Research Article Open Access

David Lobato*, Yulia Sandamirskaya*, Mathis Richter, and Gregor Schöner

Parsing of action sequences: A neural dynamics approach

Abstract: Parsing of action sequences is the process of segmenting observed behavior into individual actions. In robotics, this process is critical for imitation learning from observation and for representing an observed behavior in a form that may be communicated to a human. In this paper, we develop a model for action parsing, based on our understanding of principles of grounded cognitive processes, such as perceptual decision making, behavioral organization, and memory formation. We present a neural-dynamic architecture, in which action sequences are parsed using a mathematical and conceptual framework for embodied cognition—the Dynamic Field Theory. In this framework, we introduce a novel mechanism, which allows us to detect and memorize actions that are extended in time and are parametrized by the target object of an action. The core properties of the architecture are demonstrated in a set of simple, proof-of-concept experiments.

Keywords: action parsing, sequence learning, elementary behaviors, dynamic neural fields

***Corresponding Author: David Lobato:** Vision Lab (LARSyS), FCT, University of the Algarve, Gambelas Campus, 8000 Faro, Portugal, E-mail: dav.lobato@gmail.com
***Corresponding Author: Yulia Sandamirskaya:** Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, E-mail: yulia.sandamirskaya@gmail.com
Mathis Richter: Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, E-mail: mathis.richter@ini.rub.de
Gregor Schöner: Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany, E-mail: gregor.schoner@ini.rub.de

1 Introduction

When artificial cognitive systems are meant to interact with humans, they require an ability to interpret human behavior based on their own observations. This may enable such systems to report back to a human about its observations, to assist a human operator in a joint task by selecting a complementary or supportive action, and also to learn new behaviors [1–3]. In natural language processing, *parsing* is the process of analyzing a stream of spoken or written language according to the rules of grammar. Here, we will use this term to refer to the decomposition of a visually observed behavior into a sequence of individual actions. The range of possible actions is constrained to a known set.

Action parsing entails detecting individual actions by determining their beginning and termination in time, and storing their serial order. Some actions are 'simply' movements, which amount to a transition from one state of the motor system to another, as in "move the left arm up". Parsing sequences of such actions requires the segmentation of movements into individual components as well as the categorization of each component [4]. Other actions are oriented toward objects in the environment, as in "grasp the green cup". For such actions, it is not sufficient to merely extract and categorize the movement from the visual stream, but it is also necessary to segment and recognize the target object and to represent the relationship between the action and the object.

In this paper, we are interested in this second type of actions, those directed at objects in the environment. In this setting, the system needs to perform segmentation of objects in the visual array, while simultaneously detecting action boundaries in time and determining the underlying action intentions. Several difficulties arise when the parsing of such goal-directed action sequences is conducted in real-time, along with the observed behavior. First, different cues to object-oriented actions may become available at different moments in time. For instance, a reaching movement may be registered when the hand starts to move, but the object of this action may become apparent only later during the hand's movement. Second, the interpretation of the visual stream may be ambiguous at times. For instance, an observer may need to distinguish the action "move hand forward" from the action "move hand toward the red object", whereas the two actions look identical part of the time. Thus, the temporal organization of different action detectors and object detectors as well as the

process of making decisions based on these detectors become a crucial part of the sequence parsing architecture.

To cope with these challenges, we first introduce a model that entertains 'hypotheses'—different possible interpretations of an observed action—which are subsequently confirmed or rejected. This amounts to a structured representation of actions consisting of three parts: (1) a *condition of initiation* at which a hypothesis is put forward once the first characteristics of a particular action are detected, (2) a *condition of satisfaction*, signaling the confirmation of a hypothesis at the end of an action when all its characteristics are detected, and (3) a *condition of failure* that signals the rejection of a hypothesis.

Second, we address the detection of goal-directed actions, in which an object plays the role that roughly corresponds to a slot in the representation of the grammatical structure of a phrase. For instance, a reach action corresponds to a verb-object phrase with a representation of the reaching motor act (a verb in the language analogy) and the object to which the reach is oriented (the object in language). In some cases, the effector that generates the action (the subject), may also need to be specified. For actions of the hand, which we show in our examples, action concepts are often effector specific (reaching refers to the hand, for instance), obviating the need to explicitly represent the effector system.

We address these challenges of online parsing of the object oriented action sequences in a neural-dynamic architecture that is based on Dynamic Field Theory (DFT) [5, 6]. DFT is a mathematical and conceptual framework, in which perceptual, motor, and cognitive processes are modeled with Dynamic Fields (DFs). DFs represent continuous feature spaces by fields of activation variables that evolve continuously in time as described by an integro-differential equation. Units of representations within these feature spaces are provided by peaks of activation that are dynamically stable states of DFs. Instabilities within the neural dynamics enable the emergence of discrete events from continuous time, such as the detection of salient information, selection among alternatives, as well as the formation of working memory. Within DFT, categorical states may be represented by discrete activation variables that may emerge from inhomogeneities in DFs.

The neural dynamics framework is attractive for online parsing of action sequences because it casts cognitive processes in the same terms as perceptual processes. Thus, continuous visual features that are driven by time varying sensory input may couple directly into the neural dynamics, which then imposes event and category structure on the perceptual flow. Grounding perceptual decisions in the form of the underlying continuous time and feature spaces enables online updating when sensory input changes on the fly.

The price to pay for the ease with which DFs link to low-level online sensory input is the relative complexity of creating sequential transitions. To transition to a new state, the previous state must be rendered unstable, while the next state is selected. The sequential activation of serially ordered action states can be organized through the concatenation of dynamical instabilities. This was established in earlier work on *serial order* in DFT [7, 8], which demonstrated the generation of simple action sequences that were acquired semi-autonomously by observation during a learning session. More complex behaviors may be organized in a neural-dynamic architecture for *behavioral organization*, introduced recently in DFT [9]. In this architecture, actions may be activated in parallel and selected flexibly, based on the current situation, rather than according to a fixed serial order.

The main goal of this paper is to combine the mechanisms for sequence representation, developed in the work on serial order, with the concept of elementary behavior, developed in the work on behavioral organization, and extend the resulting architecture toward a fully autonomous acquisition of action sequences, which consist of different goal-directed actions. We thus move beyond previous demonstrations of sequence generation in DFT by building a system that autonomously generates, confirms, or rejects different hypotheses about the observed actions and their objects as evidence for their identities arrives from the perceptual decisions. Different factors that contribute to the evaluation of the hypotheses arise at different times but are integrated by the neural dynamics through processes of memory formation and decision making.

In this paper we start the research program in a simple scenario that involves three action detectors, for reaching, grasping, and dropping, each associated with a representation of the object at which the action is directed. The cognitive architecture detects objects and actions, builds up hypotheses about the observed actions, confirms or rejects them, and learns sequences of successfully accomplished actions—all autonomously organized by the time continuous neural dynamics. In particular, we show how the system copes with abrupt changes in the perceived action sequence both in terms of the perceived action and in terms of the object at which the action is directed. Only those actions are com-

mitted to memory that were successfully accomplished during a demonstration.

The paper is organized as follows. We begin in Section 2 with a brief description of the methods used throughout the system. The architecture is presented in Section 3. In Section 4 we present result of experiments that demonstrate the robustness and flexibility of the model. The discussion in Section 5 emphasizes the novelty of this work in the context of imitation learning.

2 Methods

2.1 Dynamic Field Theory

Dynamic Field Theory (DFT) [5, 6, 10] is a mathematical framework for modeling embodied cognition. The core element of a DFT architecture is a dynamic field (DF), an activation function $u(x,t) \in \mathbb{R}$ defined over continuous feature dimension(s), x, (e.g., color or space). DFs are a mathematical idealization of the distributions of population activation prevalent in the higher nervous system [11–13]. A DF evolves in continuous time, t, based on an integro-differential equation:

$$\tau \dot{u}(x,t) = -u(x,t) + h + S(x,t) + \int f(u(x',t))\, w(x-x')\, dx', \quad (1)$$

where $\dot{u}(x,t)$ is the rate of change of the activation $u(x,t)$ at location x and time t. τ is a time constant, $h < 0$ a negative resting level, and $S(x,t)$ is time-varying external input. The last term of the equation formalizes neural interactions with field locations representing other feature values, x'. The interaction term integrates the output, $f(u(x',t))$, of the DF at every location x' along the feature dimension, weighted with an interaction kernel, $w(x-x')$. The output is determined from activation levels by a sigmoid function, $f(\cdot)$, that translates negative activation values to zero and positive values to one, with a smooth transition for activation values near zero. The kernel is positive (excitatory interaction) for nearby locations x and x' and negative (inhibitory interaction) for locations at larger distances.

This pattern of lateral interaction together with the output nonlinearity lead to the formation of localized peaks of activation. Such peaks are attractors of the neural dynamics and represent detection decisions about an instance of the feature representation. The detected feature is indicated by the location of the peak along the dimension of the DF. A subthreshold pattern of activation near the resting level is another attractor that exists for sufficiently weak input $S(x,t)$. When localized input is gradually increased to a critical strength, the subthreshold activation pattern becomes unstable and the system switches to a peak attractor that had co-existed bistably with the subthreshold solution up to this point. This detection instability [5] is the core mechanism for detecting discrete events from time-continuous graded changes in input. At these abrupt changes, activation patterns may switch on the macroscopic scale that has an impact in a DFT architecture.

The stability of peak attractors makes it easy to build cognitive architectures out of multiple DFs [10, 14]. As DFs within such architectures are coupled, each component DF retains its attractor solutions, which are structurally robust and resist the change of the dynamics that coupling brings about. This is true until an instability switches the field's regime. DF architectures require that these instabilities be carefully designed and compared to empirical evidence for transitions.

Within DF architectures, fields of different dimensionality may be combined [14]. The extreme limit case are zero dimensional fields, that is, activation nodes that have the following dynamics

$$\tau \dot{v}(t) = -v(t) + h + S(t) + c_{vv} f(v(t)), \quad (2)$$

where $v(t)$ is a discrete (in feature space) activation variable that has self-excitatory coupling of strength, c_{vv}. Such discrete 'nodes' may be thought to arise from inhomogeneous fields, which tend to generate peaks only in particular locations that have become sensitized by learning. Systems of coupled discrete dynamic activation variables essentially form dynamic recurrent neural networks that go through analogous dynamic instabilities as do DFs.

In effect, DF architectures are high-dimensional dynamical systems. Their time evolution flows from solving the dynamics, typically numerically in real time. Discrete events emerge from the continuous dynamics through the detection instability. They are not imposed by an outside algorithm.

2.2 Overview of the model

In the parsing architecture, DFs are used to represent the perceptual features detected in the visual array, which characterize the parsed actions and the objects at which they are oriented. The fields provide the perceptual grounding by receiving direct input from the visual system. Peaks of activation in these DFs represent objects in visual and feature spaces.

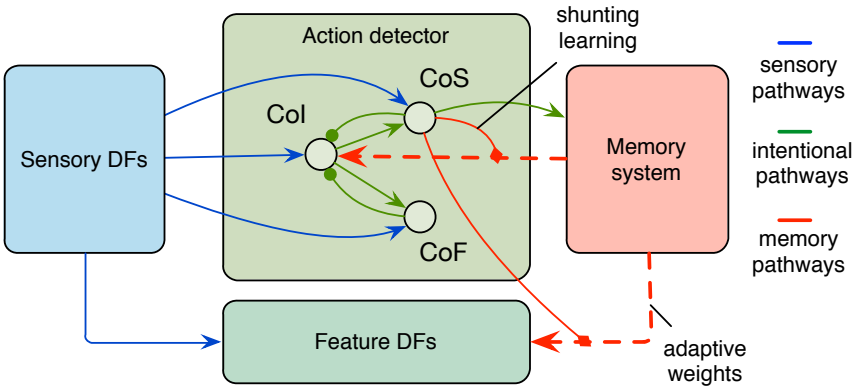

Fig. 1. The parsing model. Lines with arrow heads denote excitatory connections. Lines ending in circles denote inhibitory connections. Red lines are learned memory traces, black lines are shunting terms controlling learning. See text for details.

Discrete dynamic activation nodes are used to represent a number of categories of actions. In particular, each category detector consists of three dynamical nodes (Fig. 1): the condition of initiation node, the condition of satisfaction node, and the condition of failure (or withdrawal) node.

Finally, a set of interconnected dynamical nodes represents the serial order of the detected actions. These nodes are interconnected in a way, which ensures their sequential activation [7]. Each of these ordinal nodes are connected to the action detector and the representation of the respective object, at which the detected action is directed. These connections are realized by a set of adaptive weights, which are modified at a successful detection of an action (signaled by activation of the condition of satisfaction node). We now present these processing elements and the couplings between them in detail.

2.3 Sequential organization of actions

Our approach to the representations of actions in a sequence builds on earlier work on serially ordered sequences [7] and behavioral organization [9] in DFT. The core element of the sequence generation mechanism, developed in these previous works, is the structure of an elementary behavior, which consists of a representation of the action intention and of its condition of satisfaction (CoS), each implemented as a dynamical node. The intention node drives the motor system of the agent and pre-activates the CoS node, which becomes in this case sensitive to the sensory input that corresponds to the accomplished action. The CoS node is activated through a detection instability when it detects a successful accomplishment of the intended action. The activated CoS node inhibits the respective intention node. This gives way to the activation of the next action intention within the network of activation nodes.

In this paper, we extend this structure of an intentional action (elementary behavior) to enable its use in the perceptual mode, i.e., when an action sequence is observed. First, we connect the intention node to the perceptual system in such a way that a given intention is activated when the respective action is observed. We call this extended intention representation a condition of initiation (CoI) node. This CoI represents the hypothesis that a given action is now being performed. The hypothesis is confirmed when the action is successfully terminated as reflected by the activation of the corresponding CoS node. The CoI and CoS nodes are connected to the intention and CoS nodes of the action generation system (which resembles the mirror neurons discourse in neuroscience of action perception), however this coupling will not be investigated here. Second, in order to reject a hypothesis, which did not lead to detection of a successfully accomplished observed action, we introduce a new element into the structure of an elementary behavior—the condition of failure (CoF), which deactivates the CoI that was not confirmed. Section 3.2.3 describes the resulting action detectors, implemented in this work.

2.4 Serial order memory

To store a parsed sequence of actions in memory, we use a neural-dynamic model for serial order [7]. In this model, the order of items in a sequence is represented by a set of interconnected ordinal nodes (Equation (3)).

The ordinal nodes represent ordinal positions in a sequence (i.e., first, second, third). Each ordinal node has an excitatory connection to the respective memory node, which keeps the position in the sequence in the transition phase, when all ordinal nodes are inhibited. The memory node in turn has an excitatory connection to the next ordinal node, thus biasing activation transfer toward the node encoding the next position in a sequence during the transition phase. At any time, only one ordinal node may be active and represents the current ordinal position in the observed sequence. A sequential transition is triggered by a CoS node that inhibits all ordinal nodes including the currently active one. As a result, input is removed from the CoS node itself, which is thus deactivated. That in turn releases the ordinal set from inhibition. The next ordinal node in the serially ordered array goes through the activation threshold first, because it receives additional activation from the memory node of the previously active position. In this way, the ordinal nodes are activated in a sequence by CoS signals from the action detection system.

$$\begin{aligned}\tau \dot{v}_i^{\mathrm{ord}}(t) = &- v_i^{\mathrm{ord}}(t) + h^{\mathrm{ord}} + c_{\mathrm{exc}} g(v_i^{\mathrm{ord}}(t)) \\ &- c_{\mathrm{inh}} \sum_{i \neq i'} g(v_{i'}^{\mathrm{ord}}(t)) + c_{\mathrm{o,m}} g(v_{i-1}^{\mathrm{mem}}(t)) \\ &- c_{o,c} g(v_{\cos}(t)); \qquad (3)\end{aligned}$$

$$\begin{aligned}\tau \dot{v}_i^{\mathrm{mem}}(t) = &- v_i^{\mathrm{mem}}(t) + h^{\mathrm{mem}} + c_{\mathrm{exc}} g(v_i^{\mathrm{mem}}(t)) \\ &- c_{\mathrm{inh}} \sum_{i \neq i'} g(v_{i'}^{\mathrm{mem}}(t)) + c_{\mathrm{m,o}} g(v_i^{\mathrm{ord}}(t)). \qquad (4)\end{aligned}$$

The first three terms in Equations (3-4) are the generic neural node equations for the ordinal nodes, $v_i^{\mathrm{ord}}(t)$, and the memory nodes, $v_i^{\mathrm{mem}}(t)$, respectively, with the resting level h and a self-excitatory term. i numbers the nodes in a sequence. The fourth term in both equations is the mutual inhibition between nodes that belong to the same sequence. The fourth term in Equation 3 is a pre-activating excitatory input from the previous memory node, $v_{i-1}^{\mathrm{mem}}(t)$, and the last term is the strong negative input from the CoS node. In Equation 4, the last term is an excitatory input from the ordinal node that activates the respective memory node.

Memory for a particular sequence is stored in connection weights from the ordinal nodes to the DFs that represent the actions. Section 3.3 specifies how these connections are realized in the action parsing architecture.

3 Architecture

We exemplify the DFT architecture of object oriented action parsing around a scenario in which a human operator reaches for colored objects on a table top, may grasp these objects, transport them to another location, and drop them there (left panel of Fig. 2). The parsing architecture continuously evaluates sensory information about the scene, which is obtained from a Kinect sensor. The goal of the system is to interpret the observed behavior by detecting and classifying actions performed by the operator, identifying the object toward which the action is oriented and memorizing the parsed action sequence in a way that the respective actions and their object representations may be activated in the same order.

The architecture consists of three modules (Fig. 2): *a*) The *sensory acquisition* module segments and classifies objects, detects motion of the hand, and estimates the motion direction. Although all of these sensory processing components have previously been established in neural dynamics architectures, here we used a simplified version of this sensory preprocessing stage that employs standard computer vision techniques. *b*) The *action parsing* module is the central component of the architecture that detects actions and stabilizes their representation along with the related target objects. Dynamic Fields process perceptual inputs obtained from the sensory acquisition module, generating a representation of the object-oriented actions in terms of peaks of activation and activation of discrete dynamical nodes. *c*) The *action memory* stores the sequence of detected actions and the objects involved in a neural dynamic representation of serial order. Next, we step through the structure and function of each of the three modules.

3.1 Sensory acquisition

The sensory acquisition module is a forward stream of sensory data processing and analysis, currently implemented through fast computer vision algorithms that provide inputs to the action parsing module in real time. In the overview Fig. 2, it is illustrated as the leftmost box. The pipeline of processing steps is shown in more detail in Fig. 3. Following along the steps in this figure from left to right, we now briefly explain the sensory preprocessing.

The data pipeline shown in Fig. 3 continuously grabs a frame of raw data, an RGB image and a depth

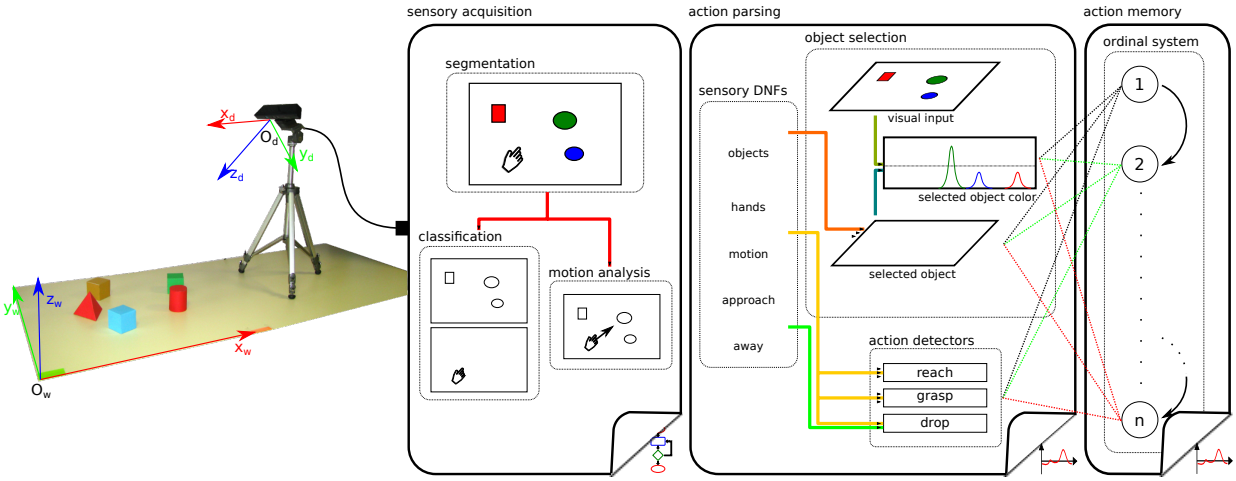

Fig. 2. Overview of the architecture. The architecture receives sensory input about the scene from a Kinect (left). This input is fed to the algorithmic sensory acquisition module (left box). The core of the architecture, the DFT based action parsing system, is illustrated in the middle box. The right box shows the DFT based action memory system.

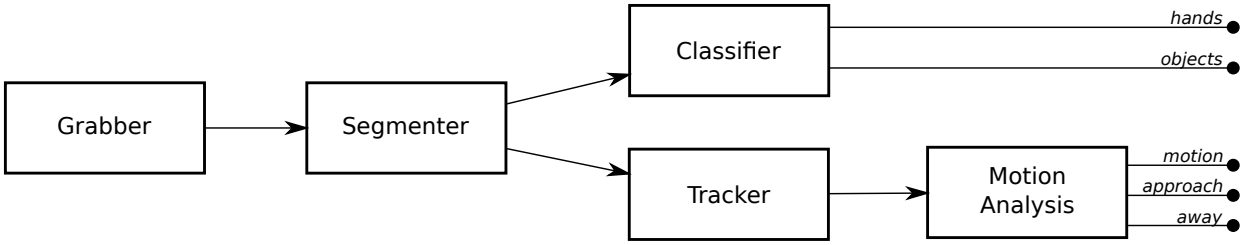

Fig. 3. Sensory acquisition module

image, from the Kinect camera. Using the three coordinates of this image, we construct a point cloud that represents the 3D scene in front of the Kinect sensor in world-based (allocentric) coordinates. To perform this transformation from image-based to world-based coordinates (anchored on a point on the table), the Kinect sensor is calibrated using two coloured markers on the surface of the table. In the current implementation, the Kinect sensor is stationary, requiring only an initial such calibration and no further recomputation of the transformation over the course of experiments. For a moving sensor, the transformation has to be recomputed, taking into account the sensor movement relative to the calibrated position. A neural-dynamic architecture for calculating such transformations in real-time has been introduced recently [15].

For subsequent steps of the analysis we use the open source 'Point Cloud Library' (PCL) [16].

3.1.1 Object detection

We segment the point cloud into points that belong to the table plane and clusters of points that lie above that plane that represent objects. The table-top plane is found using random sample consensus [17] based on the assumption that the table-top plane is the largest plane in the scene. We segment the space above the plane using Euclidean clustering, which yields a cluster of points for each object in the scene. This clustering is noisy and will be refined and stabilized by the perceptual DFs of the neural-dynamic architecture. Each cluster that contains more than a threshold number of points (here equal to 100) is classified as either an object or a hand. The binary classification hand/object derives from a color filter in the hue-saturation-value (HSV) color space. Clusters, in which skin-like colors predominate, are classified as a hand. Skin-like colors are defined by predefined ranges in HSV color space (H: $[2, 15]$, S: $[50, 100]$, V: $[50, 180]$). Clusters containing other colors are classified as objects.

3.1.2 Motion detection

Motion analysis for attentional blobs of activation can be performed in neural dynamics [18]. To simplify computation and the overall architecture, we track the 3D position of each cluster (each segmented object or hand) and estimate their velocities using a standard Kalman filter [19].

Two clusters in subsequent frames are classified as belonging to the same object when a Euclidean distance cost function remains below a threshold. This assumes that the position of an object does not change much from one frame to the next. Other possible cost functions could use a combination of other object features, such as class match or color distance. To solve the assignment problem, we use the Hungarian algorithm, also known as the Kuhn-Munkres algorithm [20, 21].

Based on the estimated velocities of the objects, we analyze the motion of all objects in the scene. For each object, we compute the norm of its velocity vector which indicates if and how fast the object is moving.

For each possible pair of a hand, A, and an object, B, in the scene, we determine whether the hand is approaching the object or moving away from it. We define a Gaussian function over the angle between the velocity vector, v_A, of the hand and the vector between the locations of A and B: $G_{AB}^{\text{appr}} = \left(\frac{\|v_A\|}{\|AB\|} + \epsilon\right) \exp^{-\frac{<v_A, AB>^2}{2\sigma^2}}$. The strength of the Gaussian is inversely proportional to the distance between the objects and proportional to the norm of v_A. The width, $\sigma = 0.3$, of the Gaussian controls how accurately v_A and AB need to be aligned to signal an approach toward the object. An analogous function is used to assess how strongly A is moving away from B: $G_{AB}^{\text{away}} = \left(\frac{\|v_A\|}{\|AB\|} + \epsilon\right) \exp^{-\frac{(\pi - <v_A, AB>)^2}{2\sigma^2}}$, where the angle is simply inverted to account for the opposite direction.

3.1.3 Output of the detectors

The outputs of the sensory acquisition system are continuous streams of values that include the color and height of all objects and of the hand, their degree of motion, and a measure of the extent to which the hand is approaching or moving away from each object (see Fig. 3). All values in the stream are defined over and bound by visual space. To reduce computational effort, we represent that stream in two dimensions defined on the table surface by projecting orthogonally from the 3D scene onto the tabletop. This projection is computed using a prior calibration that has determined the orientation of the tabletop plane relative to the camera.

3.2 The DF action parsing system

The neural dynamics of the parsing system has three components: (1) A set of sensory DFs receive sensory raw data from the sensory acquisition module and detect and stabilize representations of objects and events. (2) Attentional DFs selects the object that is the target of the currently hypothesized action. (3) The neural dynamics of activation nodes classifies sequences of detected events and objects into one of three actions, reach, grasp, or drop. We step through these three components and their interactions.

3.2.1 Sensory DFs

The sensory DFs act as detection fields and are defined over the 2D spatial dimensions of the tabletop plane. In the overview Fig. 2, they are illustrated in the second box from the left, on the left side. These fields filter noisy input data and stabilize representations of locations with salient sensory information. Peaks of activation in these DFs signal the presence of information about objects (target object or hand) or the detection of events (motion, approach, or away signals). Lateral inhibition in these fields is weak to enable multi-peak solutions. Therefore, these fields do not select the single most salient object or event but merely stabilize detected candidates.

The object detection DF receives height as an input from the sensory acquisition subsystem. Those locations in the tabletop scene at which height readings exceed a set threshold (2cm), induce peaks in the object detection DF. The locations of those peaks mark detected objects on the table.

The hand detection DF works in a similar fashion: It receives raw input from the sensory acquisition system. The detection instability in the field dynamics creates stable peaks of activation at locations in which hand color is perceived by sufficiently many samples in a cluster. Fig. 4 shows the sigmoided activation pattern in the object and hand detection fields.

The object movement DF, approach DF, and away DF all receive as input the output of the motion analysis process. In each of these fields, input is placed at the location of each object that is perceived to move, to be approached by the hand, or from which the hand

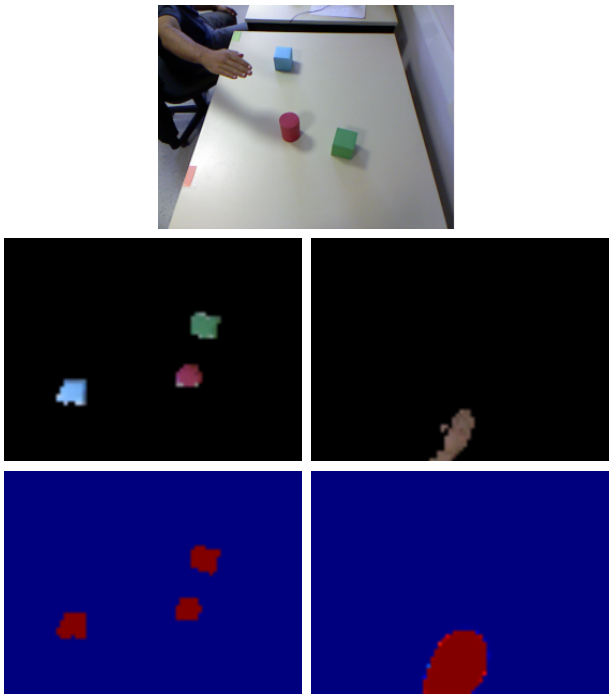

Fig. 4. Detection of hand and objects. The top row shows the camera image. The middle row shows the segmented objects (left) as well as the hand (right). The bottom row shows the activation of the object (left) and hand (right) detection DFs.

moves away. Again, inputs are effectively thresholded through the detection instability and the resulting peaks are stabilized by the DF interactions to filter out noise and subthreshold values.

3.2.2 Object selection

The hypothesized target object of an action is selected from all available objects by an attentional DF. This mechanism is illustrated in the top right corner of the middle box in Fig. 2. To enforce a selection in the attentional DF, it is set in a dynamic regime in which only one peak may be stable at any time (through global inhibition). The attentional DF receives input from the object detection DF and a combination of inputs that bias the selection of one of the available objects. Such inputs include the hand detection field that biases the selection decision toward objects close to the hand, and the approach detection field that biases the selection decision toward objects that are being approached by the hand. Thus, if the hand is moving toward an object, this object may be the target of the unfolding action. The selection of the target object is continuously updated while the human operator demonstrates actions in the scene. The initial selection may thus occur before the decision about the observed action is finalized. However, the selection process may change the selection toward an object that becomes more likely to be the target in the further course of the action. Fig. 5 shows a sequence, in which an actor moves his hand from one object to another and thus induces a switch of the object that is selected by the attention mechanism.

3.2.3 Action detectors

The DF action detectors are the central part of our architecture. In this work, we have implemented three action detectors: for the reach, grasp, and drop actions. Fig. 1 illustrates how the dynamics of an exemplary action detector is organized. At the core of each action detector are three dynamical nodes: the condition of initiation (CoI) node, the condition of satisfaction (CoS) node, and the condition of failure (CoF) node.

The CoI node receives input from one of the sensory DFs that signals that an action is probably being executed. An active CoI node thus represents a hypothesis about the currently observed action. In particular, a hand approaching an object (activation in the approach DF) activates CoIs of the detectors for 'reach' and 'grasp', whereas a hand moving away from an object activates CoI of the 'drop' detector.

The CoI is kept active by self-excitation of the CoI node until it is inhibited by either the CoS node, which detects successful accomplishment of the action or the CoF node, which signals that the hypothesis about the current action should be withdrawn. Stopping on top of the object activates the CoI for the 'reach' action and touching the object activates the CoI for the 'grasp' action. When that hand stops before reaching the object, the CoF for the 'reach' detector is activated.

Each action detector receives inputs to its constituting nodes—CoI, CoS, and CoF—from different sensory DFs. The connectivity among these nodes is identical for all detectors and defines the general processing structure that autonomously parses observed actions. Activation of the CoS node triggers a learning process in the serial order system, in which the action class and the features of the target object are stored. At the same time, the CoS node triggers a transition to the next action in the serial order system, enabling it to detect and store the next upcoming action.

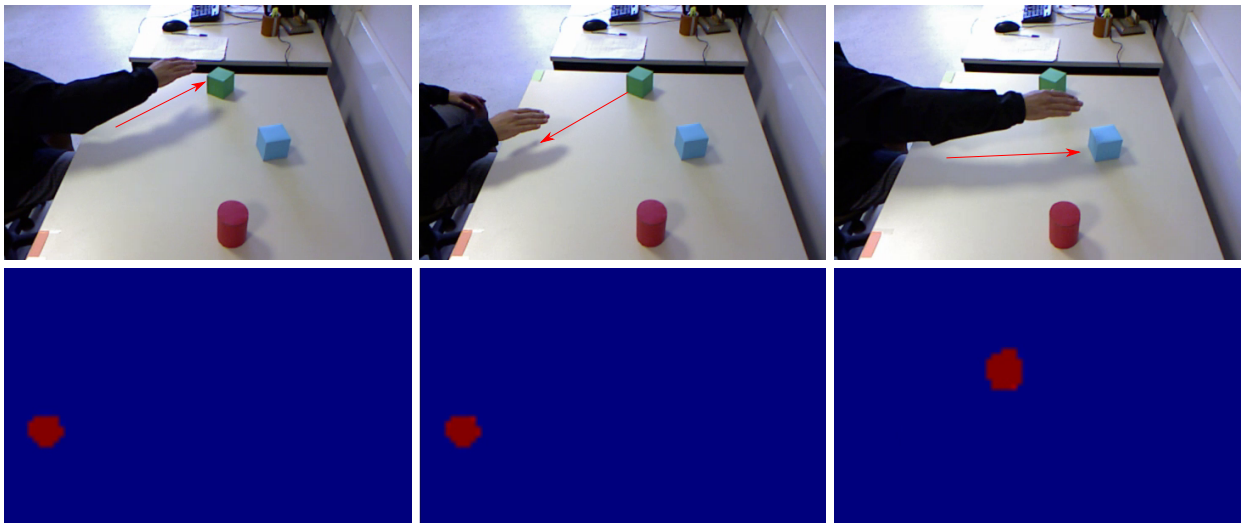

Fig. 5. Object selection: Movement toward the green object (left column) triggers the selection of this object as the target. Movement away from the green object (center column) also triggers the selection of this object. Movement toward the blue object (right column) triggers selection of the blue object as target.

3.3 Action memory

As described in Section 2.4, we use a neural dynamic architecture to represent serial order and to memorize an observed sequence of actions. Each ordinal node within the ordinal set may potentially be coupled to any of the action or target object representations within DFs by a set of adaptive weights. These weights are strengthened for the currently detected action (which CoI is activated), but only when the CoS of this action is active at the same time. This provides for a temporally limited learning window, when both nodes are activated (remember that the active CoS node inhibits the respective CoI node).

During the learning window, the connection weights are strengthened from the active ordinal node to the active CoI node as well as to the object DFs (color or space). The learning rule is similar to the memory trace formation equation used in DFT as an elementary form of learning [6]:

$$\tau \dot{W}_{i,a}(t) = f(v_{\mathrm{CoS},a}(t))\, f(v_i(t)) \Big(-W_{i,a}(t) + f(v_a(t)) \Big), \tag{5}$$

$$\tau \dot{W}^{\mathrm{o}}_i(x,t) = \sum_a f(v_{\mathrm{CoS},a}(t))\, f(v_i(t)) \Big(-W^{\mathrm{o}}_i(x,t) + f(u^{\mathrm{o}}(x,t)) \Big). \tag{6}$$

The weight $W_{i,a}(t)$ connects the i^{th} ordinal node to the action detector a, where $a \in \{\text{reach, grasp, drop}\}$. The weight grows (in a bounded way), if both the ordinal node, $v_i(t)$, and the CoI node of the action detector, $v_x(t)$, have positive activation. The shunting term $f(v_{\mathrm{CoS},a}(t))$ limits learning to time intervals when the CoS signal is emitted. The weights, $W^{\mathrm{o}}_i(x,t)$, connect the ordinal nodes with the object color field, $u^{\mathrm{o}}(x,t)$. Those weights are strengthened that connect the active ordinal node to an active region in this DF.

Once the whole sequence has been observed, the weights from each ordinal node point to one of the action detectors and a region in the object color field. Each ordinal node together with the associated connection weights thereby represent the observed action and the color of the object that action was directed at. The ordinal set may be reactivated and will generate the observed sequence of actions (not investigated in this paper, but see [7]).

4 Results

The architecture is essentially a large structured dynamical system that receives input from its Kinect sensor. We realized the process of object-oriented action parsing by solving the set of integro-differential equations on a digital computer while feeding the real-time sensory input into the equations. Here we first demonstrate the overall functionality of the architecture in a commented run of this kind. We then highlight a few of its core properties in other demonstration runs.

4.1 Autonomous parsing of an object-oriented action sequence

In the demonstration shown in Figures 6 and 7, the system observes an action sequence in which a human operator performs each of the actions that are in the architecture's repertoire, that is, reaching, grasping, and dropping.

As the scene unfolds, the architecture autonomously parses the continuous stream of sensory inputs into discrete actions and stores them in memory. Key events that emerge from these processes are illustrated in Fig. 6 for the activation nodes and feature fields and in Fig. 7 for the memory nodes and fields. In Fig. 6, the column on the left shows the evolution in time of the activation levels of the nodes associated with each action (first three rows). Plotted are the activation values (dashed lines) and sigmoided activation values (solid lines) for the CoI (blue), CoF (red), and CoS (green) nodes. On the bottom, the sigmoided activation level in the object color field is shown using a color code (blue below threshold, dark red above threshold). The center column shows snapshots of the visible scene at the moments in time marked by vertical bars numbered 1 through 7. The rightmost column shows the two-dimensional object selection field at these same points in time. In Fig. 7, the column on the left shows the evolution of the action condition of satisfaction nodes (CoS), the memory nodes (top two panels), and the color memory field (bottom panel) in time. The numbered moments in time and the associated snapshots depicted on the right are the same as in Fig. 6.

We now go through these numbered moments in time and explain what is shown in Figures 6 and 7. (1) Initially, the hand is not moving. The green object is still selected from a previous action. This can be seen from the object selection field shown on the top right. (2) When the hand starts a reaching action toward the red object, the conditions of initiation of 'reach' and 'grasp' are activated (Fig. 6, top two panels on the left, look for CoI, dashed blue). This activation pattern represents the hypotheses that the currently observed action is either a reach or a grasp. Note that the object selection field and the object color field switch to the red object, toward which the hand is moving. (3) When the hand has passed over the red object and then stops above the green object, the system has switched back to the green object as seen from the object selection field on the right. The color field at the bottom shows that the switch from red (low end of the scale near 0) to green (near 13) occurs between the events 2 and 3, just as the hand passes over the red and moves toward the green. Once the hand stops above the green object, the condition of satisfaction (CoS) for the reaching action is fulfilled (green trace in the top panel) and the system stores connections to the 'reach' action detector in memory (middle panel in Fig. 7 at event number 3) together with the connections to the green color value in the object DF (bottom panel of Fig. 7). (4) The hand touches the green object. As a result, the CoS for the grasp action is activated (red lines in top panel of Fig. 7 at event number 4). The grasp action is then stored in memory (green line in the middle panel of Fig. 7) together with the associated color value (bottom panel of Fig. 7). Note that the condition of initiation for grasp is deactivated as this happens (blue lines in middle panel of Fig. 6). (5) The hand pulls the green object to a new location. (6) When the hand lets go of the object, the CoI for the drop action is activated (blue lines in third panel of Fig. 6). (7) When the corresponding CoS of drop is activated (green line in same panel), the hand begins moving away from the green object. At this point the drop action is stored in memory (red line in middle panel of Fig. 7) together with the color value (bottom panel of Fig. 7).

4.2 Successful detection of an action: Condition of Satisfaction (CoS)

The hypothesis about the type of action that is currently being observed can either be true or false. We now show in detail how the architecture confirms such a hypothesis. We use a reaching action as an example. Other actions are detected analogously. Fig. 8 shows time courses (left column) of all dynamic elements that make up an action detector together with a sequence of snapshots (right column) taken from the camera stream at key moments. Plotted are activation values (dashed lines) and sigmoided activation values (solid lines) for the CoI, CoF, and CoS nodes of the 'reach' detector. (1) Initially, the hand is not yet moving and all nodes are below the threshold. (2) As the hand is approaching the green object, the condition of initiation (CoI) for 'reach' is activated. This represents the hypothesis that the human operator is reaching for the green object. (3) When the hand stops on top of the green object, the condition of satisfaction (CoS) of the reach action is activated. Thus, the hypothesis is confirmed and the system signals that it has detected a reaching action for the green object. This activates the reach node in the memory subsystem as well as the associated features of

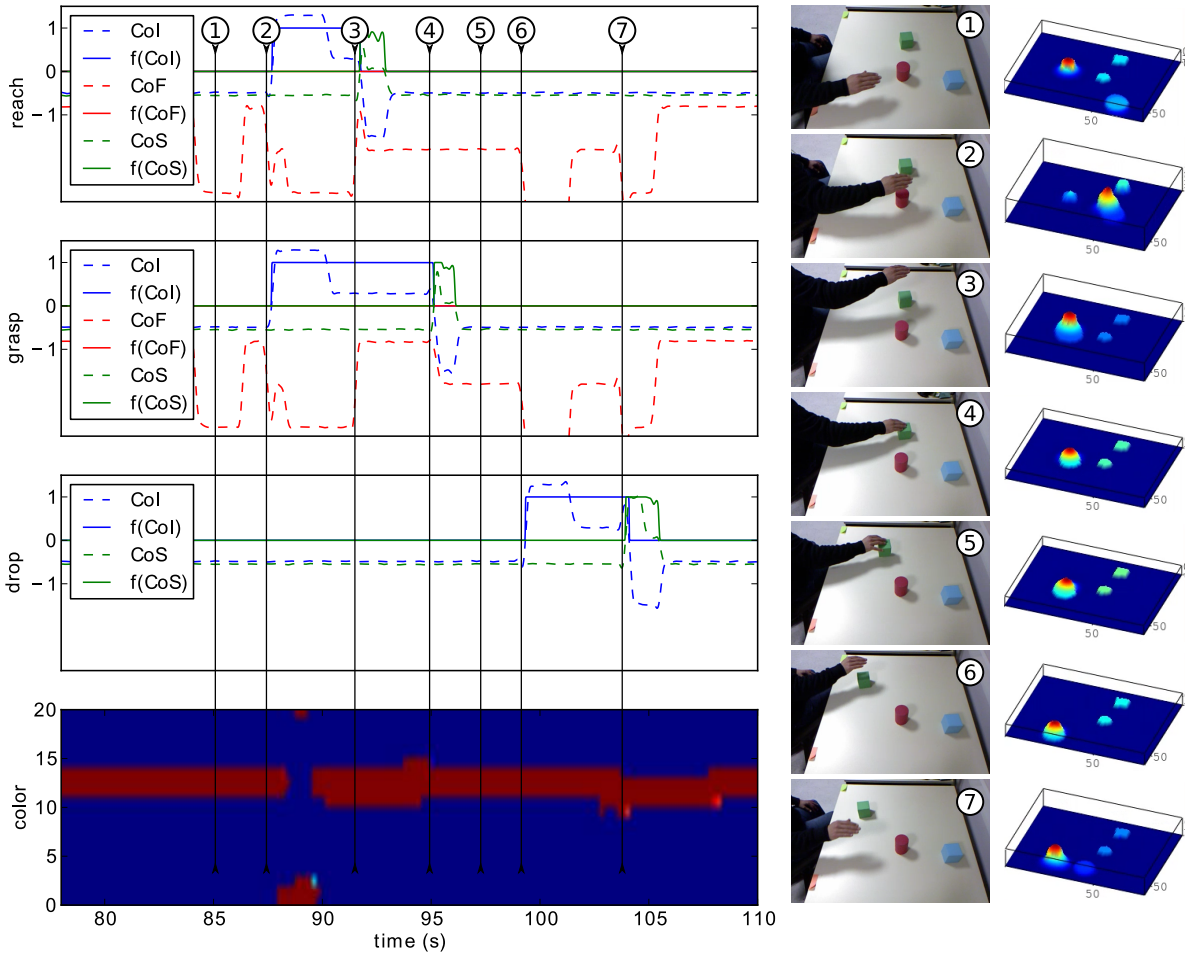

Fig. 6. Time courses and activation snapshots of the architecture for a demonstration in which an actor moves its hand above a red object, stops on a green object, and transports it away. The activation in the DF on the bottom left is color coded with blue denoting subthreshold and red denoting suprathreshold activation. See text for details.

the object (i.e., the color green). The CoI is inhibited by the CoS.

4.3 Failure to detect an action: Condition of Failure (CoF)

How does the architecture reject a false hypothesis about an observed action? Again, we use a reaching action as an example. Fig. 9 shows time courses similar to those before but now in a scenario in which the reaching action is aborted halfway. (1-2) The first two events are similar to the same events in the previous demonstration of Fig. 8: the hand moves toward the green object. As before, this establishes the hypothesis that the observed action is a reach toward that object. (3) As the hand stops before actually reaching the green object, the condition of failure (CoF) is activated. This inhibits the CoI and resets it to its initial state. (4) Once all nodes have relaxed, the system is ready to detect new actions.

Fig. 10 shows time courses of a similar demonstration for a grasping action. (1-3) The reach for the green object is successfully detected and the action stored

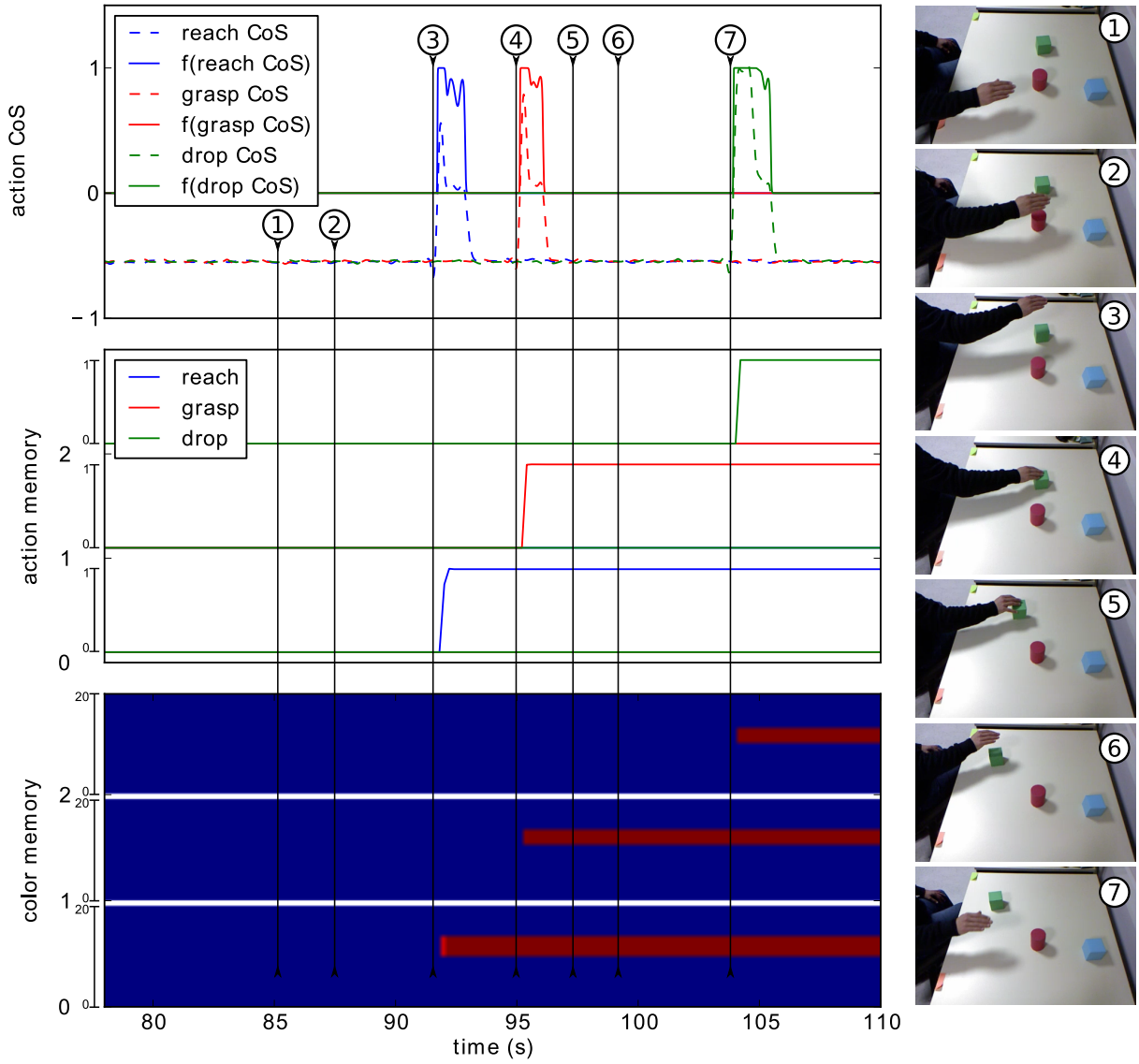

Fig. 7. Time courses of the action detectors (top left) and contents of the action memory system (middle and bottom left) for the same demonstration as shown in Fig. 6. See text for details.

across these first three events in time. At the third event, the architecture establishes a hypothesis that it is observing a grasping action of the green object. (4) When, however, the hand begins to move away from the object instead and then stops far away from the object, the condition of failure (CoF) of the grasp detector is activated, leading to the rejection of the grasp hypothesis.

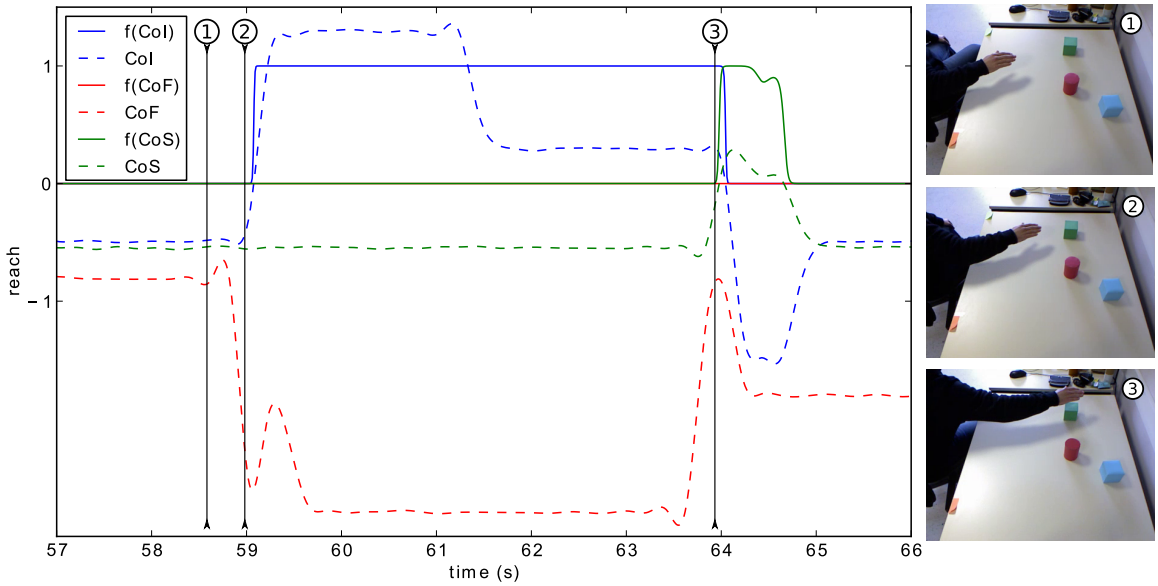

Fig. 8. Detection of a successful reaching action: condition of satisfaction (CoS).

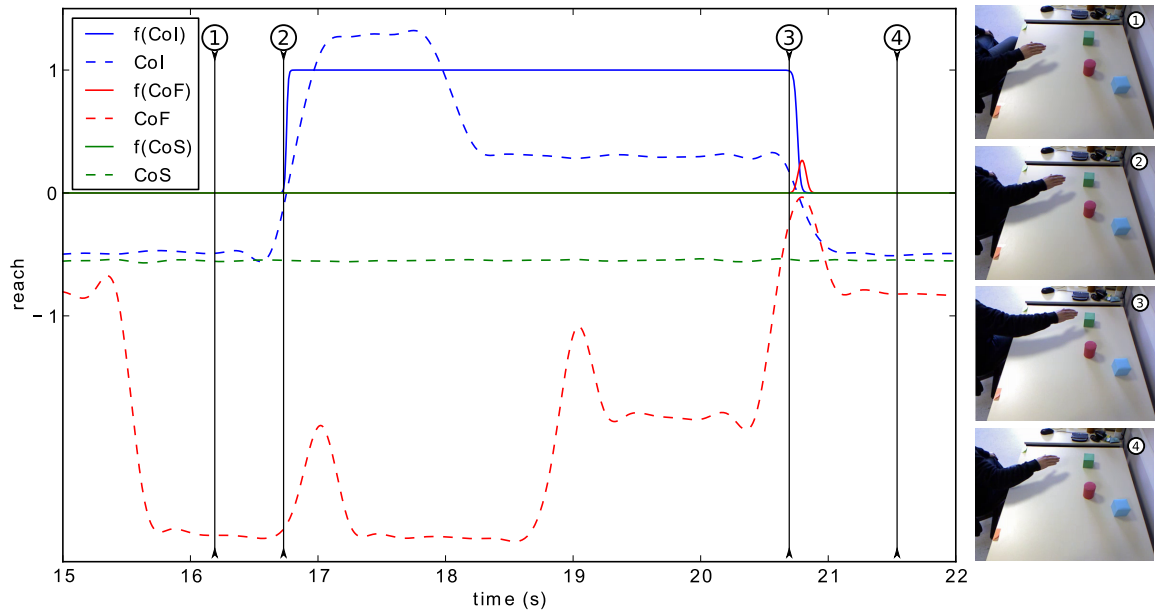

Fig. 9. Detection of a failed reaching action: condition of failure (CoF).

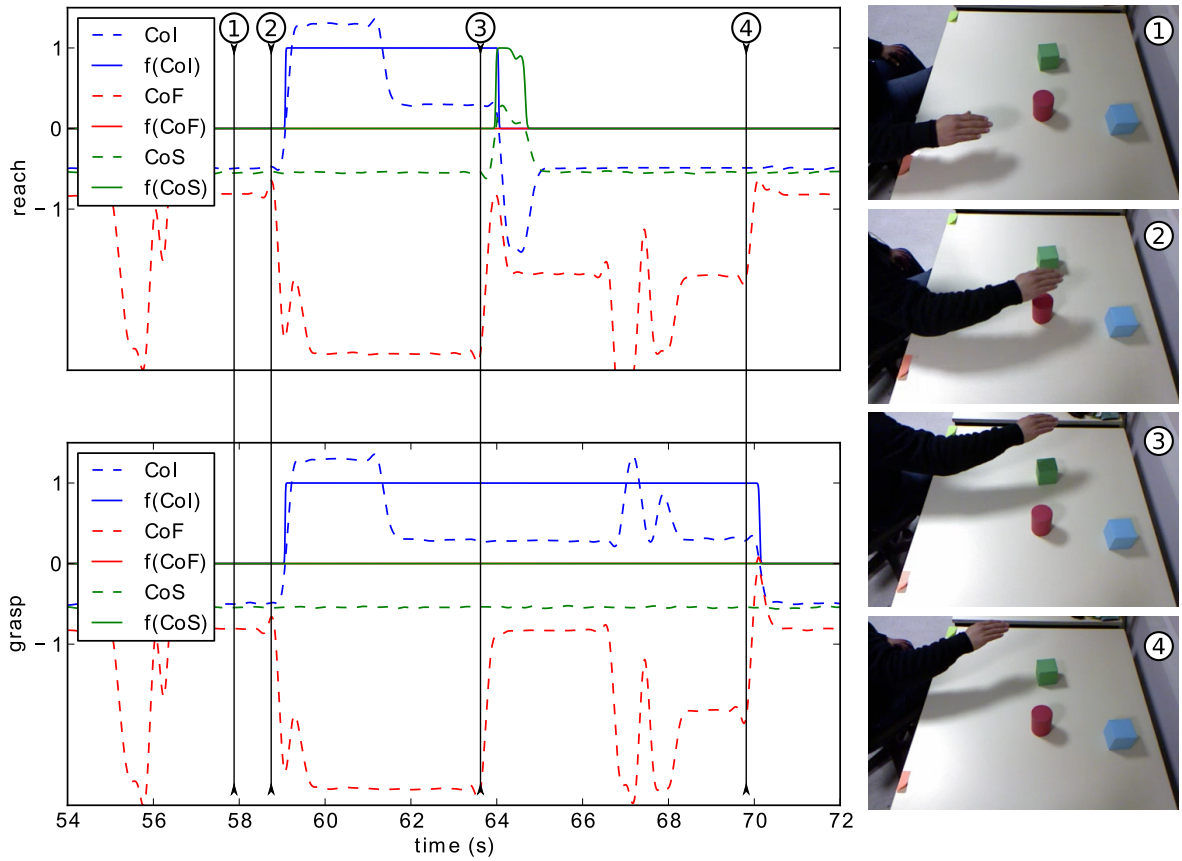

Fig. 10. Detection of a failed grasping action.

4.4 Updating the objects of the parsed action

Finally, we highlight how the system selects an object when it establishes a hypothesis about an observed action (see Fig. 11). (1) Initially, the hand does not move and the green object is selected from a previous action. (2) When the hand starts to move toward the red object, the CoI for 'reach' is triggered while the attention switches to the red object (right panel, second row). (3) When the hand begins to move away from the red object and starts to move toward the green object, attention switches back to the green object. (4) When the hand stops on top of the green object, this triggers the CoS of the reach action. The initial hypothesis 'reaching for the red object' is updated to 'reaching for the green object' before it is confirmed and stored in memory.

5 Discussion

In this paper, we have presented a proof-of-concept neural-dynamic architecture for action parsing based on dynamic fields (DFs). Our choice of the DFT framework, on the one hand, allows us to use the principles for building cognitive architectures coupled to a sensory system, developed in this framework [5, 6, 10, 14]. On the other hand, we have extended the principles of DFT for detecting and representing events (actions), which are extended in time.

In particular, the DFs in our architecture serve the following functions. First, DFs create stabilized memory representations of transient perceptual states or events. This enables the higher-level action detectors to use information that is collected at different points in time and conserved until the final decision may be made

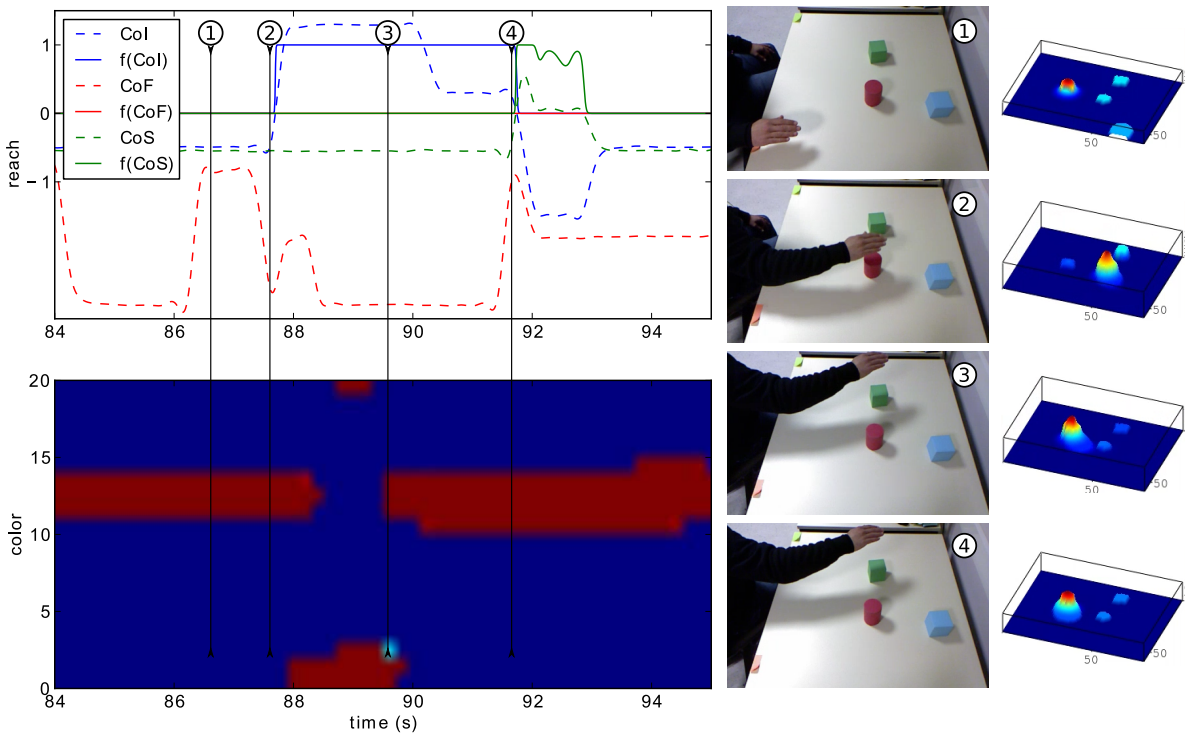

Fig. 11. Continuous update and reselection of the hypothesized target object of an action.

about the identity of the observed action. Second, the continuous dimensions of DFs allow to keep graded information about the features of the target object as well as its location and spatial arrangement in memory. In the demonstrations presented here, only the color of the target object is used as the parameter of the actions. However, we are currently working on incorporating more complex features for object representation [22].

The parsing architecture presented here adds the following functionality to DFT: first, we extend the intentional structure, introduced in DFT in the context of behavioral organization for action generation [9], to account for autonomy of action perception. Here, it is critical that actions perceived by the agent are not instantaneous but have a temporal structure. This includes the moment when the action is perceived to be initiated (CoI), is unfolding, and is perceived to be finished (CoS). In the parsing architecture, the pairs of CoI and CoS nodes accomplish detection of such extended events. Detection of the successful accomplishment of the observed action triggers the learning process, which stores the action and its parameters in memory. The condition of failure node detects when the hypothesized action is not perceived and thus is withdrawn. The proof-of-concept architecture presented here can be extended to encompass a larger action repertoire, based on our work on behavioral organization and hierarchical serial order [9, 23, 24]

Other approaches to action parsing often lack the capability to autonomously detect and represent critical events from the sensory flow. For instance, Lee and Demiris use low-level action detectors to analyze movements of the hand relative to the object, as well as the presence of objects, and the distance between the hand and the object [25]. The output of these detectors is analyzed with an HMM and context-free grammars, each representing different actions, known to the agent. Although it is hard to compare our proof-of-concept architecture with this elaborate system, our action parsing architecture features stability of the representation of the detected events and an online decision making mechanism, which leads to increased autonomy. Indeed, in order to construct an HMM, the system by Lee and Demiris needs an external signal to decide which obser-

vations are valid. The parsed actions may only be interpreted in the context of the whole sequence. In the architecture presented here, on the other hand, the outputs of the detectors are stored in a neural-dynamic memory of the system and the hypotheses about the action identities are confirmed or rejected on the fly as the behavior unfolds. In that regard, our architecture differs from other action parsing architectures that focus on movement segmentation and do not tackle the problem of parsing actions in terms of their intentional structure and target objects (e.g., [26]). Thus, not only the label for the action is stored in our system in the serial order memory, the features of the target object, at which the action is directed, are also selected and stored autonomously. The architecture is flexible and our demonstrations show how an initial hypothesis about the currently observed action and its target object may be changed.

This architecture is the first step toward a fully-fledged action parsing architecture, which should include a much richer vocabulary of available actions [1]. For each new action in the agent's repertoire, its action detector needs to be coupled to the sensorimotor system. Understanding how this coupling can be achieved in a learning process and thus how action detectors may be organized autonomously is part of our research program [27]. We are currently also working on exchanging the computer vision preprocessing stages of our architecture with neural-dynamics versions, in particular, incorporating a neural-dynamic model for the motion analysis [18].

An obvious application of the action parsing architecture is imitation learning [28, 29]. In imitation learning, the representation of the parsed sequence needs to be coupled to the agent's own sensorimotor system so that the observed sequence may be reproduced. In the architecture presented here, the observed sequence is stored in connections from the ordinal nodes to the intention nodes of the action detectors and the DF, which represents colors of the target object. During sequence replay, the ordinal nodes may be activated in a sequence and, through the learned connections, activate the intention nodes and the target field. Linking both of these structures to the sensorimotor system of the agent has been demonstrated in our previous work [8, 9] and consequently, the stored sequence may be replayed by the agent, leading to action imitation based on visual observation.

References

[1] Gutemberg Guerra-Filho and Yiannis Aloimonos. The syntax of human actions and interactions. *Journal of Neurolinguistics*, 25(5):500–514, 2012.

[2] Yiannis Aloimonos. Sensory grammars for sensor networks. *Journal of Ambient Intelligence and Smart Environments*, 1(1):15–21, January 2009.

[3] Florentin Wörgötter, Eren Erdal Aksoy, Norbert Krüger, Justus Piater, Ales Ude, and Minija Tamosiunaite. A simple ontology of manipulation actions based on hand-object relations. *IEEE Transactions on Autonomous Mental Development*, 5(2):117–134, june 2013.

[4] Georg Layher, Martin A Giese, and Heiko Neumann. Learning representations of animated motion sequences—A neural model. *Topics in Cognitive Science*, 6(1):170–182, January 2014.

[5] Gregor Schöner. Dynamical systems approaches to cognition. In Ron Sun, editor, *Cambridge Handbook of Computational Cognitive Modeling*, pages 101–126, Cambridge, UK, 2008. Cambridge University Press.

[6] Yulia Sandamirskaya. Dynamic neural fields as a step towards cognitive neuromorphic architectures. *Frontiers in Neuroscience*, 7:276, 2013.

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

[8] Yulia Sandamirskaya and Gregor Schöner. Serial order in an acting system: a multidimensional dynamic neural fields implementation. In *9th IEEE International Conference on Development and Learning (ICDL 2010)*, 2010.

[9] Mathis Richter, Yulia Sandamirskaya, and Gregor Schöner. A robotic architecture for action selection and behavioral organization inspired by human cognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, pages 2457–2464, 2012.

[10] Yulia Sandamirskaya, Stephan K.U. Zibner, Sebastian Schneegans, and Gregor Schöner. Using dynamic field theory to extend the embodiment stance toward higher cognition. *New Ideas in Psychology*, 31(3):322 – 339, 2013.

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

[12] Hugh R Wilson and Jack D Cowan. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13(2):55–80, 1973.

[13] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[14] Stephan K U Zibner, Christian Faubel, Ioannis Iossifidis, and Gregor Schöner. Dynamic Neural Fields as Building Blocks for a Cortex-Inspired Architecture of Robotic Scene Representation. *IEEE Transactions on Autonomous Mental Development*, 3(1):74–91, 2011.

[15] Y Sandamirskaya and T Storck. Neural-dynamic architecture for looking: Shift from visual to motor target representation for memory saccade. In *IEEE International Conference on Development and Learning and on Epigenetic Robotics*

(ICDL EPIROB 2014), 2014.

[16] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA 2011)*, pages 1–4, May 2011.

[17] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[18] Michael Berger, Christian Faubel, Joseph Norman, Howard Hock, and Gregor Schöner. The Counter-Change Model of Motion Perception : An Account Based on Dynamic Field Theory The Dynamic Field Model of Counter-Change Motion. *Lecture Notes in Computer Science*, 7552:579–586, 2012.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineers - Journal of Basic Engineering*, 82(1):35–45, 1960.

[20] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[21] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial & Applied Mathematics*, 5(1):32–38, 1957.

[22] Christian Faubel and Gregor Schöner. Learning to recognize objects on the fly: a neurally based dynamic field approach. *Neural Networks*, 21(4):562–576, 2008.

[23] Yulia Sandamirskaya, Mathis Richter, and Gregor Schöner. A neural-dynamic architecture for behavioral organization of an embodied agent. In *IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL EPIROB 2011)*, pages 1–7, 2011.

[24] Boris Duran and Yulia Sandamirskaya. Neural dynamics of hierarchically organized sequences: A robotic implementation. In *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 357–362, 2012.

[25] Kyuhwa Lee and Yiannis Demiris. Towards incremental learning of task-dependent action sequences using probabilistic parsing. *IEEE International Conference of Development and Learning (ICDL)*, 2011.

[26] Falk Fleischer, Vittorio Caggiano, Peter Thier, and Martin a Giese. Physiologically inspired model for the visual recognition of transitive hand actions. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 33(15):6563–80, April 2013.

[27] Sohrob Kazerounian, Matthew Luciw, Mathis Richter, and Yulia Sandamirskaya. Autonomous reinforcement of behavioral sequences in neural dynamics. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.

[28] Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009.

[29] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.