A Framework for Robust Cognitive Spatial Mapping

Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop and Patric Jensfelt

Abstract— **Spatial knowledge constitutes a fundamental component of the knowledge base of a cognitive, mobile agent. This paper introduces a rigorously defined framework for building a cognitive spatial map that permits high level reasoning about space along with robust navigation and localization. Our framework builds on the concepts of *places* and *scenes* expressed in terms of arbitrary, possibly complex features as well as local spatial relations. The resulting map is topological and discrete, robocentric and specific to the agent's perception. We analyze spatial mapping design mechanics in order to obtain rules for how to define the map components and attempt to prove that if certain design rules are obeyed then certain map properties are guaranteed to be realized. The idea of this paper is to take a step back from existing algorithms and literature and see how a rigorous formal treatment can lead the way towards a powerful spatial representation for localization and navigation. We illustrate the power of our analysis and motivate our cognitive mapping characteristics with some illustrative examples.**

I. Introduction

An autonomous mobile agent needs to represent its surroundings in order to reason an plan actions within it. The typical spatial knowledge representations used in mobile robotics are purely metrical and rely on information extracted from simple, but accurate metric sensors. However, as the robots are designed to perform human-like tasks in more and more complex and dynamic environments [3], [8], [14], metrical global maps become harder to control and observe [5]. Moreover, it is not clear that the level of detail offered by such maps is necessary, or even desirable, when the agent is a cognitive system intended to interact with the world in a human-like way [5], [14]. It is commonly accepted [5], [8], [9], [14], that the spatial knowledge of a cognitive agent should be abstracted in order to make it robust to dynamic variations, easier to maintain and useful for spatial reasoning. At the same time, the agent should be able to exploit sensory information that might be complex and non-metric [3], [8], [9], yet reflects crucial aspects of the environment.

This paper is motivated by desire to create a powerful cognitive mapping framework, which is suitable for cognitive conceptualization, encompasses complex spatial information, and provides robustness against natural changes in the environment, while maintaining a description that permits formal proofs and derivations. Although the literature contains many algorithms for spatial mapping, there is little work on the formal analysis of their fundamental requirements and properties. The idea of this paper, is to take a step back and see how a rigorous formal treatment can lead the way towards a powerful spatial representation for localization and navigation.

The authors are with the Centre for Autonomous Systems at the Royal Institute of Technology (KTH), Stockholm, Sweden. This work was supported by the Centre for Autonomous Systems (CAS) and the EU FP7 project CogX and the Swedish Research Council, contract 621-2006-4520 (K. Sjöö) and 2005-3600-Complex (A. Pronobis).

The contribution of the work presented here is a cognitive mapping framework that builds on the concepts of *places* and *scenes* expressed in terms of arbitrary, possibly complex features as well as local spatial relations. The resulting map is topological and discrete, robocentric and specific to the agent's perception. We analyze spatial mapping design mechanics in order to obtain rules for how to define the map components and attempt to prove that if certain design rules are obeyed then certain map properties are guaranteed to be realized. Moreover, we suggest localization and navigation strategies that can be applied in this framework. Finally, we illustrate the power of our analysis and motivate our cognitive mapping characteristics with illustrative examples.

The paper is organized as follows: after a general overview of the framework, Section III presents the formal definition of the map and its components. Then, Section IV gives a method for expressing the map through a set of functions and provides rules that must be obeyed in order for the map to be valid. Sections V and VI propose methods for performing navigation as well as probabilistic localization within the framework. The paper concludes with a summary and a brief discussion.

II. An Overview of the Framework

The role of a cognitive map is not to represent the world as accurately as possible, but rather to allow the agent to act in an environment despite uncertainty and dynamic variations. Such a map does not need to provide perfect global consistency as long as the local spatial relations are preserved with sufficient accuracy. In our framework, the map is represented as a collection of basic spatial entities called *places*.

A *place* is defined by a subset of values of arbitrary, possibly complex, distinctive features and spatial relations reflecting the structure of the environment. The features provide information about the world and can be perceived by an agent when at that place. In this sense, the places build on the perception of the agent and are based on its perceptual capabilities. Additionally, we introduce the concept of a *scene* which facilitates the generation of places by providing groupings of similar feature values. In addition to this, a scene provides a segmentation of space that serves as a basis for defining spatial relations.

The structure of the framework and its formalization described in the next section represent a certain view on

a cognitive map. First, the map is defined in terms of the agent's perception of space and adapts to its perceptual capabilities. Second, the perceived features can be abstract and non-metric and describe for instance visual properties of the world. In this sense, the map is subjective and robocentric as the robot's observations do not have to be expressed in terms of any objectively defined quantities or any global coordinate system. The map is fragmented (consists of a set of independent places), topological and does not require maintaining global spatial consistency.

This framework is designed so that a robot can build from the bottom-up a cognitive map of the environment which follows certain cognitive principles. The idea is that such principles can actually lead to better performance in localization, navigation and loop-closing for robots moving in large-scale environments; e.g. see the practical demonstrations in [5], [8]. The work of [5] involves a similarly designed mapping framework to the one analyzed in this paper and motivates the need to take a step back and analyze what desirable properties of the cognitive map can be provably obtained. The next section provides a formal definition of the place map and each of its components.

III. Definition of the Place Map

Consider a set $\{f_i\}_{i=1}^{n_f}$ of *features* f_i defined as

$$f_i\left(\boldsymbol{x}, t\right) : \mathcal{C} \times \mathbb{R} \rightarrow \mathcal{F}_i \in \mathbb{R}^n \tag{1}$$

where $\mathcal{C}$ represents the *configuration space* of the agent (e.g. $\mathcal{C} = \mathbb{R}^2$ if only position in a 2D metric space is considered and $\mathcal{C} = \mathbb{R}^2 \times SO(1)$ if the value of features can depend on both position and heading), $t \in \mathbb{R}$ represents time, and $\mathcal{F}_i$ is the range of values of the feature f_i. Features thus provide information about the world as it would be perceived by an agent located at the configuration $\boldsymbol{x} \in \mathcal{C}$. Each feature can be time-varying.

An example feature-type is Euclidean distance, $f_i(\boldsymbol{x}, t) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$, with $\mathcal{F}_i = [0, \infty)$, which maps every point in $\mathcal{C}$ to a value dependent on how far $\boldsymbol{x} \in \mathcal{C}$ is to a specific landmark located at $\boldsymbol{y} \in \mathcal{C}$. Features do not necessarily have to describe metric properties of the world (such as distance or size). Consider for instance a visibility-type feature for which $\mathcal{F}_i = \{0, 1\}$, which relates every pose $\boldsymbol{x} \in \mathcal{C}$ to a binary output depending on whether or not a specific landmark is visible in that pose. Another example would be a feature $f_i(\boldsymbol{x}, t)$ with $\mathcal{F}_i = [0, 1]$, which represents the average hue perceived by the robot's visual sensor, or even the full HSV color space, in which case $\mathcal{F}_i = [0, 1] \times [0, 1] \times [0, 1]$. Such features may be time-varying e.g. due to changes in illumination.

Other, more abstract, feature types are possible in this framework. An example could be features typically employed in visual topological localization [3], [5], [9] such as clouds of image keypoints characterized by the local SIFT [7] or SURF [2] descriptors. Features such as the "gist" of a scene [13] (principal components of outputs of spatially oriented image filters) or other global image features applied for visual place classification [9] could also be used in this framework in a straightforward manner. In such case, $f_i(\boldsymbol{x}, t)$ is a vector representing local descriptors for the N strongest keypoints or the global descriptor.

Given the definition of features, we can now introduce the *feature space*

$$\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \ldots \times \mathcal{F}_{n_f}, \tag{2}$$

in which each tuple $(\zeta_1, \ldots, \zeta_{n_f})$ of the feature values $\zeta_i = f_i(\boldsymbol{x}, t)$ corresponds to a single point. We are now ready to define the concept of a *scene*.

Definition 1: We introduce a set $\{\mathcal{S}_i\}_{i=1}^{n_s}$ of *scenes* $\mathcal{S}_i$ defined as

$$\mathcal{S}_i = \left\{(\zeta_1, \ldots, \zeta_{n_f})\right\} \subseteq \mathcal{F} \tag{3}$$

such that $\forall_i \mathcal{S}_i \neq \emptyset$ and $\forall_{i \neq j} \mathcal{S}_i \cap \mathcal{S}_j = \emptyset$. In other words, scenes are (non-overlapping) collections of tuples of features that could be perceived by the robot. Then, it is possible to specify the extent of a scene in the configuration space at time t:

$$\mathcal{C}_{\mathcal{S}_i}(t) = \left\{\boldsymbol{x} \in \mathcal{C} : \left(f_1(\boldsymbol{x}, t), \ldots, f_{n_f}(\boldsymbol{x}, t)\right) \in \mathcal{S}_i\right\} \tag{4}$$

It is important to note that no assumptions need to be made about the properties or structure of the feature functions in order to determine if a point $\boldsymbol{x} \in \mathcal{C}$ is within the spatial extent of a scene $\mathcal{S}_i$ at time t. In particular, a closed-form expression is not required as long as the feature values can be obtained. This has important practical implications as it permits the use of more complex features.

The definition of scenes gives raise to a segmentation of the configuration space. Depending on the features, however, this segmentation may not reflect the spatial relationships in the world that constitute a large portion of the spatial knowledge. Intuitively, the definition of scenes leads to a division of metric space into regions based on properties such as appearance. As such, two distant disconnected regions could share similar properties (see e.g. Figure 1(a)). Additional power to distinguish between such regions can be attained using knowledge about spatially neighboring regions.

Consider a set $\{r_i\}_{i=1}^{n_r}$ of *spatial relations* r_i defined as

$$r_i\left(\boldsymbol{x}, t\right) : \mathcal{C} \times \mathbb{R} \rightarrow \mathcal{R}_i \in \mathbb{R}^m, \tag{5}$$

where $\mathcal{R}_i$ is the range of values of the relation r_i. Each *spatial relation* r_i is defined with respect to the set of scenes $\{\mathcal{S}_i\}_i$ and describes the spatial relation of the point $\boldsymbol{x}$ in the configuration space $\mathcal{C}$ at time t to some or all of those scenes. Relations permit discriminating between points using region-based concepts such as the region connection calculus, RCC [10], often applied in qualitative spatial reasoning. Moreover, in many cases, the values of relations can be estimated in practice by performing a dynamic action in the environment (e.g. the agent moving between points in configuration space that correspond to different scenes).

Consider, for instance, the adjacency relation for which $\mathcal{R}_i = \{1, 0\}$. The adjacency relation $r_{\mathcal{S}_i}(\boldsymbol{x}, t)$ of a point $\boldsymbol{x} \in \mathcal{C}$ to the region $\mathcal{C}_{\mathcal{S}_i}$ can be expressed in terms of the RCC-8 [10] predicate EC (externally connected) as

$$r_{\mathcal{S}_i}(\boldsymbol{x}, t) = \bigvee_{\mathcal{S}_j} \boldsymbol{x} \in \mathcal{C}_{\mathcal{S}_j} \wedge \mathsf{EC}(\mathcal{C}_{\mathcal{S}_i}, \mathcal{C}_{\mathcal{S}_j}). \tag{6}$$

Alternatively, a relation could be defined based on the minimum distance between a region $\mathcal{C}_{\mathcal{S}_i}$ and a point $\boldsymbol{x}$ in the configuration space as follows

$$r_{\mathcal{S}_i}(\boldsymbol{x}, t) = \min_{\boldsymbol{y} \in \mathcal{C}_{\mathcal{S}_i}} \|\boldsymbol{x} - \boldsymbol{y}\|. \tag{7}$$

We have now defined scenes and spatial relations, the main building blocks of the spatial entities constituting our map. Analogously to the feature space, we can introduce the *place descriptor space*

$$\mathcal{D} = \mathcal{F} \times \mathcal{R}_1 \times \mathcal{R}_2 \times \ldots \times \mathcal{R}_{n_r}, \tag{8}$$

in which each tuple $D = (\zeta_1, \ldots, \zeta_{n_f}, \rho_1, \ldots, \rho_{n_r})$ of the feature values and relation values $\rho_i = r_i(\boldsymbol{x}, t)$ corresponds to a single point.

Definition 2: Let us define the *place map* as a set

$$\mathcal{M} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{n_p}\} \tag{9}$$

of *places* $\mathcal{P}_i$ defined as

$$\mathcal{P}_i = \{D\} \subseteq \mathcal{D} \tag{10}$$

such that $\forall_i \mathcal{P}_i \neq \emptyset$ and $\forall_{i \neq j} \mathcal{P}_i \cap \mathcal{P}_j = \emptyset$.
In other words, similarly to scenes, places are groups of values of features; however, they encompass additional knowledge about the structure of the world encoded in the values of relations.

As a result, it is possible to specify the extent of a place in the configuration space at time t, as follows

$$\begin{aligned} \mathcal{C}_{\mathcal{P}_i}(t) \quad = \quad & \{\boldsymbol{x} \in \mathcal{C} : (f_1(\boldsymbol{x}, t), \ldots, f_{n_f}(\boldsymbol{x}, t), \\ & r_1(\boldsymbol{x}, t), \ldots, r_{n_f}(\boldsymbol{x}, t)) \in \mathcal{P}_i\} \end{aligned} \tag{11}$$

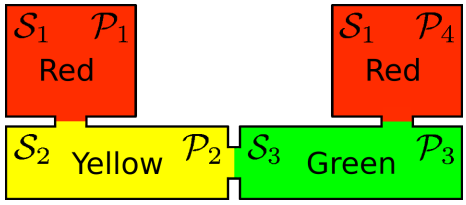
Note that not every point $\boldsymbol{x} \in \mathcal{C}$ is necessarily assigned to a place $\mathcal{P}_i$. The set of points $\mathcal{Q}(t) = \mathcal{C} \,/\, \bigcup\{\mathcal{C}_{\mathcal{P}_i}(t)\}_{i=1}^{n_p}$ is denoted *unassigned space* at time t. Again, no assumptions have to be made about the structure of the functions used to obtain the values of features and relations in order to determine if a point $\boldsymbol{x} \in \mathcal{C}$ is within the spatial extent of a place at time t.

Let us discuss the properties of places in the configuration space. Places are defined exclusively in terms of the values of features and spatial relations that are in functional relation to $(\boldsymbol{x} \in \mathcal{C}, t \in \mathbb{R})$. Moreover, places do not overlap in the descriptor space. As a consequence, places do not overlap in configuration space: $\forall_{i \neq j, t \in \mathbb{R}} \mathcal{C}_{\mathcal{P}_i}(t) \cap \mathcal{C}_{\mathcal{P}_j}(t) = \emptyset$.

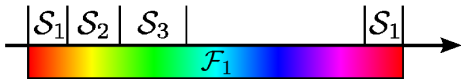
Also, if features and relations are time-invariant, the extents of places will be time-invariant as well. Typically, the nature of relations will mean that they are time-invariant as long as the features are. Note that if the configuration space reflects both position and heading, the places might spread across several positions and only a subset of headings.

A. Example 1 - Abstract Features and Relations

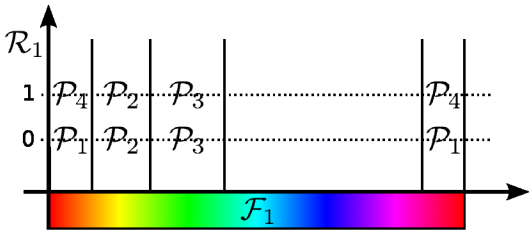
Consider a simple example of a small environment presented in Figure 1(a) consisting of 4 rooms characterized by the color of the floor. We define a single feature $f_1(\boldsymbol{x}, t) : \mathcal{C} \times \mathbb{R} \rightarrow \mathcal{F}_1$ that corresponds to the hue of the floor color at the location $\boldsymbol{x} \in \mathcal{C} = \mathbb{R}^2$. Then, the feature space is simply



(a) Map of the environment and metric extents of places.



(b) Scenes defined in the feature space.



(c) Places defined in the descriptor space.

Fig. 1. Illustrative example of an environment and definitions of places in the descriptor space.

defined by the range of the hue values e.g. $\mathcal{F}_1 = [0, 255]$. If we divide the feature space into regions as presented in Figure 1(b), we can differentiate between three scenes: red ($\mathcal{S}_1$), yellow ($\mathcal{S}_2$) and green ($\mathcal{S}_3$). We can clearly see that the scene $\mathcal{S}_1$ corresponds to two separate rooms which could be distinguished if we consider their relations to other scenes. Let us define an adjacency relation with respect to the scene $\mathcal{S}_3$, $r_1(\boldsymbol{x}, t) : \mathcal{C} \times \mathbb{R} \rightarrow \mathcal{R}_1 = \{1, 0\}$ as explained in Section III, and create the place descriptor space $\mathcal{D} = \mathcal{F}_1 \times \mathcal{R}_1$. In that space, we can create four non-overlapping places $\mathcal{P}_1$-$\mathcal{P}_4$ by dividing the scene $\mathcal{S}_1$ into two places, one of which is adjacent to the scene $\mathcal{S}_3$ and the other is not. This division is reflected in the clustering of the descriptor space presented in Figure 1(c).

IV. Space Segmentation Using Applicability

The division of the feature space and descriptor space that gives rise to scenes and then to places can be expressed in many different ways. This section describes the segmentation in terms of real-valued functions over space, which encode the degree of belonging to the different places or scenes.

This view imposes certain restrictions on the functions and thereby on the features and relations used, but given that these are satisfied it is shown that a consistent segmentation results. As will be demonstrated in Sections V and VI, this information can also be used to support both navigation and localization. We describe these functions both for scenes and places, denoting the feature/descriptor space (as the case may be) by $\mathcal{A}$, and an arbitrary point in that space by A. The reasoning is analogous for both cases.

We introduce a set $\{g_i\}_{i=1}^{n_g}$ of *applicability functions* g_i defined as

$$g_i(A) : \mathcal{A} \rightarrow \mathcal{G}_i \subseteq (\mathbb{R}^+ \cup \{0\}), \tag{12}$$

Definition 3: Given the set of applicability functions $\{g_i\}_{i=1}^{n_g}$, we define a cluster $\mathcal{K}_i \subseteq \mathcal{A}$ as

$$\mathcal{K}_i = \{A \in \mathcal{A} : g_i(A) > g_j(A) > 0, \forall i \neq j\} \quad (13)$$

and note especially that $g_i(A) = 0 \Rightarrow A \notin \mathcal{K}_i$.
Definition 3 suggests that we can think of the functions $g_i(A)$ as *measures* of how applicable a point A is to the cluster $\mathcal{K}_i$. The clusters are non-overlapping in $\mathcal{A}$: $\forall_{i \neq j} \mathcal{K}_i \cap \mathcal{K}_j = \emptyset$. We now examine the requirements this places on the spatially defined feature and relation functions.

To do this let us introduce an additional function

$$\chi_i(\boldsymbol{x}) = g_i(A) = g_i(a_1(\boldsymbol{x}, t), \ldots, a_{n_a}(\boldsymbol{x}, t)) \quad (14)$$

which represents the applicability over the configuration space. (Here, the a_i may be features only or features and relations, depending on whether $\mathcal{A} = \mathcal{F}$ or $\mathcal{A} = \mathcal{D}$.) As a result, it is similarly possible to specify the extent of a place in the configuration space at time t, as follows

$$\begin{aligned} \mathcal{C}_{\mathcal{P}_i}(t) &= \{\boldsymbol{x} \in \mathcal{C} : \chi_i(\boldsymbol{x}) > \chi_j(\boldsymbol{x}) > 0,\ \forall i \neq j\} \\ Q(t) &= \{\boldsymbol{x} : \chi_i(\boldsymbol{x}) = 0, \forall i\} \end{aligned} \quad (15)$$

However, this leaves parts of $\mathcal{C}$ undefined wherever no χ_i is greater than any other. If this occurs anywhere but on an infinitesimal borderline between places/scenes, it represents an ambiguity. To avoid this we introduce the following:

Definition 4: Let $\mu(\mathcal{S}) \geq 0$ denote the Lebesgue measure of the set $\mathcal{S}$ and Δ the set of all points not defined by Eq. 15. If $\mu(\Delta) = 0$, the spatial segmentation by $\{\chi_i\}$ is said to be *consistent*.

Proposition 1: Suppose that χ_i is a *piecewise analytical* function, i.e. that $\chi_i = \{\chi_{i,\alpha},\ \text{if } \mathbf{x} \in \mathbb{D}_{i,\alpha}\}, \forall \alpha$ where α is a countable index and where each $\chi_{i,\alpha}$ is a real analytic function on its open domain $\mathbb{D}_{i,\alpha}$ for all t. Assume that $\mu(\mathbb{D}_{i,\alpha}) > 0$ and $\{\bigcup_\alpha \text{cl}(\mathbb{D}_{i,\alpha})\} = \mathcal{C}$ where $\text{cl}(\mathbb{D}_{i,\alpha})$ is the closure of $\mathbb{D}_{i,\alpha}$ in $\mathcal{C}$. In the same way, let $\chi_j = \{\chi_{j,\beta},\ \text{if } \mathbf{x} \in \mathbb{D}_{j,\beta}\}, \forall \beta$ in the same way. Now assume that χ_i and χ_j are not identical on any entire intersection of their analytical pieces (except where both are identically zero):

$$\begin{aligned} &\forall \alpha \forall \beta : \mathbb{D}_{i,\alpha} \cap \mathbb{D}_{j,\beta} \neq \emptyset \Rightarrow \\ \Rightarrow\ &\chi_i(\mathbf{x}) \not\equiv \chi_j(\mathbf{x}) \vee \chi_i(\mathbf{x}) \equiv \chi_j(\mathbf{x}) \equiv 0 \text{ on } \mathbb{D}_{i,\alpha} \cap \mathbb{D}_{j,\beta} \end{aligned}$$

If the above holds for all pairs $i \neq j$, the segmentation of space into place via Eq. 15 is consistent, as per Definition 4.

Proof: The function $\chi_i - \chi_j$, is real and analytic on each non-empty $\mathbb{D}_{ij,\alpha,\beta} \triangleq \mathbb{D}_{i,\alpha} \cap \mathbb{D}_{j,\beta}$ Because of this, on $\mathbb{D}_{ij,\alpha,\beta}$ the zeros of $\chi_i - \chi_j$ are isolated unless χ_i and χ_j are equivalent functions, which is disallowed by the assumption, except where both functions are identically zero. Thus, the Lebesgue measure of the zero set of $\chi_i - \chi_j$ is zero (the borders of the $\mathbb{D}_{i,\alpha,\beta}$ also have measure 0). The proposition follows immediately. ■

A simple, but useful, corollary of this proposition is as follows.

Corollary 1: The segmentation of space into places via Eq. 15 is consistent, as per Definition 4, if χ_i and χ_j are real analytic functions on the domain $\mathcal{C}$, and $\chi_i \not\equiv \chi_j$ on $\mathcal{C}$.

If a_i are piece-wise analytic functions and each applicability function g_i is analytic on $\mathcal{A}$, then χ_i is piece-wise analytic on a partitioning of $\mathcal{C}$ (where the partitioning is a function of the domains on which a_i are analytic).

The requirement that χ_i, $\forall i$ are real analytic functions on all of $\mathcal{C}$ is sufficient but not necessary. In some cases this requirement is too restrictive; e.g. it prohibits binary (true/false) type features. The following result provides an useful augmentation.

Proposition 2: Suppose that $\chi_i = (\chi_i^d + \chi_i^a)\chi_i^b$ and $\chi_j = (\chi_j^d + \chi_j^a)\chi_j^b$, where χ_i^a and χ_j^a are real analytic functions on $\mathcal{C}$, and χ_i^d and χ_j^d are piecewise constant on $\mathcal{C}$. Moreover, χ_i^b and χ_j^b are functions taking values in $\{0, 1\}$ over all $\mathcal{C}$. Assume that $\chi_i^a - \chi_j^a \not\equiv C$ where C is a constant. Then the segmentation of space into places is consistent, as per Definition 4.

Proof: Note first that with no loss of generality we can ignore the effect of χ_i^b and χ_j^b and consider only the remaining functions. χ_i^d is a constant function $\chi_i^d \equiv C_\alpha$ on each open domain $\mathbb{D}_{i,\alpha}$, where $\{\bigcup_\alpha \text{cl}(\mathbb{D}_{i,\alpha})\} = \mathcal{C}$, and analogously for χ_j^d. Then, $\chi_i - \chi_j$ is a piecewise real analytic function on each non-empty $\mathbb{D}_{ij,\alpha,\beta} \triangleq \mathbb{D}_{i,\alpha} \cap \mathbb{D}_{j,\beta}$, and $\chi_i - \chi_j \equiv \chi_i^a - \chi_j^a + C_\alpha - C_\beta$ on $\mathbb{D}_{ij,\alpha,\beta}$. The zero set of this function can only have a non-zero Lebesgue measure if $\chi_i^a - \chi_j^a$ is constant, which is disallowed. ■

The last proposition accounts for discrete-valued feature types to be used in admissibility functions as a special case (given that they are accompanied by a continuous component).

Features of the type $f_i(\mathbf{x}, t) : \mathcal{C} \rightarrow \{0, 1\}$ are useful since so-called visibility features are of this type. That is, a point $\mathbf{y}^* \in \mathcal{C}$ is either visible (1) or not visible (0) from another point $\mathbf{x} \in \mathcal{C}$. The support of a visibility feature $f_i(\mathbf{x}, t) : \mathcal{C} \rightarrow \{0, 1\}$ belongs to the class of so-called star-shaped sets; e.g. see [4].

In the final corollary, we show how two useful classes of feature functions can be combined in an applicability function to provide a consistent segmentation of space:

Corollary 2: Assume that

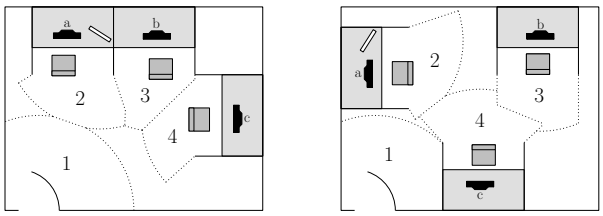
$$\chi_i = \Omega_i\left(\{a_k^b\}_{k \in M^b}\right)\left(\sum_{k \in M^d} \lambda_{ik} a_k^d + \chi_i^a\right) \quad (16)$$

where a_k^b are binary-valued features from A, and a_k^d are piece-wise constant functions taken from A. Ω is any logical expression on the a_k^b. Assume that $\chi_i^a - \chi_j^a \not\equiv C$ where C is a constant. Then the segmentation of space into places is consistent, as per Definition 4.

A. Example 2 - Distance and Visibility Features

As a theoretical illustration, consider a small office with three desks (see Figure 2(a)). The desks each have a computer screen and one additionally a framed picture. They are partially surrounded by partitions which block the view.

Four places have been assigned, all defined by different features (t omitted for clarity):



(a) Configuration 1 (b) Configuration 2

Fig. 2. Two configurations of an office and their consequent place regions.

- $\mathcal{P}_1$ - "Close to door object"
  $\chi_1(\boldsymbol{x}) = f_{door_c}(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{1}+\|\boldsymbol{p}_{\mathbf{door}}-\boldsymbol{x}\|}$
- $\mathcal{P}_2$ - "Close to picture and picture visible"
  $\chi_2(\boldsymbol{x}) = f_{pic_v}(\boldsymbol{x}) \cdot f_{pic_c}(\boldsymbol{x}) = f_{pic_v}(\boldsymbol{x}) \cdot \frac{1}{1+\|\boldsymbol{p}_{pic}-\boldsymbol{x}\|}$
- $\mathcal{P}_3$ - "Close to computer b and in front of desk"
  $\chi_3(\boldsymbol{x}) = f_{desk_f}(\boldsymbol{x}) \cdot f_{comp2_c}(\boldsymbol{x})$
  $= f_{desk_f}(\boldsymbol{x}) \cdot \frac{1}{1+\|\boldsymbol{p}_{comp2}-\boldsymbol{x}\|}$
- $\mathcal{P}_4$ - "Close to computer c and computer c visible"
  $\chi_4(\boldsymbol{x}) = f_{comp3_v}(\boldsymbol{x}) \cdot f_{comp3_c}(\boldsymbol{x})$
  $= f_{comp3_v}(\boldsymbol{x}) \cdot \frac{1}{1+\|\boldsymbol{p}_{comp3}-\boldsymbol{x}\|}$

Here,

$$f_{pic_v}(\boldsymbol{x}) = \begin{cases} 1 & \text{if picture unoccluded from } \boldsymbol{x} \\ 0 & \text{otherwise} \end{cases}$$

and analogously for f_{comp2_v} and f_{comp3_v}. The "in front of" feature is also binary:

$$f_{desk_f}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \mathcal{X}_{desk} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{X}_{desk}$ is a region projecting straight outward from the edge of the desk – cf. Figure 2(b).

These applicability functions fulfill the requirements of Proposition 2, as the radial components have different centers. It is assumed that there is a threshold for the applicability functions, below which a point is not considered part of any of the places (hence the circular borders). In effect, the regions belonging to the four places "compete" for the space and the best match wins out at each point.

These features exemplify the different sorts of functional aspects that define places to a cognitive agent. In a real-world scenario, places would likely be characterized by a larger number of features combined, for increased robustness. For the same reason, the granularity of places would typically also be finer. Also, since the features would be selected autonomously by a robotic agent their definition might be less human-comprehensible than the above selection. Still, this discrepancy will ideally be kept small, so that the spatial conceptualization of human and robot are invariant to similar types of features.

In Figure 2(b), the same office is shown after a rearrangement of the desks. Note how the regions, though their shape and size have changed, remain well-defined and how the cognitively conceptualized places (in the sense of having functionally conceived features) maintain their semantic significance despite having entirely different metric properties.

V. Navigation

The places discussed in Section III provide the segmentation of space into discrete units, and allow an agent to localize itself in the environment, by evaluating places' descriptor sets at its current location using its sensors. A map must, besides allowing for localization, provide a means for navigating through it. We do this in terms of *paths*, which represent the (potential) movement from one (start) place to another (goal) place. Just as places are defined by descriptors, so each path is associated with a *path precept*.

Definition 5: Let $\mathcal{S}$ represent the space of low-level sensor inputs available to the agent. Similarly, let $\mathcal{O}$ represent the space of low-level control outputs. Then, a path precept is a mapping from a low-level sensory state $s \in \mathcal{S}$ to a control output $o \in \mathcal{O}$:

$$\pi_i : \mathcal{S} \mapsto \mathcal{O} \tag{17}$$

A path is always associated with exactly one precept. $\mathcal{S}$ is of course given by the system instantiation, and may include virtual sensor modalities, such as local metric maps built over a period of time. It is in general a richer representation than the feature space $\mathcal{F}$, and allows for low-level considerations such as obstacle avoidance and other reactive behaviours.

The above definition is very general and admits path precepts that produce any sort of output. We therefore distinguish between *proper* and *improper* path precepts.

Definition 6: A *proper* path precept will, if applied continuously while moving from the start place of the path, bring the agent to the goal place.

Note that, in an unpredictable real-world application, this property of path precepts is a random variable; a precept might be more or less proper depending on its success rate. Also, a dynamic world implies that path precepts may cease to be proper due to changes in the environment.

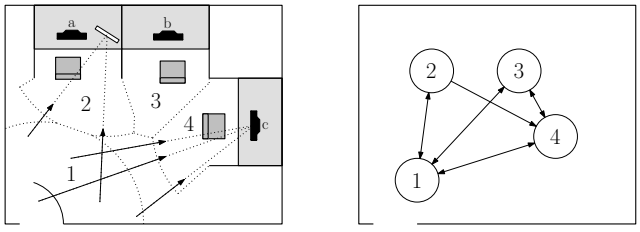
A. Principles for path precepts

The fundamental attribute of a proper path precept is that the output brings the agent to the place to which the path is leading. Places, in turn, are defined in terms of descriptors. These two facts give rise to the following basic rule for creating proper path precepts:

Remark 1: A path precept should be defined such that it, given a sensory state, produces a control output that is expected to increase the relative (compared to those of competing places) applicability function of the goal place.

Thus, the form of the precept naturally arises from the descriptor that define places: A precept that keeps successfully increasing the applicability function must eventually reach the goal place; conversely, the goal cannot be reached without increasing it. Obviously, the method of accomplishing this can vary. Local hill-climbing approaches are general, but suffer from local maxima, whereas global maximization though more robust requires more information and sophisticated control. The actual control policy chosen will depend on available sensory information, control outputs, and efficiency considerations.

Remark 2: If the instantiation permits applicability to be evaluated outside of the immediate surroundings of the



(a) Paths leading from place 1 to places 2 and 4.

(b) Place graph for the office.

Fig. 3. Examples of paths.

current configuration $\boldsymbol{x} \in \mathcal{C}$ and if the control output is of an abstraction level that admits set-points in $\mathcal{C}$, then the following specialization of the above rule can be made:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{C}}{\operatorname{argmax}} \left(\chi_i(\mathbf{x}) - \max_{j\neq i} \chi_j(\mathbf{x}) \right) \tag{18}$$

where $\mathbf{x}^*$ is the set-point for the agent's controller, i is the goal place, and χ_k the applicability function for place k.

The above principles may still leave some ambiguity as to the precise contents of the precept; different descriptors may suggest entirely different movement rules, and the way different descriptors change with movement may be more or less easy to predict in varying sensory circumstances. Any implementation that mixes different types of descriptors will therefore require a facility for estimating the applicability of the goal place at a distance – or at least, caching such information when it is available – and, based on this, producing a local navigation goal for lower-level navigation to carry out.

Apart from being proper, a path precept also needs to be well-defined for all sensor states. Moreover, it should be efficient in execution (i.e. minimizing the time, distance, energy etc. necessary to reach the goal) and efficient to evaluate (i.e. computationally).

B. Example

As a simple example of path precepts derived from place descriptors, regard the office in Figure 2(a). The simplicity of each place's applicability function makes it easy to define path precepts through Remark 2. Take for example $i = 2$:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^2}{\operatorname{argmax}} \left(\chi_2(\mathbf{x}) - \max_{j\neq 2} \chi_j(\mathbf{x}) \right) = p_{pic}$$

In other words, the precept is simply to move towards the picture in order to reach place $\mathcal{P}_2$. Figure 3(a) illustrates how different points in place $\mathcal{P}_1$ will give rise to different trajectories into place $\mathcal{P}_2$, and correspondingly for place $\mathcal{P}_4$. Note that once the agent enters the goal place and detects this, there's no point in continuing to the set point; the path precept is simply meant to take it within the boundary of the place.

The above path precept for $\mathcal{P}_2$ does not necessarily work as well in $\mathcal{P}_3$ and $\mathcal{P}_4$, however. If it is assumed that the agent is unable to detect the picture behind the partition (such as by virtual sensing), or if it lacks the obstacle avoidance capacity to approach p_{pic} except by a straight line, then this path precept is not proper to the paths from $\mathcal{P}_3$ and $\mathcal{P}_4$ to $\mathcal{P}_2$.

In the same way, the natural path precept from $\mathcal{P}_2$ to $\mathcal{P}_3$ (moving toward computer b) is not proper to that path. Figure 3(b) shows a graph containing the paths which have proper precepts. Note that the path from $\mathcal{P}_2$ to $\mathcal{P}_4$ is more proper than its reverse.

The distinction between proper and improper path precepts is not clear-cut even in this simple example: there are points in $\mathcal{P}_3$ from which the picture in $\mathcal{P}_2$ is visible, and points in $\mathcal{P}_2$ where the computer in $\mathcal{P}_4$ cannot be seen.

If the room is rearranged, as in Figure 2(b), then while the path precepts remain the same (being defined as in Remark 2) they will no longer be proper or improper to the degree indicated by the graph in Figure 3(b). An agent relying on that information to navigate in the office may fail to do so, but can update its representation by invalidating paths that fail and creating new ones from the unchanged precepts.

VI. Localization

According to the definition of *places* in Section III, given the true values of place descriptors (features and spatial relations) $D_t = (\zeta_{1,t}, \ldots, \zeta_{n_f,t}, \rho_{1,t}, \ldots, \rho_{n_r,t})$ obtained at time t for location $\boldsymbol{x}(t)$, the place to which that $\boldsymbol{x}$ corresponds is uniquely identified. Consider a function $D(\boldsymbol{x}, t) = (f_1(\boldsymbol{x}, t), \ldots, f_{n_f}(\boldsymbol{x}, t), r_1(\boldsymbol{x}, t), \ldots, r_{n_r}(\boldsymbol{x}, t))$ that provides the true values of place descriptors for location $\boldsymbol{x}$ and time t. Then, for $D_t = D(\boldsymbol{x}(t), t)$, the true place is given by $L_t \triangleq i : D_t \in \mathcal{P}_i$.

However, in the real world an agent is moving through space, following paths to get from place to place and needs to maintain its localization in the face of uncertainty. Let us denote the observation of all descriptors at time t as $\hat{D}_t = D_t + e_t$, where e is an error. We view the agent's progress from place to place as a Markov process with L_t the state at (discrete) time t and $\hat{D}_t$ the measurement. Localization is then carried out iteratively according to the following formula:

$$\begin{aligned} p\left(L_t \mid \{\hat{D}\}_t, \{\alpha\}_{t-1}\right) \\ = \sum_{L_{t-1}} p\left(L_t \mid L_{t-1}, \hat{D}_t, \alpha_{t-1}\right) \\ \times p\left(L_{t-1} \mid \{\hat{D}\}_{t-1}, \{\alpha\}_{t-2}\right) \end{aligned} \tag{19}$$

where $\{\hat{D}\}_t$ represents all measurements up until time t, and equivalently for the actions α.

The probability update in Eq. 19 is computed as follows:

$$\begin{aligned} p\left(L_t \mid L_{t-1}, \hat{D}_t, \alpha_{t-1}\right) \\ = \gamma \cdot p\left(\hat{D}_t \mid L_t\right) p\left(L_t \mid L_{t-1}, \alpha_{t-1}\right) \end{aligned} \tag{20}$$

Here, γ is a normalization constant, and α_t is the action taken at time t; that is, a choice of a path to follow and an according path precept.

The factors in Eq. 20 represent respectively the measurement integration step, and the prediction step, of the localization update.

A. Prediction

The prediction step encapsulates the probability of transitioning from one place to another given the action α_t. If $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ are the configurations at time t and $t+1$ respectively, then

$$\begin{aligned} & p\left(L_{t+1} \,|\, L_t, \alpha_t\right) \\ &= \int\limits_{\boldsymbol{x}_{t+1}} p(L_{t+1} \,|\, \boldsymbol{x}_{t+1}) p(\boldsymbol{x}_{t+t} \,|\, L_t, \alpha_t) d\boldsymbol{x}_{t+1} \\ &= \iint\limits_{\substack{\boldsymbol{x}_{t+1} \in \mathcal{C}_{L_{t+1}} \\ \boldsymbol{x}_t}} 1 \cdot p(\boldsymbol{x}_{t+1} \,|\, \boldsymbol{x}_t, \alpha_t) p(\boldsymbol{x}_t \,|\, L_t) d\boldsymbol{x}_t d\boldsymbol{x}_{t+1} \end{aligned} \tag{21}$$

The factor $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \alpha_t)$ represents the evolution of the exact configuration during the transition, and can be computed via the Fokker-Planck equation (see e.g. [11]); we assume the continuous-time process can be written:

$$\begin{aligned} d\boldsymbol{\xi} &= f_\alpha(\boldsymbol{\xi}) d\tau + N(\boldsymbol{\xi}) d\eta \\ \boldsymbol{\xi}(0) &= \boldsymbol{x}_t \\ \boldsymbol{x}_{t+1} &= \boldsymbol{\xi}\left(\min\{\tau : S_\alpha\left(\boldsymbol{\xi}(\tau), \tau\right) = 0\}\right) \end{aligned} \tag{22}$$

where f_α represents the motion model, given the chosen path precept, and $d\eta$ represents the random evolution of a stochastic process such as a Brownian motion. N is a configuration-dependent transformation of the process noise. The transition ends when the stopping condition S, given by the path precept, evaluates to 0.

The resulting integral is very difficult to compute in general, and an analytic solution will not be feasible except for the very simplest cases.

Because of this, it may be more profitable to view the state transition probabilities as hidden model parameters:

$$p\left(L_{t+1} = j \mid L_t = i, \alpha\right) = \theta_{i,j,\alpha} \tag{23}$$

Given an initial estimate for $\theta_{i,j,\alpha}$ and observations of outcomes of action execution in a real or simulated setting, the parameters can be iteratively estimated through Expectation-Maximization.

The basic constraint is that $\sum_i \theta_{i,j,\alpha} = 1$. Reasonable initial estimates will vary with instantiation, and may be taken from appropriately defined relations; as an example, a transition to a nearby or adjacent place might be assigned a higher probability by default. The simplest assumption is that of uniform probability: $\theta_{i,j,\alpha} = 1/n_P$ where n_P is the number of places.

B. Measurement integration

After the action is finished, the measurement step incorporates observations of descriptors into the probability distribution. As is seen below, this expression is complicated by the fact that knowing the place does not imply a probability distribution over exact locations $\boldsymbol{x}$, nor over descriptor values D.

Observed descriptor values are conditionally independent of place, given true descriptor values D':

$$\begin{aligned} & p\left(\hat{D}_t \mid L_t\right) \\ &= \int_{D' \in \mathcal{D}} p\left(\hat{D}_t \mid D'\right) p\left(D' \mid L_t\right) dD' \end{aligned} \tag{24}$$

The first factor is simply the likelihood of the observation. Expressed using the probability distribution of the measurement error, it becomes:

$$p\left(\hat{D}_t \mid D'\right) = p_e\left(\hat{D}_t - D'\right) \tag{25}$$

If observation errors are taken to be conditionally independent, given the true descriptor values, the likelihood function can be written:

$$\begin{aligned} & p\left(\hat{D}_t \mid D'\right) \\ &= \prod_{i=1}^{n_f} p\left(\hat{\zeta}_{i,t} \mid \zeta_i\right) \prod_{i=1}^{n_r} p\left(\hat{\rho}_{i,t} \mid \rho_i\right) \\ &= \prod_{i=1}^{n_f} p_{e_i}\left(\hat{\zeta}_{i,t} - \zeta_i\right) \prod_{i=1}^{n_r} p_{e'_i}\left(\hat{\rho}_{i,t} - \rho_i\right) \end{aligned} \tag{26}$$

where e_i and e'_i are the errors associated with the measurement of feature i and relation i, respectively.

The second factor in Eq. 24 represents the way descriptor values are distributed inside places. One way of dealing with it is to assume a normalized distribution of D' over $\mathcal{P}_i$, i.e. a constant. However, this distribution is dependent on the details of the instantiation. If it cannot be modeled or estimated, another approach is to evaluate

$$\begin{aligned} & p\left(D' \mid L_t\right) \\ &= \int_{\boldsymbol{x} \in \mathcal{C}} \delta\left(D' - D\left(\boldsymbol{x}, t\right)\right) p\left(\boldsymbol{x} \mid L_t\right) d^m \boldsymbol{x} \\ &= \int_{\boldsymbol{x} \in \Psi} \frac{p\left(\boldsymbol{x} \mid L_t\right)}{|\nabla D(\boldsymbol{x}, t)|} d^{m-1} \boldsymbol{x} \end{aligned} \tag{27}$$

where Ψ denotes all $\boldsymbol{x}$ which satisfy $D' = D(\boldsymbol{x}, t)$. δ is the Dirac distribution, and the final step uses the generalized scaling property of integrals over Dirac distributions. m is the dimension of $\mathcal{C}$.

$p\left(\boldsymbol{x} \mid L\right)$ can be modeled either as a constant over $\mathcal{C}_{L_t}$ or estimated based on observations. If a place is defined in terms of an applicability function, the spatial information encoded in it can also be used to model this distribution.

VII. Discussion

Despite the fact that the framework presented in the previous section represents a certain view on the structure of a cognitive map, it is also very general and allows for expressing many existing approaches as specific cases. Consider for instance the topological map constituting a part of the Multi-Layered Conceptual Spatial Representation presented in [14]. The authors propose to create a topological representation on top of a two-dimensional metric line map,

and ground each topological node around a point anchored to the metric map. Such approach can be easily expressed in our framework if we define a feature $\zeta = f(\boldsymbol{x}, t) = \boldsymbol{x}$, where $\boldsymbol{x} \in \mathcal{C} = \mathbb{R}^2$ represents the coordinates on the metric map, and a set of applicability functions $\{g_i(\zeta)\}_{i=1}^{n_t}$ such that $g_i(\zeta) = 1/(1 + |\boldsymbol{t}_i - \zeta|)$ for each of the n_t topological nodes, where $\boldsymbol{t}_i$ is the center of the node expressed in the coordinates of the metric line map.

The generality of the presented approach can accommodate a very wide range of different methods for abstracting space into places. Exact grid decomposition [1] as well as fixed decomposition can both be described in terms of this framework, given properly chosen features, as can the "islands of reliability" of [12]. Even a system such as the Spatial Semantic Hierarchy [6] is possible to express in these terms; however, to accomplish this, a relatively high level of abstraction must be assumed for the features and the sensor input. Nevertheless, it is our expectation that such requirements will not apply in general to powerful and cognitively well-founded instantiations of this framework.

A. Future work

Possible directions in which to extend this work include:

1) Feature selection: Within this paper we have assumed a set of features as given. In a practical system, an agent will have access to high-dimensional low-level sensor data and the features used for building scenes will need to be abstracted from this data. This can be done in either a pre-programmed or an automatic manner.

2) Virtualized sensors: Herein, features are defined as functions of single points in configuration space; in effect, a feature is conceived of as an abstract sensor output while the agent is at that point. In practice, techniques that allow information to be integrated over time may serve as "virtual" sensor input permitting more advanced features to be defined.

3) Clustering: This paper has suggested one way of clustering the feature space into scenes using applicability functions. Methods for automatic and dynamically updated clustering could be applied.

4) Spatial reasoning: One principal use for segmenting space, in a cognitive systems context, is high-level spatial reasoning, planning, learning and communication. It would be useful to explore the implications of a feature-based place concept when integrated as a component of a full cognitive system.

VIII. Conclusions

We have presented a general framework for building a spatial map based on places and scenes which supports localization and navigation using arbitrary features and higher-level spatial relations. We suggested how the framework would be used to instantiate a system with cognitively plausible features, as well as how to extract precepts for moving from one place to another. Probabilistic expressions used for localization in the framework were presented and the necessity for additional assumptions was highlighted.

The framework has been shown to entail existing spatial representations. In the future, we hope to demonstrate instantiations built directly on the proposed framework, which will prove the viability of the approach and its usefulness in higher-level reasoning.

References

[1] J. Barraquand and J.-C. Latombe. Robot motion planning: a distributed representation approach. *International Journal of Robotics Research*, 10(6), 1991.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006.

[3] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6), 2008.

[4] A. Ganguli, J. Cortes, and F. Bullo. Maximizing visibility in non-convex polygons: Nonsmooth analysis and gradient algorithm design. *SIAM Journal on Control and Optimization*, 45(5), 2006.

[5] A. S. Huang and S. Teller. Non-metrical navigation through visual path control. Technical Report MIT-CSAIL-TR-2008-032, 2008.

[6] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, New Orleans, USA, 2004.

[7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[8] M. J. Milford and G. Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, 24(5), 2008.

[9] A. Pronobis, O. Martinez Mozos, and B. Caputo. SVM-based discriminative accumulation scheme for place recognition. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008.

[10] D. A. Randell, Zhan Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the International Conference on Knowledge Representation and Reasoning*, 1992.

[11] H. Risken and T. K. Caugheyz. The fokker-planck equation: Methods of solution and application, 2nd ed. *Journal of Applied Mechanics*, 58(3):860–860, 1991.

[12] S. Simhon and G. Dudek. A global topological map formed by local metric maps. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[13] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, Nice, France, 2003.

[14] H. Zender, Ó. Martínez Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6), June 2008.