landmarks. In the rest of this paper, the two following assumptions are made:

1) The motion of the robot is based on a random walk model and it is constrained only by the walls or fixed obstacles of the room. So no preferential directions are followed by the robot;
2) The distance between pairs of successor nodes is slightly larger than the reading range of the vision system. This implies that the probability that two successor nodes are physically quite faraway is not zero.

It is worth emphasising that these assumptions do not affect the general validity of the model, but they make the example described in Section IV consistent with the requirements of the ACANTO project.

A final important remark about the proposed abstraction concerns with the positioning uncertainty due to dead-reckoning. In particular, the positioning uncertainty accumulated over the path between successor nodes V_j and V_i can be described by a random variable $\Delta_{i,j}$. In theory, $\Delta_{i,j}$ is a continuous random variable. However, for the sake of simplicity and without loss of generality, we can assume that it is multiple of a "base" quantity $\overline{\Delta}$ (for instance due to the minimum resolution of the measurement system). Under this assumption, the probability density function of each variable $\Delta_{i,j}$ can be approximated by a probability mass function (PMF) $\mathcal{P}_{i,j}$.

III. Optimal sensor placement

As discussed in the previous section, our problem is based on the definition of three sets of nodes: the source nodes $\mathcal{S}$ (cardinality L), the intermediate nodes $\mathcal{V}$ (cardinality N) and the destination nodes $\mathcal{D}$ (cardinality G). The transitions between the nodes are associated with a conditional probability. If $\pi_k \in \mathbb{R}^{L+N+M}$ denotes the vector of probabilities of finding the target in one of the nodes, we can express the transition across one step using the customary matrix notation: $\pi_{k+1}=\pi_k M$, where M is the transition matrix. The initial probability π_0 is a vector in which only the elements associated with the source nodes can have a value different from 0. In the discussion below, we will also denote by $\rho(U_i,U_j)\overline{\Delta}$ the uncertainty associated with the transition from U_i to U_j, where U_i, U_j can be any node in the set $\mathcal{S}\cup\mathcal{V}\cup\mathcal{D}$ and $\rho(U_i,U_j)\in\mathbb{N}$ is a random variable. We will make the reasonable assumption that these variables are independent for different choices of the nodes U_i, U_j and denote by $\Gamma_{U_i,U_j}(.)$ the probability mass function of $\rho(U_i,U_j)$: $\Gamma_{U_i,U_j}(d)=\Pr[\rho(U_i,U_j)=d]$. The values of the

uncertainty for a node with a sensor will be a value $\tilde{\Delta} = c\overline{\Delta}$, with c being a constant given by the type of sensor we are using. For the sake of simplicity, we will henceforth imply the use of the granularity $\overline{\Delta}$ and talk about the uncertainty as if it was a natural number.

We can now summarise our definition of discrete abstraction:

Definition 1: A discrete stochastic abstraction for the system is a discrete–time Markov Chain (DTMC) consisting of a tuple $\mathcal{D} = \{\mathcal{V}, \mathcal{S}, \mathcal{D}, \pi_0, M, \mathcal{G}\}$, where the function $\mathcal{G}$ is a mapping that associates the pair of nodes U_i and U_j with the PMF $\Gamma_{U_i,U_j}(.)$ of the uncertainty accumulated across the transition from U_i to U_j.

Each intermediate node V_i is associated with a binary variable $b(V_i)$ that is 1 if the node is covered by a sensor and 0 otherwise.

The assignment of a set of boolean variables $b(V_i)$ determines the evolution of the uncertainty, which is described at each step k by the PMF $p_k(j) = \Pr\left[\Delta_k = j\overline{\Delta}\right]$. Our problem can then be stated in the following terms:

$$\min \sum_{b(V_i)} b(V_i) \quad s.t. \quad \sum_{j=0}^{H} p_k(j) \geq \mu, \forall k \tag{1}$$

In essence, we want to minimise the number of sensors, such that at all steps the probability of having an uncertainty below the threshold $H\overline{\Delta}$ is at least μ. In this section, we will see how to compute $p_k(j)$ as a function of $b(V_i)$. This will enable us to cast the optimisation problem above into the framework of mixed boolean linear programming [24] via some simple manipulations. Finally, we will discuss the existence of a maximum bound K for the step k, such that if the constraints in (1) are enforced up to K then it is implicitly verified for all $k \geq K$.

A. Computation of the probability mass function $p_k(j)$

In order to explain how the probability terms $p_k(j)$ can be determined, let us consider the very simple graph, consisting of just 5 nodes, shown in Fig. 2(a). The numbers next to each edge represent the transition probabilities and the worst-case uncertainties (between brackets) associated with each transition. Starting from the single source node, we can define a Decision Tree (DT), which is a tree that specifies possible sequences of choice. Each node of the tree corresponds to a path that can be taken by making a sequence of choices in traversing the DTMC. Denote by $\mathcal{T}_k = \{n_{k,1}, n_{k,2}, \dots, \}$ the nodes of the DT at depth k (i.e., after k transitions). We introduce two families of functions: $\sigma(.)$ and $\phi(.)$ that have $\mathcal{T}_k$ as domain. $\sigma(n_{k,h})$ is the ordered sequence of k locations traversed by the path from the root node to the DT node $n_{k,h}$, and we will denote it by a sequence of symbols enclosed between angular brackets, e.g. $\langle S_1, V_1, V_5, V_7, D_2\rangle$. $\phi(n_{k,h})$ represent the probability of the path associated with the node $n_{k,h}$.

Example 1: If we refer to the example in Figure 2(a), the corresponding DT is shown in Figure 2(b). For the sake of clarity, we mark each node with the last location of the path it is associated with. After one transition, we have $\sigma(n_{1,1}) = \langle S, V_1\rangle$ and $\phi(n_{1,1}) = 0.2$, $\sigma(n_{1,2}) = \langle S, V_2\rangle$ and $\phi(n_{1,2}) = 0.8$. After two transitions, we have $\sigma(n_{2,1}) = \langle S, V_1, V_4\rangle$ and $\phi(n_{2,1}) = 0.2 \times 0.4$, $\sigma(n_{2,2}) = \langle S, V_1, V_3\rangle$ and $\phi(n_{2,2}) = 0.2 \times 0.6$, $\sigma(n_{2,3}) = \langle S, V_2\, V_3\rangle$ and $\phi(n_{2,3}) = 0.8 \times 0.43$, etc. When a transition reaches a destination node, which by our assumption is associated with a self transition with probability 1, the same node will be repeated throughout the next transition with the same probability.

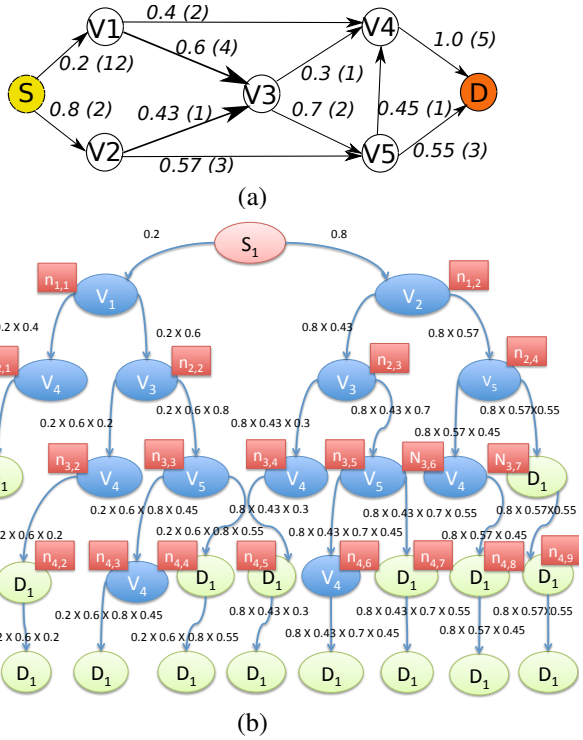

Fig. 2. (a) Graph example and (b) corresponding Decision Tree (DT).

The transition system just introduced governs the evolution of the PMF $p_k(j)$ for each configuration of the vector of boolean variables $b(V_1), b(V_2), \dots, b(V_n)$. In the following discussion, we will use B_i to mean $b(V_i) = 1$ and $\overline{B}_i$ to mean $b(V_i) = 0$. The logical conjunction of the boolean variables B_i and B_j will be denoted by $B_i \cdot B_j$. Therefore the notation $B_i \cdot B_j \cdot \overline{B}_f$ stands for: $b(V_i) = 1 \wedge b(V_j) = 1 \wedge b(V_f) = 0$. The initial uncertainty is c for the assumed presence of a sensor in each source location:

$$p_0(j) = \Pr\left[\Delta_0 = j\overline{\Delta}\right] = \delta(j - c),$$

where $\delta(j)$ denotes the Kronecker delta function: $\delta(j) = 1$ if $j = 0$, $\delta(j) = 0$ otherwise.

Since each node at level k of the DT represent a possible path of length k, the probability $p_k(j)$ of having uncertainty $j\overline{\Delta}$ after k steps can be computed by looking at the contribution of all the DT nodes $\mathcal{T}_k$. Let $H_k = \text{card}(\mathcal{T}_k)$ be the cardinality of $\mathcal{T}_k$, and $\Phi_j(n_{k,h})$ be the function computing the contribution of the $n_{k,h}$ to the probability $p_k(j)$, i.e.

$$p_k(j) = \sum_{h=1}^{H_k} \phi(n_{k,h}) \Phi_j(n_{k,h}), \tag{2}$$

therefore, the contribution $\Phi_j(n_{k,h})$ is weighted by the probability that the path is actually taken.

We now focus on the computation of $\Phi_j(n_{k,h})$ as a function of the variables $b(.)$ for the nodes in the sequence. We start with the case of *linear paths*, i.e., paths where each node occurs once. The possible values of the uncertainty Δ_k accumulated after k steps depend on the last node in the sequence in which a sensor is placed. If the last element of $\sigma(n_{k,h})$ is a destination node then the uncertainty will be c for the assumed presence of a sensor. Thereby, $\Phi_j(n_{k,h})$ will be 0 for all values $j \neq c$ and 1 for $j = c$. In the general case, the sequence $\sigma(n_{k,h})$ has length $k+1$, but the first element of the sequence is always a source node and is also associated with a sensor. Therefore, only the last k elements are associated with decision variables. Define by $V^{(f)}_{n_{k,h}}$, with $f \in [1,k]$, the f^{th} element of this sequence and let $B^{(f)}_{n_{k,h}}$ its associated boolean variable. Define the boolean expression

$$\psi^{(f)}_{n_{k,h}} = \begin{cases} B^{(f)}_{n_{k,h}} & \text{if } f = k, \\ \overline{B^{(k)}_{n_{k,h}}} \cdot \overline{B^{(k-1)}_{n_{k,h}}} \cdots \overline{B^{(f+1)}_{n_{k,h}}} \cdot B^{(f)}_{n_{k,h}} & \text{if } f \in [1,k[, \\ \overline{B^{(k)}_{n_{k,h}}} \cdot \overline{B^{(k-1)}_{n_{k,h}}} \cdots \overline{B^{(1)}_{n_{k,h}}} \cdot \overline{B^{(0)}_{n_{k,h}}} & \text{if } f = 0. \end{cases}$$

Whenever $\psi^{(f)}_{n_{k,h}}$ evaluates to true the distribution of the uncertainty will be given by the PMF $\eta^{(f)}_{n_{k,h}}(d)$. If $f = k$, this function is simply a Kronecker δ: $\eta^{(f)}_{n_{k,h}}(j) = \delta(j-c)$. If f is in the range $[0,k[$ the expression for $\eta^{(f)}_{n_{k,h}}(j)$ is given by a convolution:

$$\eta^{(f)}_{n_{k,h}}(j) = \delta(j-c) * \Gamma_{V^{(k-1)}_{n_{k,h}}, V^{(k)}_{n_{k,h}}}(j) * \ldots * \Gamma_{V^{(f)}_{n_{k,h}}, V^{(f+1)}_{n_{k,h}}}(j),$$

where $*$ denotes the convolution, since the overall uncertainty is given by a sum of independent random variables.

Example 2: Let us consider node $n_{3,3}$ in Example 1 in Figure 1 and Figure 2, which is associated to the path $\sigma(n_{3,3}) = \langle S, V_1, V_3, V_5 \rangle$. The possible expressions for the uncertainty with this path for the different truth assignments of the boolean variables are the following:

c	if $\psi^{(3)}_{n_{3,3}} = B_5$
$(c + \rho(V_3, V_5))$	if $\psi^{(2)}_{n_{3,3}} = \overline{B_5} \cdot B_3$
$(c + \rho(V_3, V_5) + \rho(V_1, V_3))$	if $\psi^{(1)}_{n_{3,3}} = \overline{B_5} \cdot \overline{B_3} \cdot B_1$
$(c + \rho(V_3, V_5) + \rho(V_1, V_3) + \rho(S, V_1))$	if $\psi^{(0)}_{n_{3,3}} = \overline{B_5} \cdot \overline{B_3} \cdot \overline{B_1}$

Hence, if $\psi^{(1)}_{n_{3,3}}$ is true, the PMF will be $\eta^{(1)}_{n_{3,3}}(j) = \delta(j-c) * \Gamma_{V^{(2)}_{n_{k,h}}, V^{(3)}_{n_{k,h}}}(j) * \Gamma_{V^{(1)}_{n_{k,h}}, V^{(2)}_{n_{k,h}}}(j)$, which is $\eta^{(1)}_{n_{3,3}}(j) = \delta(j-c) * \Gamma_{V_3,V_5}(j) * \Gamma_{V_1,V_3}(j)$.

From the previous analysis it follows immediately that

$$\Phi_j(n_{k,h}) = \sum_{f=0}^{k} \psi^{(f)}_{n_{k,h}} \eta^{(f)}_{n_{k,h}}(j),$$

that substituted in (2) gives the PMF

$$p_k(j) = \sum_{h=1}^{H_k} \phi(n_{k,h}) \sum_{f=0}^{k} \psi^{(f)}_{n_{k,h}} \eta^{(f)}_{n_{k,h}}(j). \tag{3}$$

This is a pseudo-boolean expression in which the boolean literal $\psi^{(f)}_{n_{k,h}}$ multiplies the expression of the value of the probability in case it is true. It has to be noted that this model accounts also for the presence of natural landmarks, e.g., mapped entities of the environment, that can be used as already available sensors with associated uncertainty.

The definition of $\psi^{(\cdot)}_{n_{k,h}}$ and $\eta^{(\cdot)}_{n_{k,h}}$ has to be slightly modified to account for cyclic paths. The presence of cyclic paths has an impact on the computation of $\psi^{(f)}_{n_{h,k}}$ and of $\eta^{(f)}_{n_{h,k}}$ if $f \in [1,k[$. Indeed, if the f^{th} node of $\sigma(n_{k,h})$ is present in the sequence in a position from $f+1$ to k, the value of its associated variable is bound to 0 from its previous occurrence in the chain. In this case the term $\psi^{(f)}_{n_{h,k}}$ is outright skipped in the computation of $\Phi_j(n_{k,h})$.

B. Setting up the optimisation problem

We can now use Expression (3) to reformulate the optimisation Problem (1) as

$$\min \sum_{B_i} B_i$$
$$s.t. \sum_{j=0}^{H} \sum_{h=1}^{H_k} \phi(n_{k,h}) \sum_{f=0}^{k} \psi^{(f)}_{n_{k,h}} \eta^{(f)}_{n_{k,h}}(j) \geq \mu, \forall k \tag{4}$$

The left hand side of each constraint is a sum of terms, and each term is a real constant that multiplies the logical product a the boolean variables $\psi^{(f)}_{n_{k,h}}$.

We transform this into a different problem where the logical products are "linearised". The first step is to consider B_i as a numeric value that can be 0 or 1. The second step to do this is replacing each term $\psi^{(f)}_{n_{k,h}}$ with an auxiliary variable w_i, defined over the real numbers. If two different terms have the same expression, then the same auxiliary variable is used to replace them. The third step is to insert additional constraints to enforce the fact that the auxiliary variable can be 1 or 0 depending on whether the logical expression is true or not. Consider a variable w_i and the term it is associated with. Let $\mathcal{A}_i$ be the index set of asserted variables in the term and $\mathcal{N}_i$ be the index set of negated variables. The constraints to be inserted are:

$$w_i \geq \sum_{j \in \mathcal{A}_i} B_j - \sum_{j \in \mathcal{N}_i} B_j + 1 - \text{card}(\mathcal{A}_i)$$
$$w_i \leq B_j, \qquad \forall j \in \mathcal{A}_i$$
$$w_i \leq 1 - B_j, \qquad \forall j \in \mathcal{N}_i$$

Example 3: Consider the term $\overline{B_5} \cdot \overline{B_3} \cdot B_1$ of Example 2 and suppose it is associated with an auxiliary variable w_3. Then the constraints will be:

$$w_3 \geq B_1 - B_3 - B_5$$
$$w_3 \leq B_1$$
$$w_3 \leq 1 - B_3$$
$$w_3 \leq 1 - B_5$$

We can easily see that $w_3 = 1$ if and only if $B_1 = 1$, $B_2 = 0$ and $B_3 = 0$. By applying this technique, we end up with a binary linear problem that can be solved by using any specialised tool (e.g., GLPK, CPLEX).

C. An upper bound for the verification horizon

If the topology of the graph describing the discrete abstraction has the structure of a Directed Acyclic Graph (DAG). Hence, there exists a maximum number K of steps, given by the diameter of the graph, beyond which all trajectories started from a source node will have reached a destination node. Given the presence of a sensor in the destination node, we have: $p_k(j) = \delta(j-c)$ for all $k \geq K$. Therefore there is no need to enforce the constraint for values of k greater than K. If there are cycles in the graph, this is not evidently true any more. Indeed, some of the trajectories can cycle through the same nodes again and again potentially forever (although with a small probability).

We assume that the Markov Chain is such that every node has at least one path (of arbitrary length) to a destination node. The destination nodes are associated with a self transition taken with probability 1. In the jargon of the Markov Chain this is called an "absorbing state" and a DTMC for which every node has a path to called an Absorbing Markov Chain.

It is convenient to adopt a numbering for the states so that the absorbing states are the last ones in the π_k vector. By doing so the transition matrix M will have the following form:

$$M = \left[\begin{array}{c|c} Q & R \\ \hline 0 & I_G \end{array}\right],$$

where Q is a $(L+N)\times(L+N)$ matrix and is associated to all states that are not destinations, I_G is the identity matrix of order G and accounts for the evolution of the system once it has reached a destination node (i.e., an absorbing state). The evolution of the system after n steps is given by:

$$\pi_n = \pi_0 M^n = \pi_0 \begin{bmatrix} Q^n & (Q^{n-1} + Q^{n-2} + \ldots + Q + I)R \\ 0 & I_G \end{bmatrix} \tag{5}$$

Using this expression, we are able to state the following result.

Theorem 1: Consider a discrete stochastic abstraction as in Definition 1 and assume that from every intermediate node source and intermediate node there is a path to at least a destination node. Then there exist a maximum value of K such that if $\Pr\left[\Delta_K \leq H\overline{\Delta}\right] \geq \mu$, then $\Pr\left[\Delta_k \leq H\overline{\Delta}\right] \geq \mu$ for all $k \geq K$.

Proof: We can use a standard line of reasoning for Absorbing Markov Chains [25] and write (5) as

$$\pi_n = \pi_0 M^n = \pi_0 \begin{bmatrix} Q^n & (I-Q)^{-1}(I-Q^n)R \\ 0 & I_G \end{bmatrix} \tag{6}$$

By partitioning the vector π_n into the sub-vectors $\pi_k^{(1)}$, related to the source and intermediate nodes, and $\pi_k^{(2)}$, related to the destination nodes, i.e. $\pi_k = \left[\pi_k^{(1)} \pi_k^{(2)}\right]$, the equation (6) can be written as

$$\pi_n^{(1)} = \pi_0^{(1)} Q^n, \tag{7}$$

$$\pi_n^{(2)} = \pi_0^{(2)} + \pi_0^{(1)}(I-Q)^{-1}(I-Q^n)R. \tag{8}$$

It is easy to see that $\lim_{n\to\infty} Q^n = 0$ [25]. Therefore for $n \to \infty$ the system will converge to a state where the probability of being in source or intermediate state will vanish $\pi_n^{(1)} \to 0$, while the probability of finding the target in destination states converges to the vector

$$\pi_n^{(2)} \to \pi_0^{(2)} + \pi_0^{(1)}(I-Q)^{-1}R = \pi_0^{(1)}(I-Q)^{-1}R.$$

This convergence is monotone. Indeed, if we consider the evolution of the state across one step, we have: $\pi_{k+1}^{(2)} = \pi_k^{(2)} + \pi_k^{(1)}R \geq \pi_k^{(2)}$ where the $\geq$ is applied element wise.

Since all the trajectories converge monotonically in probability to a node with a sensor (a destination node), the distribution of the $p_k(j)$ will approach the Kronecker $\delta(j-c)$ and, hence, there will be a value K such that the $\Pr\left[\Delta_K \leq H\overline{\Delta}\right]$ will be greater than μ from that point on. ■

The result above shows the soundness of the theoretical foundation of the paper. In essence, it means that the problem can be correctly set up and solved in finite time. Besides, given a spectral decomposition of the matrix Q, we can use the equation (8) to find a close estimate of the value of K for which the convergence takes place. We omit this discussion for the sake of brevity.

IV. Use Case

Many realistic scenarios have been considered in the testing phase to prove the solution effectiveness and here, for space limits, we report only a use case of the 200 m^2 exposition area "Salone Donatello" of the Bargello National Museum in Florence, Italy. As reported in Fig. 3, our model considers the room entrances and exits as source and destination nodes (hence, $L = G = 2$) and, additionally, the set of POIs like paintings and sculptures (marked with pointers). The room is split into regular squares on a grid, representing intermediate nodes. Each square covers approximately 3 m^2 and there are $N = 50$ available squares on which the sensor can be placed. The range and the shape of the area of detection has been set in order to satisfy these constraints:

- For every position of the robot in our environment at least one node is in the detection range of the vision system;
- The Robot is assumed to move freely in the space as explained in Section II;
- If the robot moves on a diagonal trajectory from a node to another one of the grid, the sensing system firstly detects a sensor on the same row or column.

The deployed sensing uncertainties and the ego-motion uncertainty are supposed to be modelled as in [6].

The Markov chain describing the motion of the vehicle inside the environment has been generated taking into account the attractiveness of the destination nodes as well as the presence of the POI. More in depth, each POI has been modelled as an attractive potential for the generation of the node sequences by modifying the probabilities of the transition matrix M, as exemplified in Fig. 4. This definition of the attractive potential fields has a close resemblance to standard robot motion planning algorithms. Even if simulated

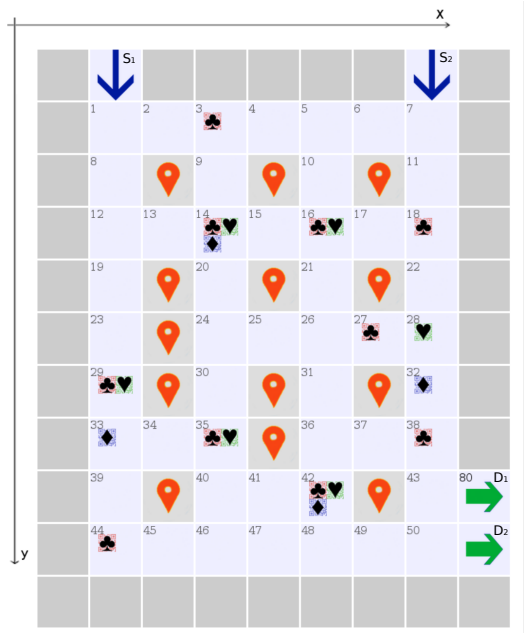

Fig. 3. Map of the "Salone Donatello" of the Bargello National Museum use case with optimal sensors placement for increasing target uncertainty of $H = 5$ (clubs), $H = 10$ (hearts) and $H = 15$ (diamonds). In all cases $\mu = 95\%$. Pointers represent the mapped POIs, considered as obstacles.

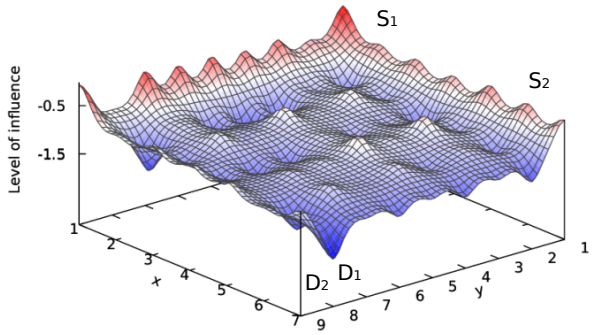

Fig. 4. Points of interest influence on the Markov chain transition matrix M of the use case of Fig. 3.

data are used for this example, mainly due to the impossibility of accessing real data from the museum, the model here synthesised is in perfect accordance with widely used stochastic motion models derived from direct observations, such as the Social Force Model [26] or the probability–based cellular automata motion models [27] or the agent–based probability models [9], [28], and similar Markov chain motion descriptions have already been synthesised on real data for robotic applications [29].

A. Optimisation

Once the Markov chain is available, the implemented tool visits the graph with a mixed approach of breadth first and depth first search. The level of the depth and the acceptable threshold of uncertainty $H\bar{\Delta}$ defined in (1) are input parameters. For each path analysed, the tool computes both the probability of the path $\phi(n_{k,h})$ and the maximum amount of uncertainty $\Phi_j(n_{k,h})$ as in (2).

The number of paths increases exponentially with the number of steps k. For this reason this tool implements an analysis of the generated auxiliary variables w_i in order to reduce the space complexity. The variable w_i represents the boolean expression $\psi^{(\cdot)}_{n_{k,h}}$ described in Section III. Each w_i is preprocessed in order to reduce redundancy and memory usage by combining all of them sharing common variables in the constraints. After this preprocessing, the data is fed to the binary linear problem solver, such as GLPK or more efficient optimisers like CPLEX or GUROBI.

H	Front-End	Paths	w_i	Optimiser	Sensors
5	88m 5s	511M	590	5s	10
10	92m 10s	511M	5598	60s	6
15	111m 28s	511M	46801	31m 35s	4

TABLE I

PERFORMANCE ANALYSIS OF THE "SALONE DONATELLO" OF THE BARGELLO NATIONAL MUSEUM USE CASE.

B. Performance and results

The optimisation algorithm determines $b(V_i)$, $i = 1, \ldots, N$, assuming $\mu = 95\%$ and different values of $H\bar{\Delta}$. All the computations has been performed on a desktop equipped with a consumer quad core CPU and 16 GByte of RAM, while the total amount of trajectories, evaluated by a front-end coded on purpose in C++, is slightly more than $511 \cdot 10^6$. The computation time of this front-end tool and the GLPK optimiser software for the described example are reported in Table I, while the optimal position of the sensors for each target uncertainty $H\bar{\Delta}$ are depicted with different shapes in Fig. 3.

The number of deployed sensors lowers down from 10 to 4, as expected, when H increases from 5 to 15 (i.e., a lower accuracy is requested). However, both the front-end and the optimiser computation times increase with lower requests in term of uncertainty (being the front-end the more computational intensive). This is due by the larger number of paths to be checked (see the number of literals w_i). It is worthwhile to point out that a time limit can be imposed to the software that in these cases provides a suboptimal but feasible solution. For instance, for $H = 15$, the optimiser reports a suboptimal solution with 5 sensors in less than 10 minutes, one third of the time reported for the optimal 4 sensors.

Finally, the optimal solutions has been fed to an evaluation tool that verifies the constraints and provides the cumulative distribution functions of the positioning uncertainty after optimal placement for each choice of $H\bar{\Delta}$. The solution is computed for a maximum value of $k = 22$ steps which means that over the 88% of the trajectories reached a destination node. This value can be considered as an index of the likelihood of the results.

V. Conclusions

In this paper we have focused on the problem of indoor localisation of mobile robots in large indoor environments. The type of robots we target perform the self localisation task using a combination of on board sensing devices with occasional readings of environmental markers, e.g., visual markers or RFIDs. The latter give absolute position information and allow the system to reset its uncertainty within acceptable boundaries. We have shown a method for the optimal placement of the markers and for the minimisation of their number. The idea is based on the introduction of a stochastic discrete abstraction that describes in probabilistic terms the trajectories of the robot in the environment observed in the past executions. The use of this abstraction allows us to set up a mixed boolean optimisation problem, which finds the minimal number of markers and their position subject to the constraint that the uncertainty remains below a target value with an assigned probability.

The concrete robotic application that motivated our effort is the self-localisation of a robotic walker, used to support senior users to navigate and use large public spaces. But, we believe that the outreach of our results goes far beyond and applies to all mobile robots that execute repetitive tasks in a large environments. The main future research directions are related to the extension to different application scenarios, to the improvement of the expressiveness of the method (e.g., by including a probability of sensor detection) and the development of more efficient solution strategies. Finally, an extensive collection of data in a real environment leading to an experimental validation of the work is currently under way.

References

[1] A. Colombo, D. Fontanelli, D. Macii, and L. Palopoli, "Flexible Indoor Localization and Tracking based on a Wearable Platform and Sensor Data Fusion," *IEEE Trans. on Instrumentation and Measurement*, vol. 63, no. 4, pp. 864–876, April 2014.

[2] H. Chung, L. Ojeda, and J. Borenstein, "Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 1, pp. 80–84, Feb. 2001.

[3] ACANTO Consortium, "Grant Agreement n° 643644 of the ACANTO project - Description of Action," Available in the ACANTO project website: www.ict-acanto.eu, Feb. 2015.

[4] A. Colombo, D. Fontanelli, A. Legay, L. Palopoli, and S. Sedwards, "Efficient customisable dynamic motion planning for assistive robots in complex human environments," *Journal of Ambient Intelligence and Smart Environments*, 2015, to appear.

[5] L. Palopoli, A. Argyros, J. Birchbauer, A. Colombo, D. Fontanelli, and et. al., "Navigation Assistance and Guidance of Older Adults across Complex Public Spaces: the DALi Approach," *Intelligent Service Robotics*, vol. 8, no. 2, pp. 77–92, 2015.

[6] P. Nazemzadeh, F. Moro, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor Positioning of a Robotic Walking Assistant for Large Public Environments," *IEEE Trans. on Instrumentation and Measurement*, vol. 64, no. 11, pp. 2965–2976, Nov 2015.

[7] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[8] T. Sasaki, D. Brscic, and H. Hashimoto, "Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1401–1410, April 2010.

[9] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 4015–4020.

[10] A. Cameron and H. Durrant-Whyte, "A bayesian approach to optimal sensor placement," *The International Journal of Robotics Research*, vol. 9, no. 5, pp. 70–88, 1990.

[11] H. Zhang, "Optimal sensor placement," in *IEEE International Conference on Robotics and Automation*, May 1992, pp. 1825–1830 vol.2.

[12] S. Martnez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661 – 668, 2006.

[13] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[14] P. Nazemzadeh, D. Fontanelli, D. Macii, and L. Palopoli, "Indoor Positioning of Wheeled Devices for Ambient Assisted Living: a Case Study," in *Proc. IEEE Int. Instrumentation and Measurement Technology Conference (I2MTC)*. Montevideo, Uruguay: IEEE, May 2014, pp. 1421–1426.

[15] M. Beinhofer, J. Müller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1060–1069, 2013.

[16] N. Du Toit and J. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, Feb 2012.

[17] D. Althoff, J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.

[18] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[19] B. Kuipers, "The spatial semantic hierarchy," *Artificial intelligence*, vol. 119, no. 1, pp. 191–233, 2000.

[20] A. Alempijevic, R. Fitch, and N. Kirchner, "Bootstrapping navigation and path planning using human positional traces," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 1242–1247.

[21] F. Lin and P. Chiu, "A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks," *IEEE Communications Letters*, vol. 9, no. 1, pp. 43–45, Jan 2005.

[22] D. B. Jourdan and N. Roy, "Optimal Sensor Placement for Agent Localization," *ACM Trans. Sen. Netw.*, vol. 4, no. 3, pp. 13:1–13:40, Jun. 2008.

[23] P. Nazemzadeh, D. Fontanelli, and D. Macii, "Optimal placement of landmarks for indoor localization using sensors with a limited range," in *Proc. of International Conference on Indoor Positioning and Indoor Navigation (IPIN) 2016*, Alcalá de Henares, Spain, October 2016.

[24] P. Barth and I. Stadtwald, "A davis-putnam based enumeration algorithm for linear pseudo-boolean optimization," 1995.

[25] *Finite Markov Chains*. Springer-Verlag, 1976, ch. 3: Absorbing Markov Chains, p. 224, iSBN 978-0-387-90192-3.

[26] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, 1995.

[27] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, "Simulation of pedestrian dynamics using a two-dimensional cellular automaton," *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 34, pp. 507 – 525, 2001.

[28] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," *Computer Vision (ECCV)*, pp. 201–214, 2012.

[29] R. Asaula, D. Fontanelli, and L. Palopoli, "Safety provisions for Human/Robot Interactions using Stochastic Discrete Abstractions," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, October 18-22 2010, pp. 2175–2180.