A neural-dynamic architecture for flexible spatial language: intrinsic frames, the term "between", and autonomy

Ulja van Hengel, Yulia Sandamirskaya*, Sebastian Schneegans, and Gregor Schöner

Abstract— **Spatial language is a privileged channel of human-robot interaction. We extend a neural-dynamic architecture for grounded spatial language in three ways: First, viewer-centered vs. intrinsic reference frames may be autonomously selected. Using intrinsic reference frames requires estimation of the orientation of the reference object. Second, we employ a similar orientation estimation to enable the use of configurations of reference objects for spatial terms such as "between". Third, we enhance the autonomy of the system so that the required sequence of attentional shifts, coordinate transforms, and selection decisions emerges from the time-continuous neural dynamics. By implementing the approach in a robotic setting we demonstrate that simple feature information obtained from video cameras is sufficient to ground spatial language.**

I. INTRODUCTION

Spatial language is an important communicative channel in many tasks that include locating and manipulating objects in a scene. We humans use language about spatial relations between objects flexibly, applying terms such as "to the left of" or "between" to complex visual scenes both in "where" tasks when answering such questions as "where is the blue pen?", as well as in "what" tasks when answering, e.g., "what object is to the left of the green cup?". Studies on human cognition reveal the complex processes underlying spatial language behaviors [18], [20], including identification, localization, and representation in working memory of the reference and target objects, representation of the spatial terms' semantics, application of these semantic representations to the particular arrangement of objects, alignment of reference frames, and tracking of changing sensory information.

Recently, we have proposed a theoretical framework based on Dynamic Fields (DFs) in which models accounting for human spatial language behaviors and the underlying neural processes can be formulated [17] and be used to enable spatial communication with a robotic system [22]. Based on a neural color-space scene representation and representation of spatial semantic templates, this system is able to ground spatial terms such as "to the left of" or "behind" in a visually perceived scene. The user may ask questions about a visual scene using spatial terms, or she may direct attention to objects in a scene using spatial language. The dynamics of the architecture converges on a spatial or color term response and builds a representation of the target object that may guide the robotic action. The spatial relations in the previous architecture are anchored on a reference object in reference frame aligned with the viewing axis. The representations consist of stable peaks of activation, induced by sequences of inputs that instantiated different types of queries.

In this paper, we generalize the DF framework for spatial language in three ways. First, we enable the flexible use of intrinsic and viewer-centered reference frames, consistent with human spatial language. This requires extending the spatial language architecture with a mechanisms to estimate the intrinsic orientation of the reference object, rotating the spatial semantic templates to be aligned with the reference object, and autonomously choosing one of a set of competing reference frames. Second, we introduce the spatial term "between" as an exemplary model case that requires a configuration of two reference objects. Third, we increase the autonomy of the architecture, by generating the sequence of inputs autonomously from a neural dynamics of sequence generation. We implemented the extended framework to take visual input form a real robotic camera. Our demonstrations illustrate the flexibility of the system when faced with moving objects in the visual input.

Our project contributes in different ways to the literature. We build on the extensive work that has been done on the psychophysics of spatial language [18], [20], [10] including the integration of different reference frames [5], and on the cognitive models that formalize the mappings from space to spatial terms [15], [9], [14]. When models of spatial language are used to endow cognitive robots with natural language capabilities, a broad set of problems in architecture and system integration must be solved [1], [12], [16], [7]. We address a subset of these problems, most critically, we focus on the grounding of spatial language in perceptual representations that are obtained from visual input [21], [19], [11], [13], [4]. Integration of the multiple required processes emerges in the neural dynamic framework we use [8]. We extend earlier process models within that framework [22], [3], [17] through the autonomous selection of intrinsic vs. viewer-centric reference frames from image information and the selection of reference object configurations for terms such as "between". The system can track moving objects while processing spatial language tasks. The enhanced autonomy of the neural dynamics enables it to generate the sequence of internal states entailed in computing spatial relations. Conversely, our work demonstrates that the process models used to account for human data may actually deliver the claimed performance when linked up to real sensors in real environments. Although we do not address learning here, our approach is broadly consistent with the research program of developmental robotics [6].

Institut für Neuroinformatik, Ruhr-Universität Bochum, Universitätstr. 150, 44780 Bochum, Germany

*yulia.sandamirskaya@ini.rub.de

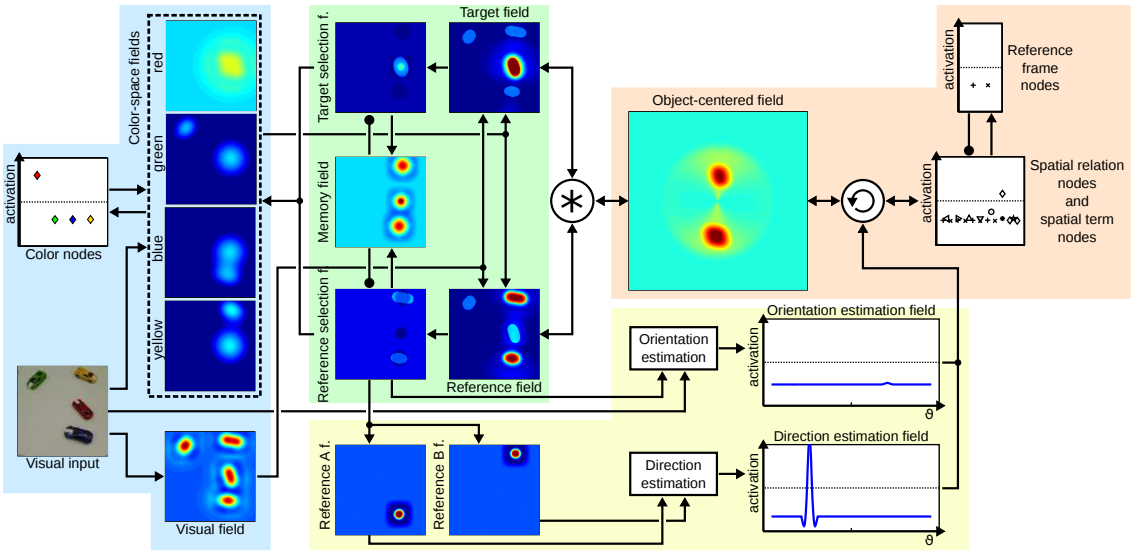

Fig. 1. The neural dynamic architecture is shown together with a snapshot of the activation patterns in the different Dynamic Fields. The snapshot is taken at the final stage of answering the query "what is between the blue and the yellow object?". The response of the model can be read off the color nodes, the elevated activation level of the node for red designating the red object.

II. FRAMEWORK

Here, we describe the extended spatial language framework (Fig. 1).

The scene representation is established by the visual and the color-space dynamic fields (DFs) and the color-term nodes (blue box in Fig. 1). The visual DF constitutes a spatial representation of the scene in image coordinates. All objects in the observed scene induce activity peaks in the visual field at locations that correspond to their positions in the image. The color-space DFs build a single peak over location and color of the selected object. The object may be selected by providing a specific user input to one of the color-term nodes in a "where" task, or it may be autonomously selected by the dynamics of the architecture and a homogeneous boost of all color-term nodes during processing of a "what" task. To provide the color-term input to the system and to represent the color-term output of the architecture, the color-term nodes are reciprocally coupled to the color-space DFs.

The DFs grouped in the green box in Fig. 1 are used to label an object either as the reference object or as the target object. The target and reference DFs receive input from the visual field about all objects in the scene and from the color-space field about the currently selected object. This visual input impacts on either the target or the reference DF depending on which of these DFs is currently brought to the activation threshold by a homogeneous boost. The boosts are introduced sequentially to build the representations of the target and the reference objects from the shared scene representation. Since a task might require representation of multiple target objects (e.g. the task "What objects are to the left from the green object?") as well as representation of multiple reference objects (e.g. in a task "What is between the two objects?") both target and reference DFs support multiple activity peaks. The selection of a single object for processing of the spatial task is done in the reference and target selection fields. The memory DF inhibits locations of the already selected target and reference objects and ensures that each object is selected only once.

The spatial relations ("left", "right", "behind", "in front", "between") between the objects are represented by a set of spatial relation nodes. A spatial relation node may be activated by a specific user input in a "what" task. In a "where" task, all spatial relation nodes are boosted in the final step of the process and the node with the highest activation is selected to yield the spatial-term response of the system. The spatial relation nodes are coupled through the spatial semantic templates to the object-centered DF, which holds a representation of the location of the target object relative to the location of the reference object. The transformation of the spatial representation of the target object from the image-based reference frame to the object-centered reference frame is implemented by a convolution of outputs of the target and the reference selection fields.

The system may use either intrinsic or viewer-centered reference frames to process a scene. Both reference frames are centered on the reference object; the viewer-centered reference frame is aligned with the viewer, whereas the intrinsic reference frame is aligned with the intrinsic orientation of the reference object. The intrinsic orientation of each reference object is estimated in the orientation estimation field. The two reference frames are represented by two dynamical nodes (RF nodes), one of which may be activated by the direct user input or by the dynamics of the architecture through mutual competition. If the intrinsic reference frame is selected, the reference object's orientation estimate is used to rotate the spatial semantic templates for the projective spatial terms.

The processing of a spatial task in the architecture requires sequences of homogeneous boosts to the fields and sets of

nodes. For instance, to process a "where" task, the following sequence of boosts is needed: (1) specific input to one of the color nodes and boost of the target field, (2) boost of the target selection field and the memory field, (3) specific input to the second color node and boost of the reference field, (4) boost of the reference selection field, the memory field, and the orientation estimation field, (5) boost of the object-centered field, (6) boost of the spatial relation nodes and either boost of the reference frame (RF) nodes or input to one RF node, (7) boost of the the spatial term and spatial relation nodes and either boost of the RF nodes of input to one of the RF nodes. Such sequence of boosts may be produced autonomously by a serial order mechanism [23]. A set of the ordinal and the associated memory nodes constitute the ordinal system (Fig. 2). The ordinal nodes are activated sequentially, the sequential transitions are triggered by activation of the condition-of-satisfaction (CoS) nodes. In our architecture, each ordinal node projects onto the specific dynamical fields or nodes of the architecture, which receive a boost or specific input at the respective step in the processing sequence. The CoS nodes detect activation of the boosted structures and trigger autonomous transitions to the next step in the sequence when the boosts have had the desired effect. This mechanism enables autonomous processing of a spatial language task. All tasks of a particular type (e.g. "what?" or "where?") are represented by a set of ordinal nodes.

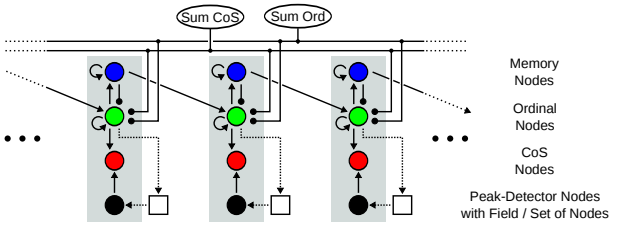

Fig. 2. Schematic of the nodes for sequence generation. Each stage in a sequence is represented by a memory node, an ordinal node, a condition-of-satisfact (CoS) node, and a peak detector node which receives input from a Dynamic Field. Each ordinal node receives inhibitory input from the weighted sum over the output of all CoS nodes and from the ordinal nodes.

III. Mathematical description of the framework

A. The general Dynamic Neural Field equation

The spatial language architecture consists of a number of dynamic fields (DFs) with activation that evolves in time according to the dynamical equation, first analyzed in the context of modeling temporal dynamics of large neural populations by Amari [2]. This equation specifies the rate of change of the activation function, $u(\mathbf{x},t)$, defined over a task-relevant space, $\mathbf{x}$:

$$\tau\dot{u}(\mathbf{x},t) = -u(\mathbf{x},t) + h + \int O(\mathbf{x}',t)K_{\text{lat}}(\mathbf{x}-\mathbf{x}')d\mathbf{x}' + S(\mathbf{x},t) + q\zeta(t). \quad (1)$$

The dynamics of the neural field is characterized by the negative resting level, $h<0$, and the homogeneous lateral interactions, realized by a convolution between the output of the DF, $O(\mathbf{x},t)$, and a sum-of-Gaussians interaction kernel, $K_{\text{lat}}(\mathbf{x}-\mathbf{x}')$. The field receives external input $S(\mathbf{x},t)$ and the activation is modulated by uncorrelated noise $\zeta(t)$.

The output of a DF is shaped by a sigmoidal non-linearity

$$O(\mathbf{x},t) = g(u(\mathbf{x},t)) = \frac{1}{1+e^{-\beta u(\mathbf{x},t)}}, \quad (2)$$

where β is the steepness of the sigmoid.

The interaction kernels throughout the architecture are Gaussian functions,

$$K(\mathbf{x}-\mathbf{x}') = \frac{c}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2\sigma^2}\right), \quad (3)$$

characterized by the amplitudes, c, and widths, σ. In the architecture, we use four different kernel widths, $\sigma_{\text{s}}=6$, $\sigma_{\text{m}}=12$, $\sigma_{\text{l}}=30$, $\sigma_{\text{b}}=45$, and will use the same subscripts to denote the kernels with the respective width, as well as the DF's output, convolved with this kernel. The lateral interaction kernel consists of three terms:

$$K_{\text{lat}}(\mathbf{x}-\mathbf{x}') = K_{exc}(\mathbf{x}-\mathbf{x}') - K_{inh}(\mathbf{x}-\mathbf{x}') - c_g, \quad (4)$$

where the last term is the global inhibition in the field.

The external inputs, $S(\mathbf{x},t)$ in Eq. (1), connect the dynamic fields and dynamic nodes in the architecture. In the following, we list the inputs to all fields and nodes of the spatial language architecture.

The visual field only receives input from the camera, constructed as described in Appendix and smoothed by a medium-width kernel, $K_m(x,y)$:

$$S_{\text{vis}}(x,y) = c_{\text{img}}I_{\text{vis, m}}(x,y). \quad (5)$$

The color-space fields receive input from the camera, I_{cs}, constructed as described in the Appendix and smoothed by the medium-width kernel, K_{m}. Each color-space field is coupled a color node and receives a homogenous boost, proportional to the node's output, O_{col}. The output of the reference selection field, O_{sr}, and the target selection field, O_{st}, smoothed by a broad kernel, support locations in the color-space fields that correspond to the selected object.

$$S_{\text{cs}}(x,y,c) = c_{\text{img}}I_{\text{cs, m}}(x,y,c) + c_{\text{col}}O_{\text{col}}(c) + c_{\text{sr}}O_{\text{sr, b}}(x,y) + c_{\text{st}}O_{\text{st, b}}(x,y). \quad (6)$$

The color nodes receive a user-generated input, $m_{\text{col}} = \{0,1\}$, which boosts one of the nodes in the beginning of a "where" task, and the output of the associated color-space field, $O_{\text{cs}}(x,y,c)$, which, combined with the boost, b_{col}, from the ordinal system activates one of the color nodes to represent the color-term answer in a "what" task:

$$S_{\text{col}}(c) = c_{\text{col,cs}}\iint O_{\text{cs}}(x,y,c)\,\text{d}x\text{d}y + c_{\text{cin}}m_{\text{col}}(c) + b_{col}. \quad (7)$$

The reference field receives as input the output of the visual field, $O_{\text{vis}}(x,y)$, a sum of outputs from the color-space fields, the output of the transformation of the object-centered field through the target field, $T_{\text{obj, tar}}(x,y)$, and a

homogeneous boost from the ordinal system, b_{ref}:

$$S_{\text{ref}}(x,y) = c_{\text{vis}} O_{\text{vis}}(x,y) + c_{\text{cs}} \sum_{c \in C} O_{\text{cs}}(x,y,c) + c_{\text{ref,obj}} T_{obj,tar}(x,y) + b_{ref}. \quad (8)$$

Here, the spatial transformation (shift) is computed as an outer product (convolution) of the outputs of the object-centered and the target fields:

$$T_{\textbf{obj, tar}}(x,y) = \iint O_{\text{tar}}(x',y') O_{\text{obj}}(x'-x, y'-y) dx' dy'. \quad (9)$$

The target field receives inputs from the visual field and the color-space field, output of the transformation of the object-centered field through the reference field, $T_{\text{obj, ref}}(x,y)$, and a homogeneous boost from the ordinal system, b_{tar}:

$$S_{\text{tar}}(x,y) = c_{\text{vis}} O_{\text{vis}}(x,y) + c_{\text{cs}} \sum_{c \in C} O_{\text{cs}}(x,y,c) + c_{\text{tar,obj}} T_{obj,ref}(x,y) + b_{tar}. \quad (10)$$

Here, the spatial transformation (shift) is computed as an outer product (convolution) of the outputs of the object-centered and the reference fields:

$$T_{\textbf{obj, ref}}(x,y) = \iint O_{\text{ref}}(x',y') O_{\text{obj}}(x-x', y-y') dx' dy'. \quad (11)$$

The reference selection field is driven by the output of the reference field, $O_{\text{ref}}(x,y)$, and is inhibited by the output of the memory field, $O_{\text{mem}}(x,y)$, which represents objects that have already been selected and processed. This field also receives a homogeneous boost from the ordinal system, $b_{\text{ref,sel}}$. The reference selection field has strong lateral interactions and may sustain only a single activity peak.

$$S_{\text{sr}}(x,y) = c_{\text{ref}} O_{\text{ref}}(x,y) - c_{\text{mem}} O_{\text{mem}}(x,y) + b_{\text{sr}}. \quad (12)$$

The target selection field is driven by the target field and the boost from the ordinal system and is inhibited by the output of the memory field:

$$S_{\text{sr}}(x,y) = c_{\text{tar}} O_{\text{tar}}(x,y) - c_{\text{mem}} O_{\text{mem}}(x,y) + b_{\text{st}}. \quad (13)$$

The memory field receives as input the outputs of the reference selection field and the target selection field and a homogeneous boost from the ordinal system, b_{mem}. The lateral interactions in this field are chosen to enable multiple self-sustaining peaks (no global inhibition).

$$S_{\text{mem}}(x,y) = c_{\text{sr}} O_{\text{sr}}(x,y) + c_{\text{st}} O_{\text{st}}(x,y) + b_{\text{mem}}. \quad (14)$$

The object-centered field has no lateral interactions and is driven by the output of the spatial relation nodes, $O_{\text{spr}}(s), s \in T = \{\text{left}, \text{right}, \text{in front}, \text{behind}, \text{between}\}$, shaped by the spatial semantic templates, $W(x,y,s)$, the summed over color output of the color-space fields, smoothed by a medium-width kernel, K_m, the output of the transformation between the target and the reference object, $T_{\text{tar,ref}}$, and a homogeneous boost, b_{obj}:

$$S_{\text{obj}}(x,y) = c_{\text{spr}} \sum_{s \in T} O_{\text{spr}}(s) W(x,y,s) + c_{\text{cs}} \sum_{c} O_{\text{cs, m}}(x,y,c) + c_{\text{tar,ref}} T_{\text{tar,ref}}(x,y) + b_{\text{obj}} \quad (15)$$

The transformation (reference frame shift) is computed as convolution:

$$T_{\text{tar,ref}}(x,y) = \iint O_{\text{ref}}(x',y') O_{\text{tar}}(x-x', y-y') dx' dy' \quad (16)$$

Spatial term nodes receive a user input $m(s) = \{0,1\}$ that activates one of the nodes, selected by the user in the beginning of a "what" task, the output of the spatial relation nodes, $S_{\text{spr}}(s)$, and a boost from the ordinal system, b_{spt}:

$$S_{\text{spt}}(s) = c_{\text{spr}} S_{\text{spr}}(s) + c_{\text{sin}} m(s) + b_{\text{spt}}. \quad (17)$$

Spatial relation nodes receive output of the spatial term nodes, $O_{\text{spt}}(s)$, as input. Each projective spatial term in our architecture $(\text{left}, \text{right}, \text{in front}, \text{behind})$ corresponds to two spatial relation nodes: a viewer-centered and an intrinsic one. The two reference frame (RF) nodes, $O_{\text{rf}}(r), r \in$ intrinsic, viewer-centered, represent the currently selected (viewer-centered or intrinsic) reference frame. An active RF node inhibits the spatial relation nodes, which do not correspond to the selected reference frame. Output of the object-centered field is multiplied with the spatial semantic templates (Eq. 19) to provide input, $S_{\text{spr,obj}}(s,r)$, which determines the spatial relation to be selected after processing of a "where" task:

$$S_{\text{spr}}(s,r) = c_{\text{spt}} O_{\text{spt}}(s) - c_{\text{rf}} O_{\text{rf}}(r) + c_{\text{obj,exc}} S_{\text{spr,obj}}(s,r), \quad (18)$$

where

$$S^r_{\text{spr,obj}}(s) = \begin{cases} \iint O_{\text{obj}}(x,y) W(x,y,s)\, \text{d}x\text{d}y, \\ \text{if } r = \text{viewer-centered} \\ \iint O_{\text{obj}}(x-x', y-y') W(x',y',s)\, \text{d}x'\text{d}y', \\ \text{if } r = \text{intrinsic}. \end{cases} \quad (19)$$

The reference frame nodes are driven by the spatial relation nodes, $O_{\text{spr}}(s,r)$, and the user input, $m(r) = \{0,1\}$. The reference frame nodes are forced to make a selection decision by the boost from the ordinal system, b_{rf}.

$$S_{\text{rf}}(r) = c_{\text{spr}} \sum_{s \in T^r} O_{\text{spr}}(s,r) + c_{\text{rin}} m(r) + b_{\text{rf}}. \quad (20)$$

The auxiliary reference fields A and B (or short refA field and refB field) are driven by the reference selection field and the homogeneous boost from the ordinal system, which ensure that the two reference objects are input sequentially to the field A and B:

$$S_{\text{ra}}(x,y) = c_{\text{sr}} O_{\text{sr}}(x,y) + b_{refa}, \quad (21)$$

$$S_{\text{rb}}(x,y) = c_{\text{sr}} O_{\text{sr}}(x,y) + b_{refb}. \quad (22)$$

The orientation estimation field is a one-dimensional DR defined over the orientation angle, θ. This field receives input from the rotation estimation system and stabilizes the representation of the estimated orientation:

$$S_{\text{oe}}(\theta) = c_{\text{oe}} a_{\text{oe}}(\theta). \quad (23)$$

The direction estimation field, similarly to the orientation estimation field, receives input from the direction estimation

system and stabilizes representation of the estimated direction:

$$S_{\mathrm{de}}(\theta) = c_{\mathrm{de}} a_{\mathrm{de}}(\theta). \tag{24}$$

The orientation estimation of the reference object, direction estimation and scaling of the "between" pattern are performed by a dynamical mechanism outside DF framework and are described briefly in the Appendix

IV. RESULTS

We present several experiments using two different tasks to demonstrate the capabilities of the framework: Either the framework has to locate and identify a target object from a given spatial description, containing a reference object and a spatial term ("what" task); or it has to determine a spatial relation between given target and reference objects ("where" task). The visual input is a video stream of a table top scene containing multiple toy cars in changing arrangements. A white marker is applied to the front end of each car to simplify identification of the car's intrinsic reference frame. The connections within the framework and their parameters remain fixed for both tasks and all experimental settings. The different behaviors of the framework are generated by a series of control inputs together with the inputs from the concrete task, which are all organized and supplied at the appropriate time by the autonomous sequence generation mechanism.

First, we will describe exemplarily how the framework progresses through the individual sequence steps to solve the "what" task. Next, we present two demonstrations that highlight how the framework can deal with moving objects. We illustrate the basic tracking ability of the system in an instance of the "what" task, in which we query the system to identify the target object again after it has moved to a new location; and we show how the framework is able to change its answer in the "where" task to reflect a changed position of the target object. Finally, we illustrate the flexible use of reference frames in the model using the "where" task in different settings: We demonstrate how the reference frame for the spatial description can either be selected by an explicit external input, or be chosen autonomously dependent on the arrangement of objects in the scene.

A. Locating a target object ("where" task)

This demonstration shows in detail how the framework processes the task "What is between the blue and the yellow car?" This task requires first to localize the blue car and yellow car as reference objects in the scene, then to determine the region that matches the spatial description "between", select a target object from this region and give its color as a response. To solve the task in the framework, it is divided into nine elementary steps. Each step activates specific external inputs and/or global boost inputs to certain fields. This induces the transition of the framework dynamics to a new attractor state, typically through the formation of a peak. A peak detector then activates the CoS node for the step and triggers the transition to the next step. The resulting end state of the framework after these steps is shown in Fig. 1, and the corresponding time course is shown in Fig. 3.

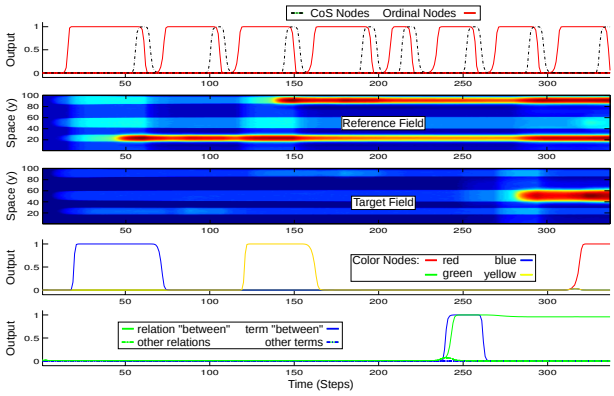

Fig. 3. Time course of activation for selected fields and nodes for the task "what is between the blue and the yellow car?" The top panel shows the ordinal and the CoS nodes, indicating each step in the sequence. The second and third panel from top show the activation patterns projected onto the vertical visual dimension for the reference and the target fields as a function of time. The two bottom panels show the output of the color nodes, and of the spatial relation and spatial term nodes. See text for an explanation of what happens.

The two reference objects are selected sequentially, starting with the blue one. The first ordinal node provides the specific input to the color node "blue" and a homogeneous boost to the reference field. A peak for the location of the red toy car first forms in the "blue" color-space field, and in turn induces a peak at the same location in the boosted reference field. This triggers the first CoS node via a peak detector, and thereby induces a transition to the next sequence step. The peak in the reference field remains stable even without the inputs. The second ordinal node provides boost inputs to the reference selection field, memory field, and refA field. This induces peaks for the previously selected location in all three fields, of which the ones in the memory and refA field remain stable for the remainder of the task. In the following two steps, the yellow car is selected as reference object in an analogous fashion and its location represented by a second peak in the reference and memory field as well as a peak in the refB field.

In the next step, the locations of the peaks in the refA and refB fields are used in the direction estimation module to determine the direction of the line between the reference objects. The direction estimation field is boosted in this step, and forms a peak based on the input from this module. The distance between the reference objects is also determined and used to adapt the semantic pattern for the term "between" (which should reflect the range and orientation of the region between the reference objects). This adaptation is done continuously and does not require its own sequence steps. The transition to the next step occurs as soon as the direction estimation field has formed a peak.

The next ordinal node activates the specific task input to the spatial term node "between" and simultaneously boosts all spatial relation node. This causes the spatial relation node "between" to become active, and the associated spatial

semantic template for "between" is projected into the object-centered field. The activation of the spatial relation node triggers the transition to the next step. At this point, there is significant activation in both the object-centered and the reference field (with the two peaks still standing there). These fields together feed into the target field via the multi-directional convolution operation. Effectively, this convolution shifts the semantic spatial template from the object-centered field so that it is aligned with the reference object locations, and projects this shifted pattern into the target field. At the same time, the target field receives continuous input from the visual field reflecting the spatial locations of all objects. The next ordinal node now boosts the target field, and through the combined input a peak forms at the location of the one object in the scene that matches the spatial description. The formation of that peak serves as CoS for this step.

In the last sequence step the target selection field and the color nodes are boosted. The target selection field selects a peak from the target field by forming a peak at the same position. Once the peak has formed, its output is used to highlight the selected position in all color-space fields, where it matches the location of one of the visual stimuli. Here, this match occurs in the "red" field, and its output activates the "red" color node. When the "red" color node reaches a certain activation level, the last CoS is fulfilled and the task sequence ends.

B. *Visual tracking of a moving target object*

The neural dynamics in the framework allow the autonomous tracking of moving objects without requiring special treatment. In particular, the reference and the target field are continuously coupled to the visual input via the visual field. If objects in the scene move, the activation peaks follow this movement. We present an instance of both the "what" and the "where" task, processed in the usual manner, in which the target object changes its position after the initial response. We then query a second response from the system by repeating the last steps of the sequence.

In the first experiment, we ask "What is between the red and the green car?" A time course is shown in Fig. 4. The system identifies the target object, then the object starts moving and the peak in the target field keeps tracking its position. The initial spatial description has effectively been used to create a "spatial label" for this object. The tracking continuous when the object leaves the "between" region, albeit with a slightly decreased activation level of the peak. When an identifying response is queried again by boosting the color nodes after the movement has stopped, the framework still produces the correct response "yellow".

For the second experiment, the task is "Where is the blue car relative to the green car?", and the time course is shown in Fig. 5. This task is processed by a different sequence of ordinal nodes: First, the blue car is selected as target object, then the green car is selected as reference object. Next, the spatial relation nodes and reference frame nodes are boosted. When the spatial relation node "intrinsic in front" and the reference frame node "intrinsic" is active, the spatial term nodes are boosted, as well. This leads to the selection of the spatial term "in front" as the framework's response for the task. After this first response, the target object moves to a new location. The system is released from its initial response by de-boosting the spatial relation nodes, spatial term nodes, and reference frame nodes. Target and reference object remain selected. We repeat the last step of the task sequence to produce a new spatial description, and this time the framework activates the spatial relation node "viewer-centered right" and the reference frame node "viewer-centered". This yields the response "to the right" (in a viewer-centered reference frame), which correctly describes the new spatial relation between target and reference object.

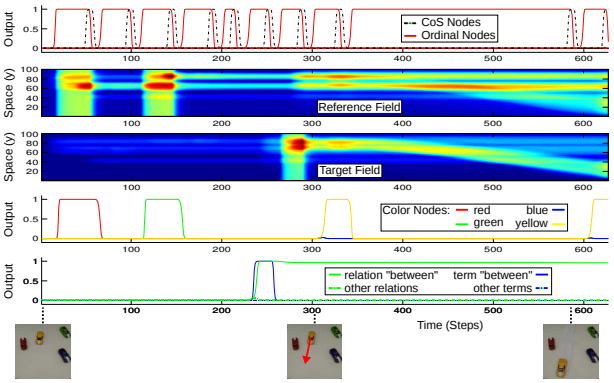

Fig. 4. Time courses of inputs and activation levels for the task "What is between the red and the green car?" The activation of the target field and the reference field is summed along the horizontal spatial dimension.

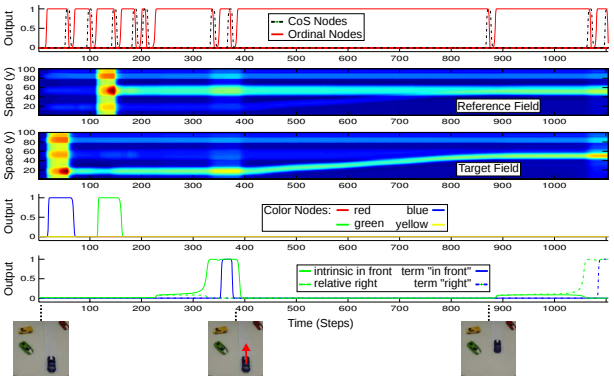

Fig. 5. Time courses of inputs and activation levels for the task "Where is the blue car relative to the green car?". The activation of the target field and the reference field is summed along the horizontal spatial dimension.

C. *Switching between different reference frames*

The reference frame for a task can either be selected by an explicit external input to the reference frame nodes (which is conveyed in the appropriate step by the ordinal dynamics), or it can be chosen autonomously by the framework to find the best fitting spatial description for the specified objects.

Fig. 6 illustrates how the reference frame given by an external input affects the behavior of the framework. Shown

is the activation of the object-centered field after processing the task "Where is the green car relative to the red one?" The experiment is performed once with the viewer-centered reference frame specified (yielding the activation in Fig. 6a) and once with the intrinsic reference frame (Fig. 6b). The visual and task input is the same for both trials (Fig. 6c), yielding in both cases the same relative position of the target to the reference object in the object-centered field (the spot of high activation indicated by the red circle). In contrast, the shape of the spatial semantic template visible in the figures changes depending on the selected reference frame and the chosen spatial relation. When the viewer-centered reference frame is selected, the system chooses the response "viewer-centered right", which only provides a moderate match for this spatial arrangement (Fig. 6a). In the intrinsic reference frame, the response "intrinsic behind" is chosen, which yields a more precise match of the relation between the two cars. This shows that the framework can use both reference frames equally.

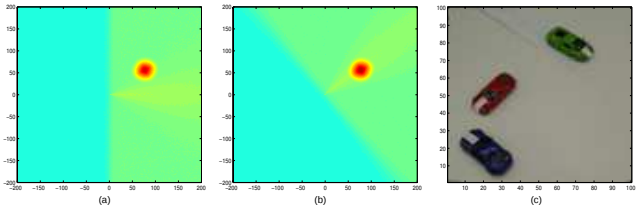

Fig. 6. Fig. (a) and (b) show the activation of the object-centered field in the final stage of processing a "where" task. In Fig. (a) the viewer-centered reference frame is specified. In Fig. (b) the intrinsic reference frame is specified. Fig. (c) shows the visual input for both trials.

In a second experiment we allow the framework to select a reference frame autonomously. The same task is processed with different visual inputs: The locations of the red and blue car remain fixed, but the green car (the target object in the task) is moved in small steps from trial to trial, describing a semi-circular path around the reference object. Fig. 7 illustrates chosen responses dependent on the target location. The system switches between viewer-centered and intrinsic reference frame to select the spatial description that best matches the arrangement of the two cars: It responds with "viewer-centered behind" for angles between target and reference object from -11.31° to 24.1°, switches then to "intrinsic behind" between 23.39° and 80.36°, then to "viewer-centered right" between 71.92° and 111.8°, and then to "intrinsic left" between 108.4° and 121.3°. We note, however, that the reaction time (number of time steps between the activation of the ordinal node and the CoS node for the response selection step) is longer at positions where two spatial relations provide a good match.

V. DISCUSSION

We extended a neural-dynamic framework that grounds spatial language in robotic settings. Spatial relations relative to the intrinsic reference frame of the reference objects can be obtained and the system can autonomously choose between

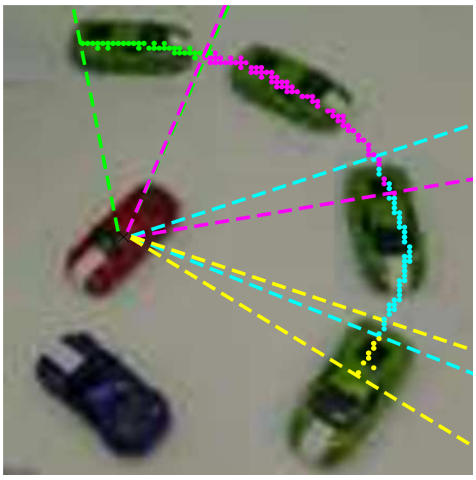

Fig. 7. The target object (green car) was presented at different positions along an imaginary circle. Four samples of visual input are show. Colored dots near the green cars indicate the positions of peaks in the object-centered field for different inputs. In the sector demarcated by green dashed lines, the model responds "viewer-centered behind". Analogously, the other response sectors are marked: magenta for "intrinsic behind", cyan for "viewer-centered right", and yellow for "intrinsic left".

the ego-centric and the intrinsic reference frame. We extended the spatial vocabulary by the spatial term "between", that brings in configurations of references objects, whose orientation matters. We showed that the system can track objects and update spatial labels in changing environments. The neural dynamics operating in real time autonomously generates the sequence of inputs and boosts that is need to ground spatial language.

VI. APPENDIX

A. Camera input.

The camera image, $I(x_i,y_i)$ ($(x_i,y_i) = 100$x100 pixels), provides external input to two structures in the architecture: the visual field and the color-space fields.

The color-space fields are a discrete along color-dimension version of the three-dimensional color-space field. Input to the color-space fields is computed by assigning color ranges to one of the basic colors, $c \in C = \{\text{red}, \text{green}, \text{blue}, \text{yellow}\}$. A binarized map $I^c_{\text{cs}}(x_i,y_i)$ is extracted for each color by thresholding the image in the 'saturation' (threshold θ_s) and 'value' (threshold θ_v) dimensions of the HSV color-space:

$$I_{\text{cs}}(x_i,y_i,c) = \begin{cases} 1, & \text{if } h(I(x_i,y_i)) \in H^c \\ & \wedge s(I(x_i,y_i)) > \theta_s \\ & \wedge v(I(x_i,y_i)) > \theta_v \\ 0, & \text{otherwise} \end{cases}, \tag{25}$$

where the HSV space if defined by $h \in [0^\circ, 360^\circ)$, $s \in [0,1]$ and $v \in [0,1]$. The discretization is done algorithmically here, but would be the result of the DF interactions in a continuous three-dimensional implementation of the color-space field. The input maps are resized to match the array of the numerically discretized DFs and the following we will denote this input $I_{cs}(x,y,c)$.

The camera input to the visual field is computed as

$$I_{\text{vis}}(x,y) = \sum_{c \in C} I_{cs}(x,y,c). \tag{26}$$

$H^{\text{red}} = [0^\circ, 30^\circ) \cup [300^\circ, 360^\circ)$, $H^{\text{green}} = [90^\circ, 150^\circ)$, $H^{\text{blue}} = [150^\circ, 300^\circ)$, $H^{\text{yellow}} = [30^\circ, 90^\circ)$, $\theta_s = 0.57$ and $\theta_v = 0.58$.

B. Direction estimation and scaling of the 'between' pattern.

In order to apply the spatial pattern 'between' to the current scene, (1) the orientation of the axis connecting the two reference objects had to be estimated to align the spatial semantic template with this axis, and (2) the distance between the reference objects has to be estimated to scale the semantic template. To accomplish this, a representation of the location of one of the reference objects relative to the other reference object is computing by convolving the outputs of the auxiliary reference fields A and B:

$$M(x,y) = \iint O_{\text{refA}}(x',y') * O_{\text{refB}}(x-x',y-y')dx'dy' \qquad (27)$$

The result of this operation, $M(x,y)$, is transformed to the polar coordinates, $M_{\text{polar}}(\rho,\theta)$. Projecting the $M_{\text{polar}}(\rho,\theta)$ to the distance dimension (Eq. 29) or the angular dimensions (Eq. (**??**)), we receive estimation of the orientation of the line connecting the two reference objects and the distance between them respectively:

$$d(\rho) = \sum_{i=1}^{\theta_{\max}} M_{\text{polar}}(\rho,i), \qquad (28)$$

$$a_{de}(\theta) = \sum_{i=1}^{\rho_{\max}} M_{\text{polar}}(i,\theta). \qquad (29)$$

The estimation $a_{de}(\theta)$ is input into the direction estimation field.

The output of the direction estimation field is used to rotate the default "between" pattern $P_{\text{default}}(\rho_i,\theta)$ to align it with the orientation of the reference objects:

$$P_{\text{direction}}(\rho_i,\theta) = P_{\text{default}}(\rho_i,\theta) * O_{\text{de}}(\theta) \qquad (30)$$

To scale the "between" pattern $P_{\text{direction}}(\rho,\theta)$ according to the distance between both reference objects, it is multiplied with $d'(\rho)$:

$$P_{\text{cut}}(\rho,\theta_i) = P_{\text{direction}}(\rho,\theta_i) \cdot d'(\rho) \qquad (31)$$

The result $P_{\text{cut}}(\rho,\theta)$ is then transformed back into Cartesian coordinates to get the transformed spatial semantic template for the term "between", $P_{\text{cart}}(x,y)$.

C. Orientation estimation

Estimation of orientation of the reference object for intrinsic reference frames is accomplished in a similar fashion. A match is computed between the output of the reference selection field, $O_{\text{sr}}(x,y)$ and a template, $T_{\text{polar}}(\rho,\theta)$, that contains a canonical view of the shape of the reference object in polar coordinates:

$$a(\theta) = \sum_{s=1}^{\rho_{\max}} \sum_{t=1}^{\theta_{\max}} T_{\text{polar}}(s,t) \cdot M_{\text{polar}}(s,\theta+t), \qquad (32)$$

which is normalized with:

$$a_{\text{oe}}(\theta) = \sigma\left(\frac{250 \cdot (a(\theta) - \min(a(\theta)))}{\sum_{j=1}^{\theta_{\max}}(a(j) - \min(a(\theta)))} - 1{,}5\right), \qquad (33)$$

with $\beta = 50$, $x_0 = 0$, where a_{oe} is input for the orientation estimation field. The output O_{oe} of the field is used to shift any projective spatial semantic template $P_{\text{polar}}(\rho,\theta)$ in polar coordinates (derived from any projective spatial semantic template $P_{\text{cart}}(x,y)$ in Cartesian coordinates) along the angle dimension θ according to the estimated rotation of the reference object relative to its canonical orientation:

$$P_{\text{shift}}(\rho_i,\theta) = P_{\text{polar}}(\rho_i,\theta) * O_{\text{oe}}(\theta) \qquad (34)$$

The result $P_{\text{shift}}(\rho_i,\theta)$ is then transformed back into Cartesian coordinates to get the rotated projective spatial semantic template $P_{\text{rot}}(x,y)$.

REFERENCES

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *The International Journal of Robotics Research*, 17(4):315–337, 1998.

[2] S Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

[3] E. Bicho, L. Louro, and W. Erlhagen. Integrating verbal and nonverbal communication in a dynamic neural field architecture for human–robot interaction. *Front. Neurorobot*, 4(5), 2010.

[4] M. Brenner, N. Hawes, J. Kelleher, and J. Wyatt. Mediating between qualitative and quantitative representations for task-orientated human-robot interaction. In *Proc. IJCAI*, volume 7, 2007.

[5] Neil Burgess. Spatial memory: how egocentric and allocentric combine. *Trends in cognitive sciences*, 10(12):551–7, December 2006.

[6] A. Cangelosi, G. Metta, G. Sagerer, S. Nolfi, C. Nehaniv, K. Fischer, J. Tani, T. Belpaeme, G. Sandini, L. Fadiga, et al. Integration of Action and Language Knowledge: A Roadmap for Developmental Robotics. *IEEE Transactions on Autonomous Mental Development*, in press.

[7] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[8] W. Erlhagen and E. Bicho. The dynamic neural field approach to cognitive robotics. *Journal of Neural Engineering*, 3(3):R36–R54, 2006.

[9] T. Fuhr, G. Socher, C. Scheering, and G. Sagerer. A three-dimensional spatial model for the interpretation of image data. In *IJCAI-95 Workshop on Representation and Processing of Spatial Expressions*, volume 14, 1995.

[10] K.P. Gapp. An Empirically Validated Model for Computing Spatial Relations. *KI-95: Advances in Artificial Intelligence*, pages 245–256, 1995.

[11] Peter Gorniak and Deb Roy. Situated language understanding as filtering perceived affordances. *Cognitive science*, 31(2):197–231, March 2007.

[12] A. Haasch, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, I. Toptsis, GA Fink, J. Fritsch, B. Wrede, and G. Sagerer. Biron–the bielefeld robot companion. In *Proc. Int. Workshop on Advances in Service Robotics*, pages 27–32. Citeseer, 2004.

[13] N. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G. Kruijff, M. Brenner, G. Berginc, and D. Skocaj. Towards an Integrated Robot with Multiple Cognitive Functions. *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 22(2):1548, 2007.

[14] J. Kelleher and F. J. Costello. Applying Computational Models of Spatial Prepositions to Visually Situated Dialog. *Computational Linguistics*, 35(2):271–306, 2009.

[15] J. Kelleher and J. van Genabith. A computational model of the referential semantics of projective prepositions. *Syntax and Semantics of Prepositions*, pages 211–228, 2006.

[16] T. Laengle, T.C. Lueth, E. Stopp, G. Herzog, and G. Kamstrup. Kantra-a natural language interface for intelligent robots. In *Intelligent Autonomous Systems (IAS 4)*, pages 357–364, 1995.

[17] J. Lipinski, S. Schneegans, Y. Sandamirskaya, J.P. Spencer, and G. Schöner. A neuro-behavioral model of flexible spatial language behaviors. *Journal of Experimental Psychology: Learning, Memory and Cognition.*, 2011.

[18] G.D. Logan. Linguistic and conceptial control of visual spatial attention. *Cognitive Psychology*, 28:103–174, 1995.

[19] N. Mavridis and D. Roy. Grounded situation models for robots: Where words and percepts meet. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 4690–4697. IEEE, 2006.

[20] T. Regier and L. Carlson. Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, 130(2):273–298, 2001.

[21] D. Roy. Semiotic schemas: A framework for grounding language in the action and perception. *Artificial Intelligence*, 167170-205:170–205, 2005.

[22] Y. Sandamirskaya, J. Lipinski, I. Iossifidis, and G. Schöner. Natural human-robot interaction through spatial language: a dynamic neural fields approach. In *19th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, pages 600–607, Viareggio, Italy, sept. 2010.

[23] Yulia Sandamirskaya and Gregor Schöner. An embodied account of serial order: How instabilities drive sequence generation. *Neural Networks*, 23(10):1164–1179, December 2010.