A Neuro-Dynamic Architecture for Autonomous Visual Scene Representation

Dissertation zur Erlangung des Grades eines Doktor-Ingenieurs
der Fakultät für Elektrotechnik und Informationstechnik
an der Ruhr-Universität Bochum

vorgelegt von
Stephan Klaus Ulrich Zibner
geboren in Hattingen

Bochum, Oktober 2015

Kurzfassung

Unsere Fähigkeit mit Gegenständen in unserer Umgebung zu interagieren ist einzigartig. Eine Grundlage dafür ist die visuelle Perzeption von Szenen, welche zur Generierung interner Repräsentationen führt, auf denen nachfolgende Verhalten aufbauen. Greifbewegungen, sowie Sprachgenerierung und -verständnis sind Beispiele für solche Verhalten. In der Robotik stellt die visuelle Verarbeitung von Szenen eine große Hürde dar, insbesondere für a priori unbekannte oder dynamische Szenen. In dieser Arbeit stelle ich eine neuro-dynamische Architektur zur visuellen Verarbeitung von Szenen vor, welche ein Arbeitsgedächtnis aufbaut, dieses bei Veränderungen der Szene aktualisiert und das akkumulierte Wissen über die Szene nutzt, um die Suche nach einem Zielobjekt effizient durchzuführen. Kern der Repräsentation sind dreidimensionale neuronale Felder, welche die Position von Objekten mit deren visuellen Eigenschaften wie ihrer Farbe oder Größe assoziieren. Ich lege dabei meinen Fokus auf die Organisation der involvierten Verhalten und zeige, wie die internen Prozesse autonom ablaufen. In Experimenten auf einer Roboterplattform evaluiere ich die von der Architektur generierten Verhalten und Prozesse und ziehe Parallelen zu Erkenntnissen über die menschliche Szenenrepräsentation. Ich stelle zwei Erweiterung der Szenenrepräsentation vor, in denen ich die zugrunde liegenden Prinzipien nutze, um Objekterkennung in die Szenenrepräsentation zu integrieren und die Szenenrepräsentation nutze, um auf Objekte gerichtete Armbewegungen zu steuern. Die Verbindung mit Objekterkennung erlaubt es, Zielobjekte anhand abstrakter Label zu suchen. Da die Objekterkennung rechenaufwändig ist, ist die Erzeugung der Label aus der Szene nicht als paralleler Prozess der visuellen Wahrnehmung gestaltbar, sondern nutzt die entwickelte sequentielle Bearbeitung von visuellen Objekten. Für die Bewegungsgenerierung leistet die Architektur, dass Bewegungsverhalten sich jederzeit an Positionsveränderungen des Ziels anpassen können, gewährleistet durch die kontinuierliche Anbindung der Szenenrepräsentation an die visuelle Wahrnehmung und durch die autonome Organisation der Prozesse der kognitiven visuellen Wahrnehmung. Solches “online updating” ist auch beim Menschen bekannt. Ich schließe mit einem Ausblick auf weiterführende Integration im Kontext von Greifbewegungen.

Abstract

Humans have a unique ability to interact with objects in their vicinity. Foundation of these interactions is the visual perception of scenes, from which internal representations are created. Behaviors such as reaching and grasping, as well as generation and understanding of utterances, build on these representations. Processing of visual scenes is a major challenge for robotics research, especially if scenes are novel or dynamic. In this thesis, I present a neuro-dynamic scene representation architecture. It creates working memory representations of scenes, updates memory content on change, and is able to re-instantiate accumulated knowledge about the scene to efficiently search for target objects. At the core of the architecture, three-dimensional dynamic fields associate the spatial position of objects with their visual features such as color or size. The main focus of my work is the behavioral organization of involved behaviors and the resulting autonomy of processes. I evaluate the behaviors and processes generated by this architecture on robotic platforms and compare the evaluation with behavioral signatures of human scene representation. I extend the principles of scene representation onto two applications: object recognition and movement generation. The integration with object recognition allows to locate target objects by means of abstract labels, whose generation is computationally demanding and thus cannot be applied in parallel to a visual scene. Instead, these labels are memorized in a sequential process provided by the scene representation architecture. Movement generation benefits from the continuous link to visual input and autonomous organization of behaviors. Changes in target position are continuously integrated into the current movement. This property of on-line updating is also found in human arm movements. I conclude with a perspective on advanced integrative work in the context of robotic grasping.

Contents

Chapter 1

Introduction and Motivation

While writing this thesis, I sit in front of my desk. I am aware of several objects in my direct vicinity. A steel coffee cup is placed to the left of the keyboard, as is my red note book, a pen, and my mobile phone. Some properties of these objects are directly available to me, without requiring me to move my gaze away from the screen while writing this text. Answering detailed questions, for example naming the brand of the pen, requires me to pay attention to a specific object, direct my gaze at it, or even pick the object up and turn it around to find the brand logo. All of these operations require little to no effort from me and are so basic for everyday interaction with objects that the complexity of brain processes realizing these operations is easily underestimated.

The common theme underlying these interactions is called scene representation. The layout of objects in my vicinity entered my mind at some point in time, leaving some traces that I can use to guide my attention back to specific objects. These traces allow me to remember certain properties of these objects, but are not complete in a sense that I have a full, detailed, pictorial representation in my mind [72, 135]. They also allow me to evaluate statements of spatial relationships between these objects (e.g., "the cup is to the left of the keyboard") and perform actions such as grasping movements directed at objects in the scene.

My brain uses an intricate machinery for these processes. This machinery takes care of attentionally highlighting objects in a sequential fashion (which is tightly connected to eye movements [71]), creating traces of object characteristics in short- and long-term memory [76], updating these traces if change occurs, and re-instantiating them if other behaviors require these details. These processes require no conscious intervention or control; they are, in a sense, autonomous. In fact, I have no direct insight into the state of representation, as my environment appears to be fully accessible to my

perception at all times. This illusion of complete representation can easily be dissolved in psychophysical experiments by manipulating[1] the unconscious processes, for example by restricting eye movements [76], increasing the amount of objects and the level of detail to be perceived and memorized [106], introducing movement [132] and masking [137], and reducing the time in which an internal representation may be created [184]. Representations are thus an essential component of human scene perception, which operates sequentially to create, update, and read out these representations.

In robotics research, visual perception is a recurring topic of interest with visual sensors delivering a rich portrayal of the surroundings of a robot. Representations of visual perception are often understood as complete, detailed world models (see, for example, work on mapping using a mobile robot [100, 174]). Creation of world models is supported by both special-purpose sensors such as laser range finders for depth measurements and an ever-growing amount of processing power and memory capacity of contemporary computers. However, managing this abundance of information is a central challenge in robotics [11]. This is evident in the limited ability of robots to interact with naturalistic scenes, an ability acquired early in human childhood and which we use effortlessly in our daily lives. A trend over the last decade of robotics research is to incorporate principles of primate vision into robotics to endow robots with comparable skills [11, 15]. This involves using low-dimensional features as description of objects inspired by findings in behavioral studies on locating objects [191], as well as using a parallel filtering operation evaluating the *saliency* of visual regions [84] to highlight possible candidates of interest. The result is used to sequentially guide a blob-shaped window of *attention* that is the gateway to further processing, creation of memory representations, maintenance of memory, and re-instantiation of contained information. Implementation of said principles happens in frameworks that are inspired by neural processes in the primate brain.

Advances in this field of research expand on neurally inspired models covering aspects and processes of scene representation, such as saliency processing and attentional selection (reviewed in [15, 83]) and cue-guided visual search [62, 191]. These models process sensory input and do not create or use an internal representation of the scene. This leads to two open research questions: (1) What is the role of internal representations in scene perception? (2) How are the sequences that create, maintain, and query the internal representations generated by the neural substrate used to model the processes?

[1] The references in the text following this footnote point to exemplary behavioral studies showing effects of these manipulations.

The latter question is part of the more general problem of how sequences of behavior are generated by neural substrate. This is a general challenge for neurally inspired models—embedding in neural substrate the mechanisms that drive the same substrate to meaningful behavior, without resorting to external algorithmic control structures that are obviously not present in the primate brain. The commitment to the neural plausibility of a model requires to solve this challenge. When applied in the context of robotics, it results in behavioral *autonomy*, that is, endowing a robotic agent with an intrinsic drive that generates sequences of behaviors and adapts to the time-varying sensory input.

In this thesis, I address the two research questions mentioned above by providing an integrated account for the processes of scene representation that are essential for any interaction with scenes and objects. My focus lies on the autonomy of these processes. How is the nervous system able to systematically scan a scene and memorize certain aspects of it? How are inconsistencies between internal representation and external world detected and what are the means to resolve them? How can an object be brought into the attentional foreground given a task and how can the suitability of a candidate be judged given the task constraints? I use dynamic field theory, a neurally-inspired modeling framework, both for modeling the neural processes that create, maintain, and read out the representation of a scene, as well as for generating the sequential structure of the involved processes within the same substrate. My main goal is to show that functionality of scene representation can be obtained from consistent neural process modeling. I evaluate this functionality by comparing its behavior to behavioral signatures of human scene representation and embedding the resulting architecture on robotic agents, which autonomously interact with scenes. I demonstrate in two applications that my architecture can provide its functionality to larger architectures that build semantic maps based on object recognition and that reach for objects. Both applications are essential for robotic interactions with naturalistic scenes.

In Chapter 2, I first present the behavioral signatures of human scene perception and resulting models to define a frame for a robotic architecture of scene representation. I then discuss how the emerging behaviors may autonomously be generated and organized. Chapter 3 introduces building blocks of dynamic field theory. In Chapter 4, I assemble a robotic architecture of scene representation from these blocks, with emphasis on behavioral organization. Afterwards, I evaluate the architecture in experiments and compare it to other models. Chapter 5 contains two applications for the mechanisms of scene representation. I conclude this thesis with a general discussion of my work in Chapter 6.

Chapter 2

Background

The interdisciplinary nature of my thesis requires a broad overview of several research areas—from human behavior research to computational modeling of brain processes, from philosophical discussion of autonomy and behavior to the concrete implementation of behaviors and their organization on robotic agents. The following sections present a cross section through the relevant research areas, without a claim of exhaustiveness.

2.1 Behavioral Signatures of Human Scene Representation

The foundation of scene representation is the visual processing pathway, spanning from the retina up to the different areas of the visual cortex. Processing in visual cortex is functionally split up into two parts: the *dorsal* pathway, which is concerned with spatial processing of visual input ("where?"), and the *ventral* pathway, which deals with object identity ("what?") [112, 181].

2.1.1 Saliency and Attention

On the dorsal pathway, primate vision has to deal with two facts. One, the overwhelming amount of visually perceivable information (see [95] for a detailed analysis) and two, the fact that, in terms of resolution, visual perception is best at the fovea of the primate eye. This poses the challenge of processing visual information to determine which regions are worthwhile to be further inspected by foveating them. The outcome of this process is termed *saliency* [94] and serves the *attentional selection* of regions for

further processing. *Attention* refers to a sequential bottleneck in visual processing that is measurable as increased capacity to discriminate and detect visual features [148]. At the same time, attention is critical to scene representation because only objects that were previously attended are reliably represented (discussed in detail below). The sequential selection can happen either *covertly*, by shifting an internal spotlight of attention, or *overtly*, by executing a saccadic eye movement that centers the fovea onto the selected region. Both variants of selection may use the same underlying neural mechanism, with eye movements being actively inhibited during covert shifts [141]. Covert attention shifts are necessary even if two or more perceived stimuli are sufficiently close together to be inspected without an eye movement [153].

An insight in saliency processing of the primate brain can be extracted from studies of saccade sequences in natural scenes. Placement of saccades in human scene perception almost exclusively covers informative regions, especially if they contain objects, with the addition that saccade placement is also driven by top-down influences of ongoing tasks (see [101] for an exemplary study and [70, 72] for reviews). Attentional selection thus relies on two components: a bottom-up, intrinsic, input-driven influence, which manifests itself in pop-out effects of visually unique regions, and a multifaceted top-down influence driven by cognitive tasks such as visual search, the history of previously attended locations, and other factors.

A first computational model of bottom-up saliency was presented by Koch and Ullman [94] and was later extended by Itti, Koch, and Niebur [84]. Here, localized features (color, intensity, orientation) are extracted from a visual input and represented in maps across multiple scales for each feature channel. Using the multi-scale structure of these maps, center-surround differences are computed. This operation assigns a uniqueness measure to each location, which depends on the difference of a center region to its surrounding region. A subsequent normalization operation favors maps with sparse high local uniqueness entries, while maps with multiple regions of high local uniqueness are suppressed. The resulting activation is represented in *feature maps*. Feature maps are then combined along each feature channel into *conspicuity maps*, with another normalization operation to favor feature channels with sparse uniqueness. Finally, conspicuity maps are combined into a saliency map, which is the basis for attentional selection using a winner-takes-all mechanism [84]. Attentional selection receives additional influence from the recent history of fixations. This influence decreases bottom-up saliency for recently selected regions and is called *inhibition of return* (IoR), as a return to a previously inspected and thus inhibited region is less likely. Together with the repeated winner-takes all selection, a scanning pattern emerges.

The fixations resulting from this model reflect bottom-up influences and

only account for a small percentage of human fixation data [15], which inspired later models to include additional top-down influence to alter this pattern, favoring regions that match top-down cues relevant for cognitive tasks [62, 116]. Through probabilistic modeling, any kind of prior may influence the selection process, for example scene context [39]. Top-down modeling was also achieved before the rise of bottom-up saliency. In a model by Houghton and Tipper [80], a comparison of internal feature cues and external visual input produced saliency-like maps using a match-mismatch detector over the whole visual field.

One insight from top-down modeling is the use of a common feature description used for both bottom-up processing and top-down cues. This ensures compatibility and solves the grounding of higher level cognitive processes into the lower level visual processing. Navalpakkam and Itti suggest that working memory representations of objects also use the common feature description [116]. Craye and colleagues [32] argue that saliency processing develops during childhood. They present an architecture that acquires saliency processing through an intrinsically motivated exploration process. Here, attentional shifts of a foveal region are used to learn saliency, which is in contrast to using saliency to pick targets for attentional shifts in the models above.

2.1.2 Visual Search

Saliency and attentional selection together with top-down orchestration form one of the most basic behaviors of primate vision – *visual search*. Visual search is the behavior of bringing a target object into the attentional foreground of the cognitive system, be it covert or overt. A small set of feature dimensions is used to efficiently locate a target object [189].

Behavior studies (see, for example, [119, 180]) distinguish two types of search paradigms: *feature* search, in which a target object is defined uniquely by its difference to distractor objects along at least one feature dimension (for example, searching a red circle between green squares), and *conjunctive* search, with the target defined by a unique combination of two or more feature values, while distractors share some of these values (for example, searching a red circle in an array of green circles and red squares). Figure 2.1 shows examples for these paradigms. In addition, subjects may know the identity of the target object before display of the search array, either explicitly (e.g., verbal announcements, such as 'the target is the red circle') or implicitly (target stays the same over a block of search arrays).

Behavioral data suggests that the former search paradigm is realized by parallel processing of the scene, as search time stays practically constant

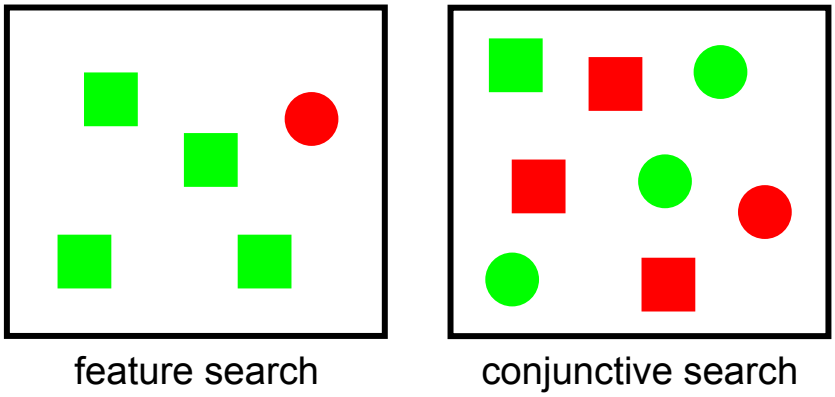

Figure 2.1: This figure shows an exemplary scene for a feature search on the left, in which the target object 'pops out'. The right array includes two objects with a unique combination of features (green square and red circle). If the target is not known in advance, only one of these objects is present in the array to unambiguously define the target.

with varying amount of distractors (termed *pop-out effect*), while the latter requires some form of sequential processing (or, in other words, explicit attention), as search time increases linearly with the amount of distractor objects. This view was made explicit in *feature-integration theory of attention* [180], which states that the *binding* of features, that is, the combination of feature values all belonging to one object, require attentional focus, whereas unbound features can be processed without using an attentional bottleneck, and thus can be analyzed in parallel for the whole visual array.

A different explanation for the gap between feature and conjunctive search has its roots in a model by Hoffman [74] who describes visual search as a two-stage process, which first applies parallel pre-processing to an input, which then drives a sequential comparison process, in which only one item can be held in the foreground at any time. Wolfe refined this view with his model of *guided search* [188, 191]. Here, a parallel processing stage extracts limited information from a visual input, which is processed by a more powerful, sequential inspection of single items highlighted by the parallel processing. This line of research has large overlap with the saliency modeling described above, as saliency is a modulatable parallel process, which then guides selective attention to regions for closer inspection.

Hamker [62] attributes the gap between feature and conjunctive search to the amount of possible candidate locations produced by the modulated saliency processing and the subsequent sequential process of evaluating each candidate for a fit. If cues bring up multiple candidates due to feature simi-

larity, visual search takes more time in comparison to search with cues that only highlight few candidates. This confirms earlier findings on the dependency of stimulus similarity defining the relationship between search time and amount of distractor objects [35].

From an embodiment viewpoint, the existence of a sequential bottleneck in visual processing follows naturally from the structure of the primate retina, with the foveal area being responsible for detailed processing of a single item, while the extra-foveal areas can be used to extract limited information about future fixations. At least in humans, this natural relation between body and cognitive functionality is broken up by the behavioral signatures of covert attention—an attentional shift that is consciously decoupled from the motor system of the eye [81, 193], while activating shared brain regions [23, 33]. On one hand, this is an advantage as visual processing is no longer tied to the fixed execution time of saccades of around 300 ms. On the other hand, it is not clear if the ability of doing covert attention comes with a price, for example, a less precise memorization of attended objects or increased reaction time in visual search. For the latter example, Nothdurft and colleagues [120] present behavioral data of macaques and humans suggesting that only using covert attention to perform visual search has a beneficial effect on the reaction time of visual search, with the restriction that the monkeys had a hard time learning to do this task with purely covert shifts in the first place.

To efficiently search for an object requires some mechanism to generate sequences of attentional fixations, given the sequential bottleneck identified above, for example by memorizing the already inspected regions and only selecting novel locations on subsequent fixations. Horowitz and Wolfe [79] evaluated search performance in an array in which objects change their position every 111 ms. Target and distractors were chosen to require a serial search (rotated letters T and L, no pop-out). They found no difference in performance in comparison to the control condition, in which the display remains static. They draw the conclusion that visual search does not have a memory for previously inspected locations, as this would lead to better search performance in the control condition.

Peterson and colleagues [126] respond to this study by recording eye movements during a visual search experiment similar to the one used by Horowitz and Wolfe. They assume that memory-less visual search frequently revisits previously inspected locations. They argue that re-inspections of objects might also happen if memory is available to visual search, as target objects may not be recognized as targets on first fixations and the eyes return on subsequent fixations to identify this object as target. Peterson and colleagues compare human performance with three models, one having no memory of previous inspections and two using perfect memory of inspection history. The

two latter ones feature varying amounts of uncertainty of target detection, which leads to objects being re-inspected after missing the detection of having selected the target. Peterson and colleagues show that the models with memory are better fits for the human data than the model without memory. The authors claim that the contradiction with results by Horowitz and Wolfe is founded in their use of a less natural search paradigm. Peterson and colleagues state that response times in flickering scenes cannot easily be compared to the ones achieved in static scenes.

Visual search performance is often tested on novel displays, that is, subjects have no previous experience with the scene and cannot rely on any representation built up previous to executing a visual search. Interactions between visual working memory and visual search are examined by Woodman and Luck [192] who conduct experiments probing the influence of a maintained representation on visual search performance. In their study, subjects are asked to keep an object with distinct color in working memory. Subjects then perform a visual search for an object with color not being relevant for the search. Distractors in visual search share the color of the object kept in working memory. The authors expect that the visual working memory representation automatically draws attention to matching distractor objects, following a theory of visual attention by Bundesen [22]. However, in their study, Woodman and Luck find that subjects are able to reduce response time if they know that the target object never is of the memorized color. In addition, subjects are able to use the color information in memory to guide attention back to the memorized object in a subsequent resampling task. The authors conclude that visual working memory has no automatic attention-grabbing effect. Its content can be used flexibly to either inhibit or highlight candidate objects.

The flexible use of working memory content is emphasized by Olivers and colleagues [124] who argue that working memory items can either be in a passive or an active state. Only active items may influence attentional selection. Kane and colleagues [88] examined whether individual differences in working memory capacity have an influence on the performance of visual search in complex search arrays requiring serial search and thus some kind of memory for previously attended locations. They found no significant difference between participant groups with low and high working memory capacity.

2.1.3 Inhibition of Return

Inhibition of return plays a crucial role in visual search [89, 90] and saliency modeling [84], as pointed out before. IoR lasts for several seconds [90] and is thus not tied to a retinal coordinate frame, but related to an environmental

coordinate frame [128], as retinal positions would map inhibition onto different, uninspected regions after a saccadic eye movement. Klein has collected other characteristics of IoR, which have to be taken into account, in a review [90]: IoR declines with distance to fixated location, which suggests a distribution rather than a precise marking. Additionally, IoR is coupled to the preservation of a search array, which implies some behavioral or cognitive control over whether IoR is retained or not (see also [186] for further evidence). Finally, IoR placement may move with previously fixated objects, which hints at a more complex source than environmental location alone.

Posner and Cohen [128] describe that inhibitory effects on visual processing are preceded by a facilitation effect for locations matching a cue. This is in alignment with top-down influences of cues on saliency, which favor locations that match a cue. Attentional selection of these locations then produces IoR, weakening the facilitation effect over time.

A study by Maylor [110] reports that the magnitude of facilitation and inhibition effects approximately halves if two cues specify two spatial locations as potential targets (instead of cuing one location only). In the bigger picture of selective attention picking salient locations, having two potential target locations with comparable saliency results in a decision at chance level, which of the two locations is inspected by covert attention and subsequently marked with IoR. A decrease in magnitude of both facilitation and inhibition thus is a logical consequence of this chance decision before the target is presented. Maylor concludes that this result is further evidence of IoR being coupled to an active orienting and not a property of sensory stimulation.

2.1.4 Representation and Working Memory

Although the visual input is highly volatile under eye movements, we perceive our surroundings as mostly stationary and invariant [135]. We know where some of the objects in our surroundings are placed, even without frequently checking them visually. We can, in principle, simply close our eyes and grasp an object on our desk. Maybe we will tip something over or it may take longer than usual, but we succeed. These simple facts point to the conclusion that what we extract from a visual fixation is not lost once our eyes move to a new location, but rather is kept in memory. But what is the nature and content of this memory? Theories cover a broad spectrum, from 'memory contains only information of the latest saccade' [8, 135, 190] to 'some information is kept' [75] up to 'each new saccade contributes to an integrated global image' [48].

Theories of limited representations are often associated with working memory being a limited and expensive resource [8]. Theories of full vir-

tual representations are problematic, both in the sense of pure storage, but also in efficient access, as discussed by Rensink [135]. There is evidence for a robust and precise representation [75], potentially using additional forms of memory [16]. In addition to the extent of coverage of the visual scene in working memory, a second angle examines the level of detail extracted at each fixation. A study by Henderson [69] shows that preserved information is not very detailed. A similar observation is made by Phillips [127] who reports a graded loss of details beginning at around 600 ms after stimulus presentation, with a preceding phase of fully detailed sensory representation. Attentionally focusing locations is a requirement of retaining the representation [6, 7]. A sequential presentation of objects leads to poorer memory precision in recall compared to a simultaneous, spatially spread out presentation of all objects in the array [57]. Gorgoraptis and colleagues find this effect for sequential presentation at the same location as well as sequences spread out in space.

To evaluate the impact of attention on visual perception and the nature of working memory, the change detection paradigm is used (reviewed in detail in [136]). The term *change blindness* describes the inability to detect a (significant) change in the visual input (reviewed in detail in [163]), be it caused by lack of attentional focus, working memory limits, poorly detailed representations, or other sources.

in change detection experiments, visual arrays are presented to participants for a varying amount of time. After initial presentation of the array, the whole array or parts of it are masked to prevent attention-catching channels such as motion detection to guide attention to the location of change. After the masking, the visual array is shown again, either with an induced change or without it. Participants are then asked to state if the visual array has changed or not, optionally stating the exact type of change. Figure 2.2 shows examples of induced changes.

Experimental findings shed some light on the internal structure of working memory. Change detection tasks can appear to be parallel (e.g., an object changed its color to one not included in the original array) or serial (e.g., two objects switch their color, while all other features remain fixed), which is comparable to visual search (see [138] for an exemplary study). Hyun and colleagues [82] argue in favor of an unlimited parallel process of change detection, but find evidence for a serial bottleneck in the reaction time of generated movements in response to change, which increases with set size. Rensink and colleagues assess that detection of change depends on previously attending the location of change [137]. Luck and Vogel find that working memory has a capacity limit of about four objects, independent of the amount of features stored per object [106]. Signatures of change detection such as a longer fixation time can be measured even without a conscious

response of participants [78, 136]. Change detection is still possible after a prolonged inter-stimulus interval of several minutes, hinting at a link to long-term memory [70]. Working memory may be corrupted by mis-bindings of features, which is a potential cause for errors in change detection (for a review of binding errors and the binding problem, see [179]). The creation and recall of representations are influenced by other factors, for example, the consistency of an object to a displayed action [162]. Dwell time correlates with recall performance, both for objects in a scene [162] and observed actions [67], indicating whether an object enters working memory.

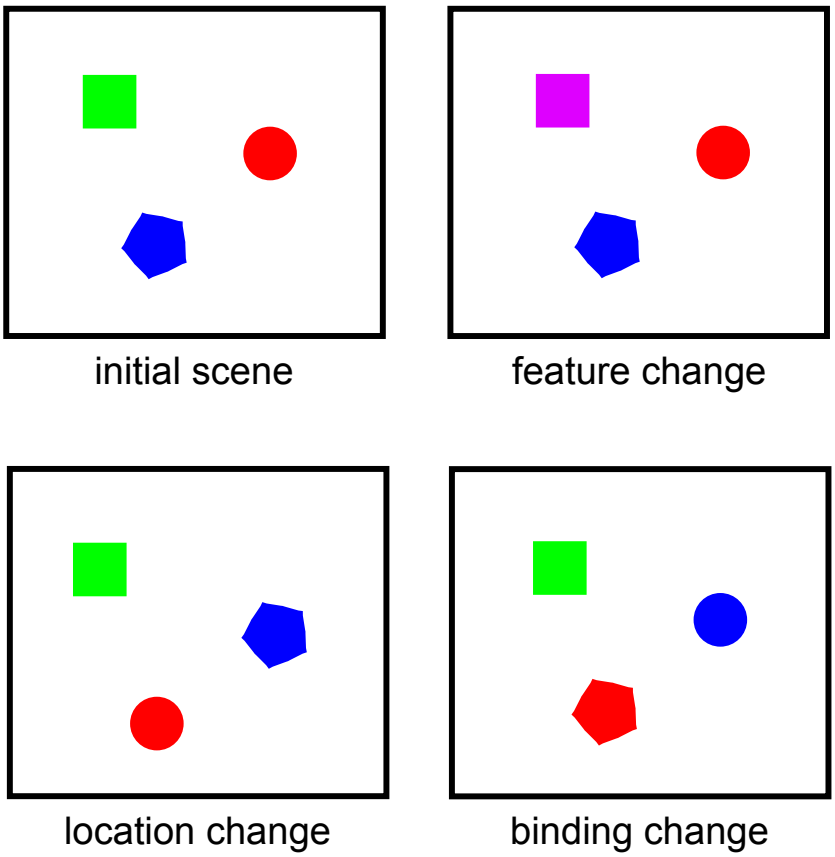

Figure 2.2: This figure shows an initial scene on the top left with three objects of distinct color and shape. Different kinds of change are present in the other arrays created by introducing a new feature value, objects switching their positions, and changing the combination of colors and shape.

Several studies examine if working memory representations are static or dynamic. Pylyshyn and Storm [132] construct displays of moving crosses, a subset of which is designated as targets of a tracking task. Subjects are asked

to respond to flashes occurring at the current location of a target object while ignoring flashes occurring at locations containing distractor objects. During the experiment, targets and distractors are visually indiscernible, which requires some form of representation marking the target objects. Subjects are capable to simultaneously track up to five objects, with error rate and reaction time increasing with number of target objects. The authors argue that there are two possible explanations for the capability of multi-item tracking. (1) An internal representation carrying along the initial distinction between targets and distractors may be updated in parallel. (2) Target objects are re-examined with a fast moving, sequential window of attention, updating the last known location of each target by executing a neighborhood search to find an object most likely being the target associated to the last known location. In a second experiment, the authors increase the distance between target objects to decrease the likelihood of a serial process being capable of executing shifts of attention that are fast enough to cover the whole array of targets. They find no significant impact on subjects' task performance, hinting at a parallel component of representation maintenance.

In a later study, Pylyshyn [131] discusses that the ability of tracking multiple target objects comes with a price. The memory associating identities to tracked target objects deteriorates over trial duration. The author observes identity swaps of pairs of objects that depend on the closeness of objects in the visual array. Identity swaps appear more likely for target-target pairs (compared to target-nontarget pairs). Cohen and colleagues [31] further investigate the tracking of multiple object identities. They find a trade-off between tracking the identity and location of target objects. Tracking identities in addition to the location of target objects decreases task performance of target localization with increasing movement speed of objects and increasing number of tracked targets. Results from experiments in which subjects may voluntarily emphasize location or identity performance lead the authors to the conclusion that location and identity tracking uses a common resource.

2.1.5 Reference Frames

Perceiving the surrounding world as stable despite highly volatile perceptual input changing with each executed eye movement implies that there is more than one reference frame for spatial positions. Feldman [48] argues that there are exactly four reference frames involved in perception. The current retinal input is represented in the *retinal frame*. A body-centered *stable feature frame* is used to align the retinal snapshots into a stable representation. The *environmental frame* expresses information in relation to the current position of the perceiving agent in space. A fourth frame contains world knowledge

and is thus not directly coupled to vision and space.

Vision plays a crucial role in the development of allocentric frames, as demonstrated in a study comparing the performance of congenitally blind subjects with sighted and blindfolded subjects in estimating distance between a target object and the participant (ego-centric) or another object (allocentric) [145].

Tadin and colleagues [169] show that discrimination of motion and coherence of changes benefits from suggesting an allocentric, object-centered reference frame, instead of relying purely on a retinal representation of presented stimuli.

2.1.6 Models of Visual Perception

Several models offer insights into the structure of visual perception and working memory. Rensink's *coherence theory* [135] splits up visual perception in two stages. In a low-level stage, proto-objects form and dissolve rapidly in parallel driven by visual stimulation, with little coherence in time and space. An attention stage stabilizes a subset of these proto-objects into a coherent object representation. If visual stimulation at the focus of attention changes, it is consciously perceived as a change. Once attention is released, the object representation vanishes, thus returning to the volatile proto-object stage. Coherence theory has no concept of memory beyond the object representation. Change detection is explained by having the changing object in the attentional foreground when change occurs. Proto-objects and object representations are comparable to unbound features and object files in feature-integration theory of attention [180].

Schneider [155] favors a different two-stage model of visual processing, based on previous work by Neisser [117]. On a low-level stage, candidates for the second stage are generated in parallel. These candidates are called *visual-spatial units*. The second stage sequentially applies high-level operations onto these candidates. Operations comprise object recognition, creation of object files as representation of objects (relating to feature-integration theory), and extraction of motor parameters for movement generation. Distinction of object files in working memory is achieved through temporal coding of neural populations, that is, all represented features of an object fire synchronously in a given time slice, while other object files occupy other time slices. Capacity limits of visual working memory arise from the minimal size of a time slice necessary to sustain an object file.

The *visual memory theory of scene representation* by Hollingworth and Henderson [77] extends on the concept of attentional binding of features by extracting a high-level object representation at the focus of attention. This

representation is transferred to long-term memory (LTM), while a link to the LTM representation is kept in short-term memory. A spatial re-selection of a previously attended location gives access to the accumulating LTM and allows for change detection. In this model, change blindness occurs if no object representation was created prior to the change.

Johnson and colleagues [87] present a model split up into feature working memory and space-feature working memory. While the former is responsible for producing fast (i.e., parallel), but location-unspecific change responses if new features are perceived, the latter is necessary to solve any change detection task requiring a binding of features. Working memory limits arise from characteristics of the chosen representation substrate—neural dynamics. The role of attention on binding features is not discussed by the authors. Assigning a specialized role to space in working memory can be traced back to feature-integration theory, as it assumes a spatial spotlight of attention to bind features [178]. Feature-integration theory also supports the storage of different feature dimensions in separate maps [187].

Bays and colleagues [9, 10, 57] favor a shared resource view of working memory. They find that with increasing load, fidelity decreases. The authors oppose an item-based limitation of working memory based on their findings. The shared resource view resonates with the previously mentioned model by Johnson and colleagues, as their flavor of neural dynamics uses continuous representation and limits arise naturally from the interactions between the active entries in working memory.

2.1.7 Summary

In summary, the various contributions to the understanding of human visual perception form a coherent picture. The main task for visual perception is visual search, that is, bringing a target object into the attentional foreground. Visual perception is drawn to salient regions defined by bottom-up uniqueness and top-down cues. An attentional bottleneck processes regions one at a time, be it covertly or overtly. Attentional focus allows access to a detailed, bound description of a region. A part of this rich description is entered into working memory, which also serves as a gateway to long-term memory. Changes between memory and the visual scene can be detected without transient cues such as motion detection.

2.2 Computer Vision and Robotics

Making sense out of complex scenes is still a hard problem for robotic agents [11]. Flexibility, both in the sense of adaptive task execution and dealing with dynamic environments, is rarely a topic. For this cross-section, I present related research in the fields of robotics and computer vision that draws inspiration from human vision and aims at an integrated account.

High-resolution cameras and depth-based sensors, such as the Microsoft Kinect and laser-range scanners, produce detailed and rich inputs with a high refresh rate, having a direct impact on the field of computer vision (see, for example, a review of Kinect-related research [65]). Sensors are combined to form an even richer portrayal of the environment (up to a full reproduction of the external world [85]). Robotic vision serves a number of tasks, as identified by a review of Chen and colleagues [24]. Among these tasks are the modeling of unknown objects and environments and using vision to manipulate objects. The robotics community offers a broad range of special-purpose solutions that are well-suited to solve such tasks or parts of it. However, there is no trend to build an integrated account of perception solving a multitude of tasks. This is understandable, as areas of application (for example, executing a work step of an assembly process) only require solutions for a subset of tasks. However, there is research that addresses coping with the sensory abundance of information and integrating vision into larger architectures interacting with the environment.

Principles of human visual perception are taken into account by a robotic architecture for object recognition and pose estimation by Björkman and Kragic [13]. Here, attention guides an object recognition process applied to a fovea-like region of the input. Poses are estimated and subsequently tracked. Later work of the same research group uses the extracted object description acquired from the object recognition process to parameterize object manipulation with a robotic arm [134]. The authors term their work *active vision*, as the robot actively interacts with its environment to gain knowledge about objects.

Ognibene and colleagues present an integrated architecture of perception, attention, and action in a reaching scenario [122]. Here, bottom-up saliency and learned top-down influences are combined to guide the attention of a robotic agent to a target object. Attention is then used to extract parameters for movement generation of eyes and arm. The architecture uses dynamic neural fields for representation of metrical estimates and selection decisions.

Kühn and colleagues combine auditory and visual perception for scene analysis [99]. Visual processing in their system is saliency-based. Auditory processing generates a similar map through sound source localization. The

fusion of both modalities is used to drive an attentional process for scene exploration. The attentional process is able to integrate top-down cues such as pointing gestures and color terms [154].

Haazebroek and colleagues combine sensory-motor processes with task-level influences for a robotic agent in a connectionist model called HiTEC [61]. They discuss that task influence weighs feature contributions to the actions taken by the robotic agent, calling it a form of attention. This relates to top-down influence on attention during visual search.

Representing dynamic scenes is the goal of work by Blodow and colleagues [14] who combine a continuous passive perception pipeline with a dynamic object store. Pose and identity of objects are continuously extracted from the input stream and integrated into the dynamic store using probabilistic modeling, which allows to express uncertainties for objects that are out of view and to track pose changes. A similar emphasis on dynamic scenes is expressed by Einecke and colleagues [38] whose integrated architecture stores perceived objects in a persistent object memory (POM). To keep the POM aligned with changes in the scene, the whole robot body is used to align a visual frustum onto the scene to keep relevant objects in view.

Similar to the saliency operation, which highlights certain areas of the image with locally unique feature combinations, computer vision uses keypoint-based feature descriptors to condense a high-dimensional visual inputs to representations of most relevant positions in the input, which are invariant under transformation operations such as rotation and scaling. A prominent algorithm is the scale invariant feature transform (SIFT) by Lowe [104], which can be applied to object recognition by matching feature descriptions at keypoints with a nearest-neighbor search. Lowe emphasizes the connection of SIFT to the parallel processing and attention prominent in human vision.

Viola and Jones [183] use a cascaded set of classifiers to detect faces in an image. Their approach resembles saliency processing as early cascade stages reject the majority of image regions. Only the remaining regions are inspected by more complex and thus more time-consuming classifiers, with each stage growing in complexity and required time. While their approach does not use a strictly serial attentional bottleneck, the resulting map contains classified, but not yet recognized entities, which may serve as a foundation for a time-consuming serial face recognition process.

A related research field in robotics is the creation of environmental maps for mobile robotics. Kuipers [100] presents a set of hierarchically related maps called spatial semantic hierarchy (SSH), which contain geometrical information as well as sensory recordings and associated actions at discrete locations. SSH is designed as an analogy to human cognitive maps. Pronobis and colleagues [130] follow a similar approach for robot navigation. They

use multiple maps of different degrees of abstraction (e.g., metrical, discrete nodes, topology) and can attach conceptual labels to these maps (e.g., a television placed in a living room). Landmarks for navigation are chosen by a visual attention system that relies on saliency to pick a limited amount of landmarks from the input stream. Nüchter and Hertzberg [121] enhance a spatial map with semantic labels. They differentiate between coarse scene classifications (e.g., floor, wall, ceiling, door) and the detection and localization of objects, both extracted from 3D point cloud data. While coarse classifications are based on orientations of surfaces, object recognition uses a trained classifier, which also estimates the 6D pose.

2.3 Behavioral Organization and Autonomy

Scene representation comprises several processes, which not only require organization among themselves when accessing bottlenecks such as attention, but also interface with other processes of cognition and movement generation, demanding an organization of sequential structure. In addition, the integration of all processes assumes behavioral autonomy, that is, the overt behavior of such an integrated system does not require a separate controlling instance that orchestrates each process. Initiation and termination of behavior, as well as coordination are thus contained in the integrated system and each involved process. The following cross-section of research motivates behavioral organization and its implementation.

The overt behavior of agents is a window into the internal processes bringing it about. Central concepts emerging from this are coordination (e.g., moving the arm and opening the hand to prepare a grasp) and sequentiality (e.g., first closing the hand around an object and then moving the arm up to lift it from the ground). How is overt behavior generated? There are two opposing theories reviewed by Cisek and Kalaska [28]. The first theory is related to information processing and considers cognition as the central component of behavior generation, with sensors and actuators having a subordinate contribution. Perception builds up complex representations of the world state (in the sense of degree of detail and completeness). Cognitive processing generates plans from these, which are subsequently handed to low-level motor processes to be executed. The second theory reduces the cognitive stage to a minimum and replaces planning by a multitude of distributed stimulus-response processes that each may contribute to the overt behavior of an agent. Overt behavior is shaped by selecting which processes actively contribute to it. Cisek and Kalaska [28] report that studies of neural data favor a distributed view of behavior generation.

Cisek and colleagues [29] model the decision to activate a single behavior in a distributed system as integrate-to-threshold process, with time to activation depending on integrated stimulus strength and an additional urgency signal. Cisek later expanded decision-making among potential behaviors to a distributed consensus across multiple levels of abstraction [27]. Here, the suitability of actions and more abstract goals is represented by relative values. Recurrent competition between potential actions and goals converges onto a single, but distributed decision of generated behavior.

For robotic agents, approaches to generate overt behavior follow the two theories above, as discussed by Brooks [19]. The first approach is labeled *sense-plan-act* and places a cognitive planning stage between the perceptual processing of the environment and the generation of action. Coordination can be attributed to this stage, as every input first passes through it before any action is generated. The second approach is labeled *behavior-based* and uses small sense-act units as basis of behavior. Hierarchies of independent sense-act units represent a pool of behaviors, with more complex higher-level behaviors recruiting lower-level behaviors. Moving coordination and sequentiality into its own stage puts computational complexity into the planning stage of the first approach, resulting in delays and the inability to adapt to changes in the sensory input. With independently acting sense-act units, delays and adaptation are minimized. Solving coordination and sequentiality with a recruitment scheme however poses the problem of how recruitment is organized in time to yield the emerging overt behavior. Both approaches imply autonomy, that is, a robotic agent is able to create on its own meaningful overt behavior for a variety of environments and tasks, either by having an exhaustive planner or flexible behavior recruitment.

Arkin [4] defines three components contributing to the overall behavior of a robotic agent using the behavior-based approach: 1. A set of *individual behaviors* that, for a given stimulus, produce some response or reaction. 2. *Attention* as means of prioritizing tasks and focusing resources given environmental constraints. 3. *Intention* reflecting the internal goals and motivation of the agent by selecting a subset of currently active individual behaviors. The overall behavior emerges from the combination of individual behaviors through intention and the environment in which the agent is placed. The behavior-based viewpoint implies that each individual behavior is independent from all other behaviors, having access to only the sensor input it requires to produce its response. Coordination and sequentiality required for more complex behaviors thus has to happen on another level of description.

Konidares and colleagues [96] present a hybrid approach in which a classical planning approach is applied to symbols acquired from low-level sensorymotor processes. They use semi-Markov decision processes as a model of

the low-level sensory-motor processes, classifying them into discrete states to which a high-level planner can be applied.

Steinhage and Bergener [166] present a neuro-dynamic approach to behavioral organization. Here, the activation of each behavior is represented by a state variable of a dynamical system. Each such dynamical system receives input from the sensory surface. Refractory dynamics smooth these inputs to prevent oscillatory activation of behaviors. In addition, the different behaviors may interact with each other, implementing competition and competitive advantage. These interactions are defined in coupling matrices, containing the pairwise relation between all behaviors. The resulting behavior emerges from this coupling structure and the current sensory input. Adding new behaviors leaves the existing coupling structure intact.

Based on Searle's work on intentionality [160], Richter and colleagues [139] define a level that organizes individual behaviors of an agent. Here, individual behaviors are called elementary units of behavior, which can be assembled through means of preconditions and suppressions to perform a given task. These units, called elementary behaviors (EB), are characterized by their elementary cognitive units (ECU), defining their initiation and termination: the *intention* and its *condition of satisfaction* (CoS). The former is the driving force of any behavioral change induced by any EB (e.g., by generating a specific motor output), the latter monitors the success of achieving whatever the intention of the associated EB implies (e.g., reaching a specific joint angle configuration). Once the CoS is reached, the behavior turns itself off. Other EBs that were suppressed by this behavior (either through a relation of suppression or precondition) can now become active on their own (that is, if all preconditions are fulfilled and no other mutual exclusive behavior is currently active). In addition to this horizontal interaction between EBs, several units can also be grouped together to higher-level behaviors that follow the same internal structure of the elementary units [36]. All interactions between EBs, horizontally and vertically, are realized between the ECUs of each EB only, which separates behavior execution from organization. The coupling structure between EBs (preconditions and suppressions) is defined by tasks that the robot is currently pursuing and can be reconfigured by turning specific tasks on or off. The connections between tasks, associated EBs, and couplings are learned at some point in the development process and will not be the focus of my thesis.

Behaviors are not solely based on current sensory input, but may also use forms of internal representation already present at initiation (think of a grasping behavior that relies on previous processing of sensory input generating an internal representation of object identity and pose in a scene). This is an inherent conflict with the stimulus-response structure of elementary

behaviors, which enables flexibility in dynamical environments (again, think of a grasping behavior that is aimed at a target object moved around by a human). One solution to this caveat is introducing perceptual behaviors. Their purpose is to establish and maintain the perceptual conditions later behaviors can be applied to. Preconditions in the behavioral organization assure that the internal representation established by a perceptual behavior is present before activating any behavior that relies on them. See [92] for an exemplary robotic architecture using perceptual behaviors that feed into motor behaviors. The behaviors involved in scene representation are of such perceptual nature, as internal representations are generated, maintained, and re-instantiated based on the history of the sensory input stream. The attentional bottleneck involved in these behaviors however is tightly linked to eye movements, an observable indicator of the internal processes.

A separation into two types of behaviors is also present in Searle's discourse [161]. Searle differentiates between perception and action by defining a direction of fit for intentionality. For perception, intentionality has a *mind-to-world* fit, that is, the content of the mind (e.g., an internal representation) is adapted to the world. For action, intentionality has reverse fit of *world-to-mind*, as an intentional agent tries to effect changes in the world so that it matches an internal state (e.g., moving my hand to a cup to fulfill an intention to have a cup in the hand).

2.4 Dynamic Field Theory

Computational modeling of brain processes happens on various levels of abstraction, from chemical processes in single neurons over intermediate levels of modeling fire rates or activation levels of neurons up to abstract descriptions that are detached from the neural substrate, for example probabilistic approaches.

Dynamic field theory is a modeling framework on an intermediate level. Dynamic field theory's premise is that neural processes can be abstracted to a certain degree without loosing behavioral significance. In this view, biological details of the nervous system, such as chemical processes of neurons, spikes, and synaptic connectivity, do not contribute to the behavior and can thus be neglected. Instead, dynamic field theory describes the activation of the nervous system on a population level. Belonging to the family of neural dynamics, it formulates activation in neural substrate as time-continuous dynamical systems.

Populations of neurons are grouped into fields and nodes, with distinct behavioral characteristics. Fields cover continua of metrical feature values,

with peaks of activation representing concrete values. Activation of the population around peak center decreases in analogy to tuning curves of single neurons. With increasing feature distance, sites show less activation as the represented value differs more and more from their preferred value. The activation of field populations is dominated by lateral interactions, implementing basic cognitive operations such as detection and selection decisions and working memory. The abstraction to population activation still keeps fields grounded in sensory and motor processes. DFT thus is not faced with the problem of grounding symbols in the world (see [68] for a definition), as cognitive operations such as detection and selection are not applied to abstract symbols, but representations directly extracted from sensory or motor surfaces. This tight integration with sensory-motor processes follows the principle of *embodied cognition*. Certain phenomena of the nervous system, such as plasticity regulated by the time difference of action potentials [109], cannot directly be expressed in DFT due to the used level of abstraction.

Dynamic field theory emerged as a theory of motor control (eye movements [97, 98], movement preparation [41]), with later applications in the area of visual processing and cognition [87, 157] and infant development [30, 159, 199, 200]. The integration with sensory-motor processes is emphasized in robotic applications [40, 93, 125, 149, 150, 158].

Chapter 3

Methods

In this chapter I present the methods used in my work. I present elementary building blocks of dynamic field theory and analyze relevant properties of the underlying dynamics. Groups of building blocks form larger recurring components with distinct functionality such as behavioral organization, match detection, and reference frame transformation. I conclude with a brief overview of how the mathematics presented in this chapter translates onto software that can be connected to robotic platforms, which I also introduce. The equations in the remaining chapters follow the notation defined in Appendix B.

3.1 Elementary Building Blocks of Dynamic Field Theory

Throughout this work, I will use the biologically-inspired modeling language dynamic field theory (DFT) [157]. In DFT, perceptual, cognitive and motor processes are formulated in neural dynamics, more precisely with interconnected dynamic fields (DFs) and dynamic nodes (DNs). Both fields and nodes are non-linear attractor dynamics.

3.1.1 Dynamic Fields

A DF of the form

$$\begin{aligned} \tau \dot{u}(x,t) &= -u(x,t) + h + s(x,t) \\ &\quad + [w * \sigma(u)](x,t) \end{aligned} \tag{3.1}$$

describes the evolution of neural activation u over time over a continuous feature space x (e.g., spatial position or hue). Over time, a dynamic field relaxes

to stable solutions (for analyses of the dynamics, see [2, 170]) defined by three components: the resting level h, external input s (e.g., activation of other fields), and the lateral interaction kernel w with excitatory and inhibitory components. Lateral interaction is determined by convolving thresholded field activation with the interaction kernel, which is expressed by the binary operator $*$. The constant τ determines how fast the field adapts to changes in these components. Whenever a region of a field pierces its threshold defined by the transfer function σ with steepness β,

$$\sigma_\beta(u(x,t)) = \frac{1}{1 + e^{-\beta u(x,t)}}, \tag{3.2}$$

Gaussian-shaped local excitation stabilizes this *detection* decision. The result is a localized bump of supra-threshold activation, which I call a peak from here on. Peaks serve as the unit of representation, as their position along the feature metric x represents a specific value. *Selecting* one out of multiple regions given a specific input is achieved by global inhibition (see Figures 3.1 and 3.2; for an in-depth analysis, see [111]). Multiple peaks may arise at different locations along the feature space by replacing global with localized inhibition. *Working memory* occurs if lateral interaction is strong enough to sustain activation even if the input is removed (see Figure 3.3). Changes in input position along the feature metric of a field are *tracked* by peaks (see Figure 3.4). Dynamic fields may span more than one feature dimension, yielding a combined representation of all involved metrics (e.g., a two-dimensional space-color field). Peaks in multidimensional fields encode estimated feature values along all metrics, for example, the representation of the color "red" at a specific spatial position. I call this a *link*, following the definition by Zibner and Faubel [195].

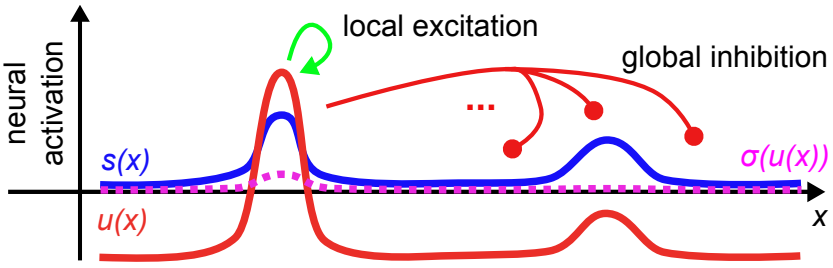

Figure 3.1: A dynamic field over x receives input at two distinct field sites and selects one of the inputs.

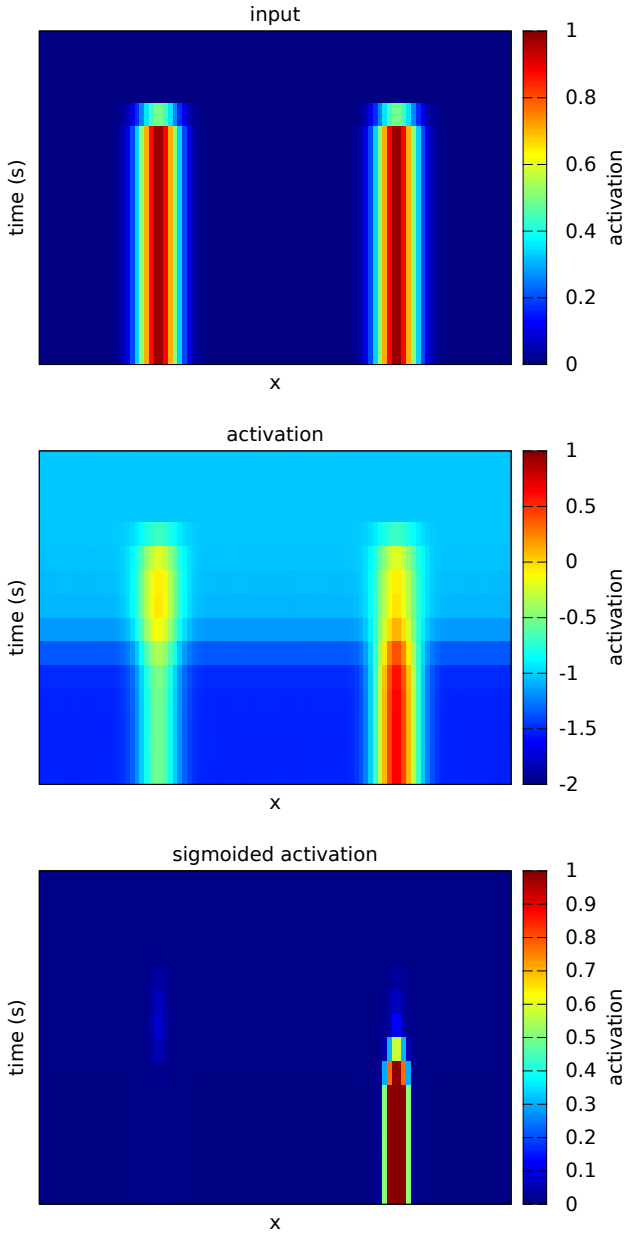

Figure 3.2: A one-dimensional DF receives localized input at two regions of the metric x. Over time, field activation at both sites increases. Field activation close to the detection threshold contributes to lateral interaction. Local excitation stabilizes the localized patterns, while global inhibition influences the whole field. As a result of these contributions, only one site may reside above threshold.

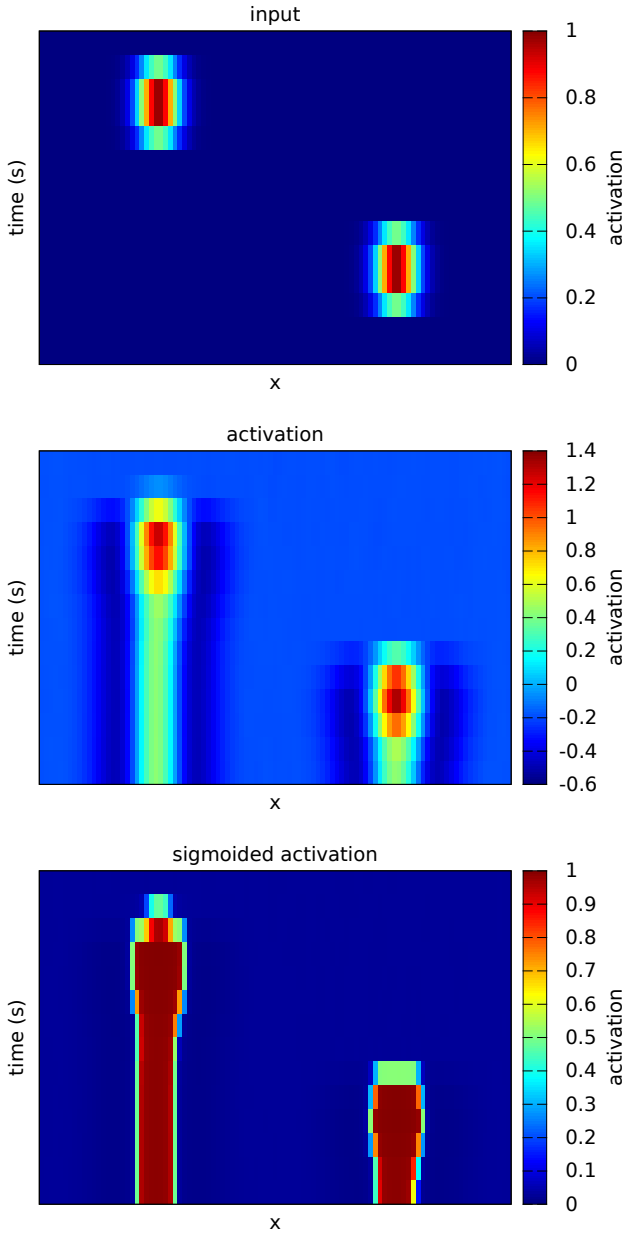

Figure 3.3: A one-dimensional DF receives a sequence of localized inputs at two regions. These inputs vanish after a brief time period. Local excitation keeps the sites above threshold after the inputs have vanished. Mid-range inhibition counteracts the excitatory component of the interaction kernel. This prevents activation from spreading along x.

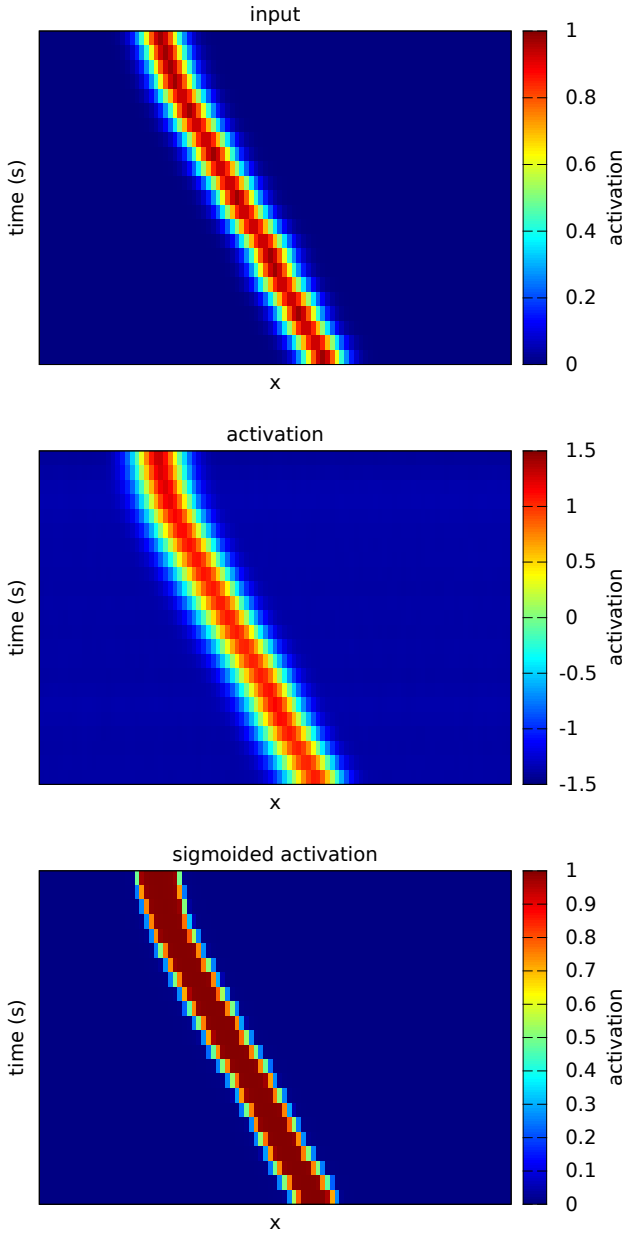

Figure 3.4: A one-dimensional DF receives localized input that moves along the field metric x over time. The peak representing the localized input moves along and tracks the changes in input, with a delay determined by the time scale τ.

3.1.2 Connections

Besides the individual parameterization of each DF, a DFT model derives its behaviors from the connections between DFs. There are four categories of connections: *direct*, dimensionality *expansion*, dimensionality *contraction*, and *arbitrary*, as described by Zibner and Faubel [195].

Direct connections exist between DFs covering the same metrical dimensions. They are used to apply different field interactions to a given input (e.g., having one field detecting multiple peaks, while a second field takes these peaks and selects one of them to let a third field memorize its location).

Expansions are projections from DFs with certain metrical dimensions to DFs covering both the metrical dimensions of the source field as well as additional metrical dimensions (e.g., a field covering color hue projecting to a field covering color hue over space). Field activation is expanded along the additional metrical dimensions, resulting in characteristic activation patterns named for their appearance (*ridge* for projections from 1D to 2D, *tube* for projections from 2D to 3D, *slice* for projections from 1D to 3D, see [195] for details). Figure 3.5 illustrates common expansion patterns. Expansions are used to create dynamic links of peaks across different feature spaces (e.g., combining spatial and color activation to form a map of color over space). This is achieved by overlapping multiple expanded inputs, which alone stay below threshold in the receiving field, but pierce the detection threshold of said field at intersections of inputs.

Contractions are projections from DFs to DFs with less metrical dimensions (e.g., a field covering color hue over space projecting to a field covering only color hue). Contractions require a function that determines how activation along the metrical dimensions not covered by the receiving field is contracted. One example is a function that integrates along contracted dimensions. Figure 3.6 illustrates common contraction patterns. Contractions dissolve dynamic links to access their components for further processing. See Appendix A for software examples of expansions and contractions.

Arbitrary connections in turn link sites of one field to arbitrary locations of another field through some learning rule (e.g., Hebbian learning), with a mapping that is trained or continuously adapts (think of learning a coordinate transformation between a retinotopic and an allocentric representation).

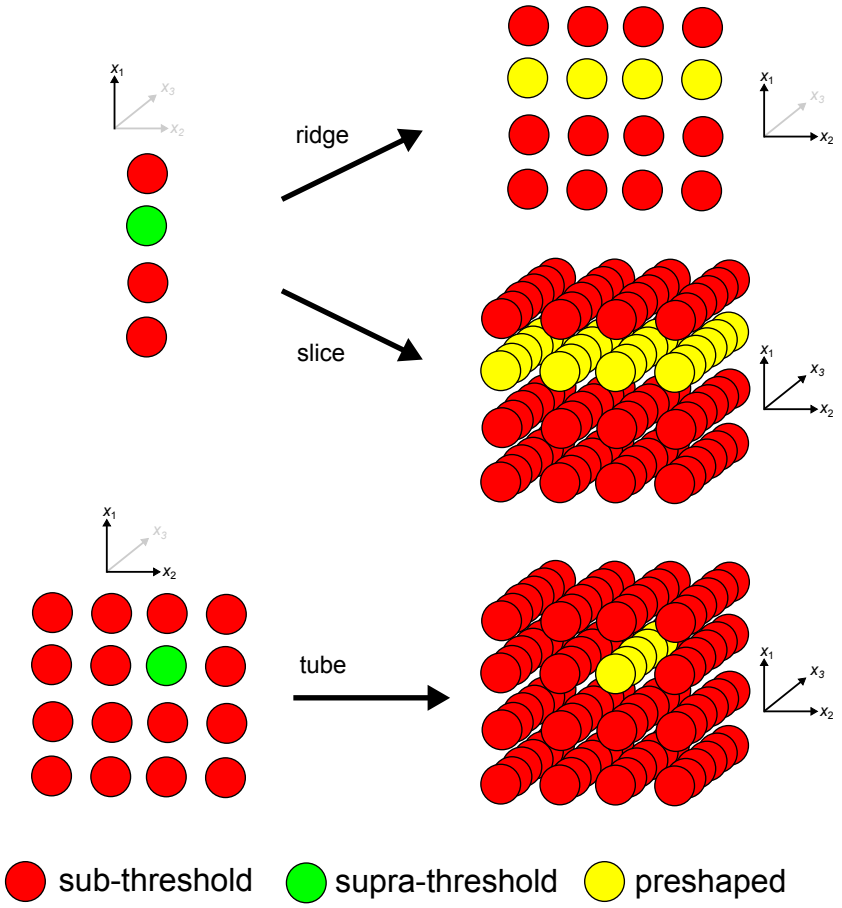

Figure 3.5: This figure illustrates three common expansion connections between DFs. Here continuous dimensions are depicted as discrete regions for the purpose of illustration. Active regions in the source field are colored green, while regions receiving input from active regions are colored yellow.

3.1.3 Dynamic Nodes

In contrast to the DF's activation along feature spaces, DFT also uses discrete dynamic nodes without metrical extend,

$$\tau \dot{u}(t) = -u(t) + h + c\sigma(u(t)) + s(t), \tag{3.3}$$

which relax to one of at most two stable states, on or off, depending on the time-dependent input s. Connections between DFs and nodes follow the *expansion* and *contraction* principles. Four types of nodes emerge from specific connection patterns. A node u^{rb} may project to every location of a

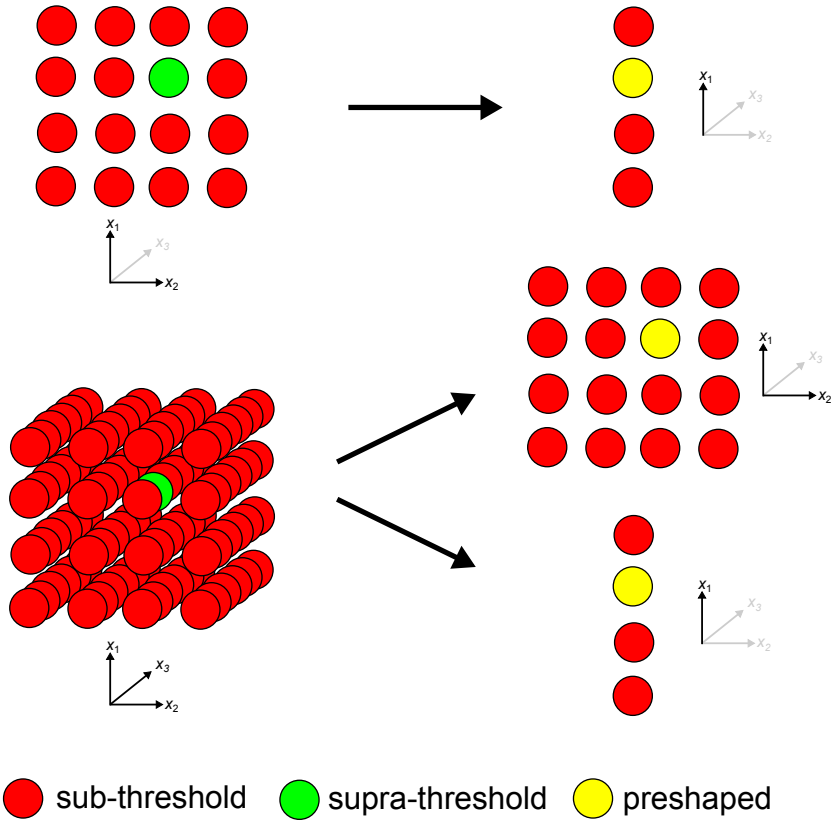

Figure 3.6: This figure illustrates three common contraction connections between DFs. Here, continuous dimensions are depicted as discrete regions for illustration purpose. Active regions in the sending field are colored green, while regions receiving input from active regions are colored yellow.

DF u, implementing a switchable boost of this field's resting level,

$$\begin{aligned} \tau \dot{u}(x,t) &= -u(x,t) + h + s(x,t) \\ &\quad + [w\sigma(u)](x,t) \\ &\quad + c_{\text{field,rb}}\sigma(u^{\text{rb}}(t)). \end{aligned} \tag{3.4}$$

Connecting every location of a field u to a node u^{pd},

$$\begin{aligned} \tau \dot{u}^{\text{pd}}(t) &= -u^{\text{pd}}(t) + h \\ &\quad + c_{\text{pd,field}} \int \sigma(u(x,t))dx, \end{aligned} \tag{3.5}$$

turns this node into a detector for a minimal amount of supra-threshold field activation (i.e., a peak detector), without specifying the location or amount

of peaks. Localized connections to and from a DF turn a node into a category inducer u^{ci},

$$\begin{aligned} \tau \dot{u}(x,t) &= -u(x,t) + h + s(x,t) \\ &\quad + [w\sigma(u)](x,t) \\ &\quad + w_{\text{field,ci}}(x)\sigma(u^{\text{ci}}(t)), \end{aligned} \tag{3.6}$$

with weights $w_{\text{field,ci}}(\cdot)$ specifying the affected range of the field, or a category detector u^{cd},

$$\begin{aligned} \tau \dot{u}^{\text{cd}}(t) &= -u^{\text{cd}}(t) + h \\ &\quad + \int w_{\text{cd,field}}(x)\sigma(u(x,t))dx, \end{aligned} \tag{3.7}$$

with weights $w_{\text{cd,field}}(\cdot)$ specifying the observed range of the field. These two node types might, for example, detect peaks around the location of "red" in a field covering color hue or induce such a peak in a DF if the node becomes active. $w_{\text{field,ci}}(\cdot)$ and $w_{\text{cd,field}}(\cdot)$ are single mode Gaussian-shaped functions. See Appendix A for software examples of the four introduced node types.

3.1.4 Gradedness of Activation

The gradedness of inputs to a DF plays a crucial role in detection and selection decisions. If localized input is sufficiently strong and consistent over time, peaks may arise and selection may occur. Without gradedness of inputs, that is, a continuum of input range, detection and selection decisions are degenerate. Think of a discrete input that can be either zero or one. Detection decisions thus may only rely on temporal stability of the signal. Selection decisions likewise depend on neural noise alone to determine a site with ones that wins the competition against all other candidates (assuming a fixed width of local regions).

The gradedness required to execute meaningful detection and selection decisions is obvious on the sensory side, as sensors are a source of graded input. Think of a camera sensor measuring the amount of "red" for each pixel or a convolution of the image with an oriented edge filter. The resulting output depends in a graded way on the redness of a region or the similarity between preferred direction of the filter and local edges in the image. If one moves away from the low-level sensory interface, one finds that suprathreshold peaks of activation in one DF layer pass a sigmoid function before affecting other layers through inter-layer connectivity. The sigmoid function compresses the gradedness contained in the field activation, up to a quasi-binary representation for large steepness of the sigmoid function. In addition,

fields may be interaction-driven instead of input-driven, that is, the shape of activation only depends on the lateral kernel and lost any link to the input which created it (e.g., in working memory fields).

How does gradedness fit into the picture of sigmoid functions and interaction-driven peak shapes? A source of gradedness in higher neural layers is peak size. The interaction and projection kernels of DFs translate peak width into a graded response. For single mode excitatory kernels, activation runs into saturation once a certain width is reached (see Figure 3.7). Kernels with mid-range or global inhibition show a preferred peak width in their graded response (see Figure 3.8).

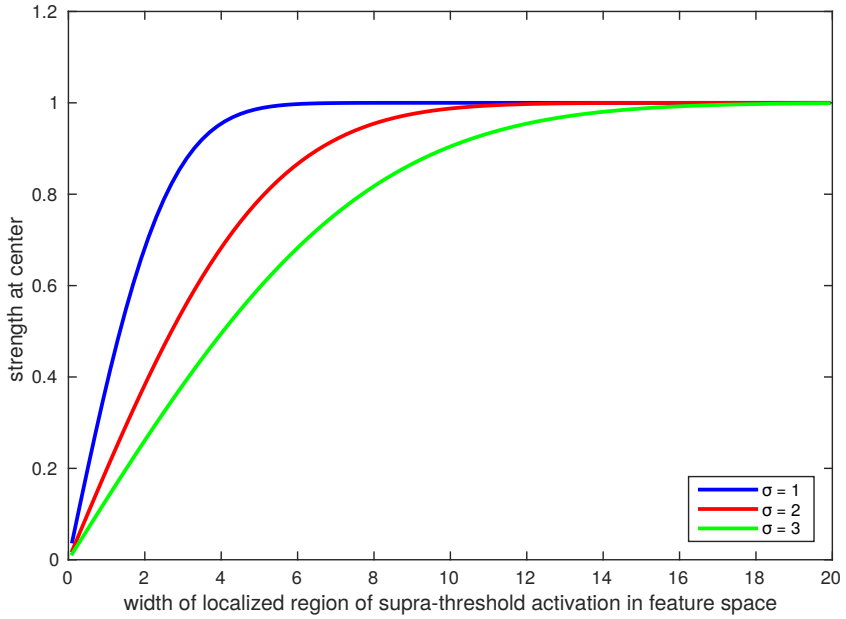

Figure 3.7: This figure shows the lateral excitation at the center of a peak after applying the sigmoid function and convolving with a single-mode kernel. Different colors denote different kernel widths. Depending on the kernel width, the activation saturates for different peak widths.

Gradedness can also be achieved by combining multiple additive inputs, either localized or on a global scope. Consider the following example for detection decisions: a space-color field receives input from separate space and color fields. Peaks in the separate fields preshape the space-color field along ridges, which are not strong enough to pierce the detection threshold. At all points of intersection, however, space and color activation combined is strong enough to induce a detection, which results in peaks combining space

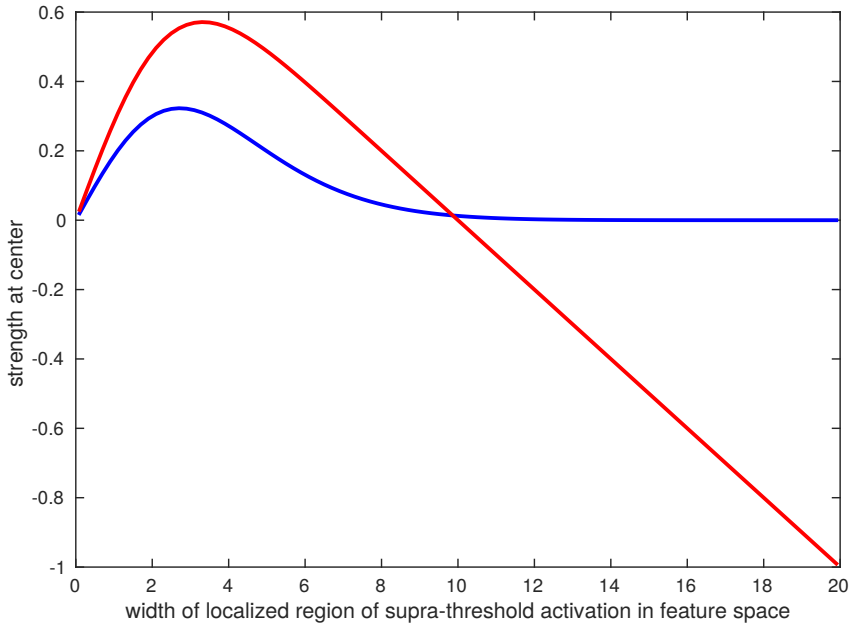

Figure 3.8: This figure shows the lateral excitation at the center of a peak after applying the sigmoid function and convolving with kernels featuring mid-range (blue line) or global (red line) inhibition.

and color representations at sparse locations in the field. If one takes the same space-color field and puts it in a working memory regime, a subsequent selection field over space would select locations at random, regardless of the color memorized at each location. To bias this decision for a specific color, one cannot simply locally increase the activation in the space-color field, as the sigmoid function is most likely in an area of saturation. Instead, using an additive ridge-like cue input at the preferred color value establishes the needed gradedness for the selection process. The closer a memory entry resembles the cued color, the greater its competitive advantage becomes. Note that even a weak cue is sufficient to bias selection. Cues have an influence on represented values. If the peak center of a cue is not exactly at the same feature value as the winning input region, the resulting selection exhibits a shift towards the cue. This shift becomes larger with increasing cue strength, up to a point where the cue dominates the other localized input. At this point, cue and localized input exchange roles in biasing the selection process. Selected peaks are closer to the cue than to the localized input, producing categorical responses with a bias given by the localized input. Examples of the influence of cue strength on peak position are shown

in Figure 3.9. Appendix A contains a software example of different degrees of biasing.

A graded bias may also develop over time in form of an adaptive memory trace as additive input [30, 196]. Memory trace dynamics are of a graded nature, as memory decays over time, decreasing its influence on selection decisions.

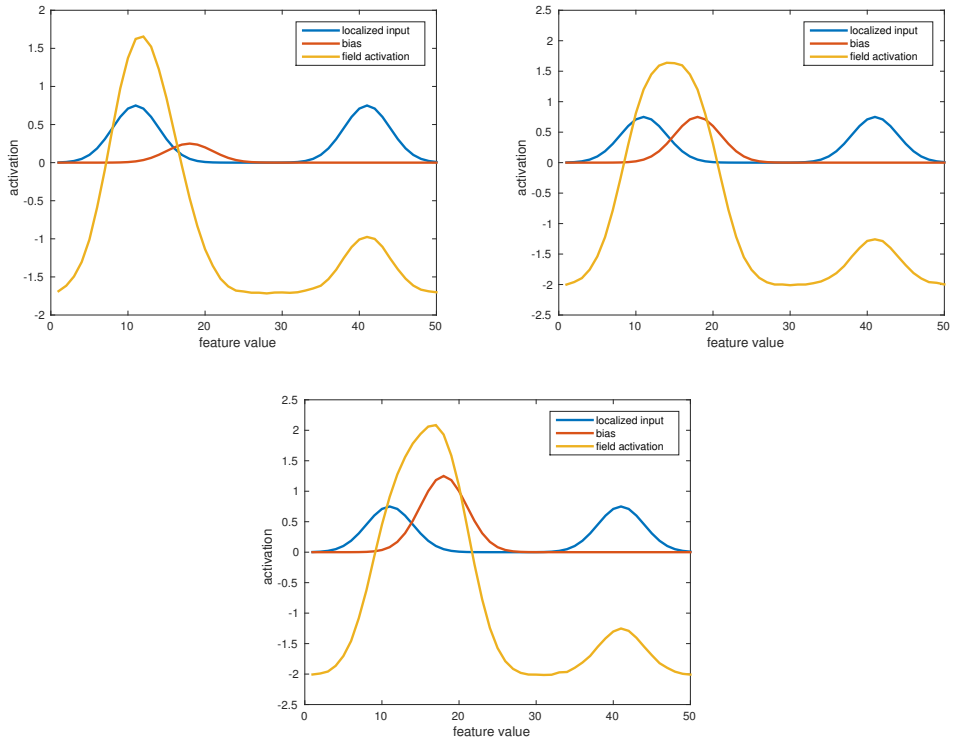

Figure 3.9: This figure shows the influence of bias strength on the representation created through a selection decision. With increasing bias strength (from left to right, top to bottom), the represented value is pulled towards the cue and away from the localized input.

3.2 Recurring Components in Dynamic Field Theory

Besides elementary building blocks of neuro-dynamic architectures, DFT offers ready-made components consisting of several building blocks and corresponding connections that provide higher-level functionality going beyond

detection, selection, working memory, and tracking. Here I present three such higher-level components and point to publications that motivate these components in more detail.

3.2.1 Behavioral Organization

Behavioral organization is concerned with both the initiation and termination of single behaviors as well as implementing relations between single behaviors such as preconditions and mutual exclusion. Richter and colleagues [139] define *elementary behaviors* (EB), which are governed by *elementary cognitive units* (ECU) consisting of the intention to execute a behavior and the condition of satisfaction signaling its successful completion. Relations between EBs are expressed through preconditions and suppressions.

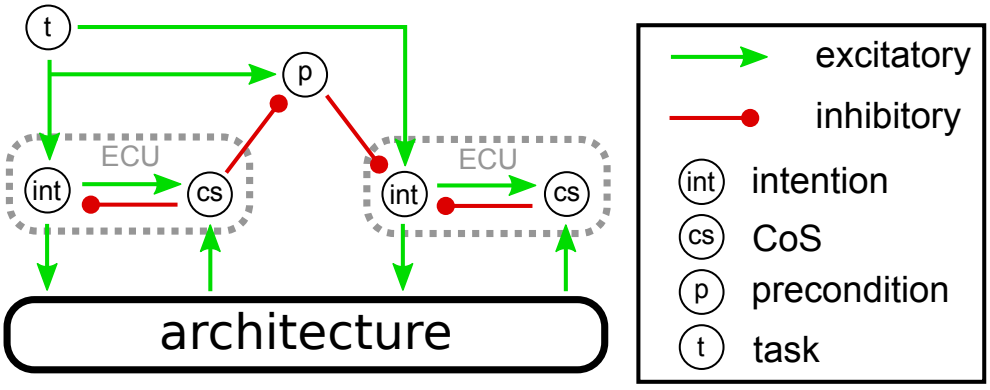

Figure 3.10: This figure shows two ECU templates, each consisting of an intention node and a CoS node. The left ECU is a precondition of the right one. Both ECUs as well as their conditional dependency can be recruited by task input.

In DFT, the structure of an EB is transferable onto a template consisting of two dynamic nodes and DFs that these nodes connect to (see Figure 3.10). The intention node u_{int},

$$\begin{aligned} \tau \dot{u}_{\text{int}}(t) &= -u_{\text{int}}(t) + h + c_{\text{int}}\sigma(u_{\text{int}}(t)) \\ &\quad -c_{\text{int,cos}}\sigma(u_{\text{cos}}(t)) + s(t) \\ \text{with } s(t) &= -\sum_i \sigma(u_{\text{pre},i}(t)) - \sum_j \sigma(u_{\text{sup},j}(t)) \\ &\quad +\sigma\left(\sum_k \sigma(u_{\text{tsk},k}(t))\right), \end{aligned} \tag{3.8}$$

expresses the intentional execution of a behavior, while the condition of satisfaction node u_{cos} is a detector for the CoS of the EB,

$$\begin{aligned} \tau \dot{u}_{\text{cos}}(t) &= -u_{\text{cos}}(t) + h + c_{\text{cos}}\sigma(u_{\text{cos}}(t)) \\ &\quad + c_{\text{cos,int}}\sigma(u_{\text{int}}(t)) + s(t). \end{aligned} \tag{3.9}$$

The intention node connects to any DF or node involved in the execution of the behavior, giving an excitatory local or global boost if the intention node is in its on state. It is preshaped by task input from a number of K nodes $u_{\text{tsk},k}$. A set of I precondition nodes $u_{\text{pre},i}$ and J suppression nodes $u_{\text{sup},j}$ keep the intention node from becoming active as long as preconditions of this behavior are not met or competing behaviors are active, respectively. After the behavior has reached its CoS, the CoS node inhibits the intention node.

The CoS node monitors DFs for a given event (e.g., the creation of a peak by means of a peak detector, as described in the previous section) to stop the EB, indicated by the input $s(t)$. Excitatory input from the associated intention node u_{int} with weight $c_{\text{cos,int}}$ is tuned to only allow for detection decisions of $s(t)$ if the behavior is intentionally activated. Since the CoS node turns off the intention node, one has to take care of sustaining the fact that the EB was already executed by either using an additional CoS memory node or letting the CoS node stay active due to strong self-excitation $c_{\text{cos}}\sigma(u_{\text{cos}}(t))$.

The inhibition that the CoS node projects back to the intention node can also be fed to inhibitory precondition nodes

$$\begin{aligned} \tau \dot{u}_{\text{pre}}(t) &= -u_{\text{pre}}(t) + h - c_{\text{pre,cos}} \sum_k \sigma(u_{\text{cos},k}(t)) \\ &\quad + \sigma\left(\sum_l \sigma(u_{\text{tsk},l}(t))\right) \end{aligned} \tag{3.10}$$

that inhibit the intention nodes of all EBs whose activation depends on the successful completion of the EB related to the CoS node. This connectivity requires the precondition nodes to be in a supra-threshold state before a CoS deactivates them. This is achieved by turning on task nodes $u_{\text{tsk},l}$, which have excitatory connections to all task-relevant precondition nodes.

Besides expressing sequential dependencies of EBs using precondition nodes, mutual exclusion of behaviors can be implemented with suppression

nodes,

$$\begin{aligned}\tau\dot{u}_{\text{sup}}(t) &= -u_{\text{sup}}(t) + h + c_{\text{sup,int}} \sum_m \sigma(u_{\text{int},m}(t)) \\ &\quad + \sigma\left(\sum_l \sigma(u_{\text{tsk},l}(t))\right).\end{aligned} \tag{3.11}$$

For this setting, the intention node of one EB projects its output to a suppression node of a competing behavior. This node is boosted if the intention node is active, thus inhibiting the intention node of the competing behavior.

CoS nodes might listen to a varying amount of N peak detectors u_i^{pd}, which all signal a partial completion of the current behavior (for example, think of a visual search for an object with a varying amount of cues that the object has to match). To accommodate for this, the resting level of the CoS can be lowered for each expected completion signal given the activation of conditional nodes u_j^{con}, which requires each signal to be active before the CoS node pierces the detection threshold. This results in a modified equation

$$\begin{aligned}\tau\dot{u}_{\text{cos}}(t) &= -u_{\text{cos}}(t) + h \\ &\quad + c_{\text{cos}}\sigma(u_{\text{cos}}(t)) + c_{\text{cos,int}}\sigma(u_{\text{int}}(t)) \\ &\quad + \sum_{i=1}^{N} \sigma(u_i^{\text{pd}}(t)) - \sum_{j=1}^{N} \sigma(u_j^{\text{con}}(t)).\end{aligned} \tag{3.12}$$

Activating an intention node of a behavior does not guarantee that its execution will ever reach the condition of satisfaction. Think of starting a visual search for a specific object that is not present in a scene or moving an end-effector towards a target object and running into joint limits rendering it impossible to reach. The two node setup of intention and CoS nodes is not able to detect this failure of executed behavior. The setup can be complemented by a condition of dissatisfaction (CoD) node, which takes care of representing any failure in achieving a behavior's goal in time. The CoD node can be connected to any field that is able to detect the failure of a behavior. In the absence of such a detector, the CoD node can also be connected to a timer, counter, or any other source representing the discontinuation of motivation to execute the current behavior. CoD nodes follow the structure of Equations 3.9 and 3.12, but listen to different parts of the architecture to detect the failure of the behavior and may trigger a different set of behaviors for error recovery. While CoS nodes may listen to varying amounts of peak detectors and only become active if all of the peak detectors are in their active state, the CoD may become active as soon as a single peak detector signals a

failure. This can be achieved by removing the input from conditional nodes in Equation 3.12. Without external input, a CoD node thus resides below, but close to its detection threshold. A single peak detector becoming active is sufficient to push the CoD node over its threshold. On a more abstract level, this coupling structure implements the logical operator OR, while the CoS node implements a logical AND.

3.2.2 Change Detection and Matching

Behavioral preconditions may be based on internal representations, which are created at some point in time (e.g., through a perceptual behavior). Internal representations and their external equivalent may drift apart over time, either through changes in the real world or internal memory diffusion. In the context of scene representation, the spatial position of objects or characteristic features (for example, the orientation) may change, while working memory peaks representing these features drift, interfere with other representations, or dissolve completely. A crucial mechanism for maintaining the internal representation is the detection of differences between the expectation given by working memory and the current state of the observed scene, regardless of the cause of this discrepancy. In DFT, this is realized by comparing feature input with a working memory representation in a perceptual field [87]. If expectation represented in working memory and current input overlap sufficiently, inhibition prevents the emergence of a peak (see Figure 3.11, left side). If the inhibitory input affects a different field site, the excitation leads to a supra-threshold peak (see Figure 3.11, right side). The activation in working memory and perceptual fields may drive discrete *response* nodes explicating the decision of the three-layer setup, as used by Johnson and colleagues [86].

Here, I extend this change detection mechanism to a more generic match detector by allowing to replace the fixed working memory input by any expectation or prediction of the current input. I formalize the observable states of the mechanism through two accompanying nodes for the *match* and *no-match* conditions, which can be connected to the behavioral organization and offer CoS and CoD signals for the compared neural activation. The *match* field

$$\begin{aligned} \tau \dot{u}_{\text{mat}}(x,t) &= -u_{\text{mat}}(x,t) + h \\ &\quad +[w_{\text{mat}} * \sigma(u_{\text{mat}})](x,t) \\ &\quad +[w_{\text{mat,exc}} * \sigma(u_{\text{exc}})](x,t) \\ &\quad -[w_{\text{mat,inh}} * \sigma(u_{\text{inh}})](x,t) \end{aligned} \tag{3.13}$$

receives both excitatory and inhibitory input from fields u_{exc} and u_{inh}, re-

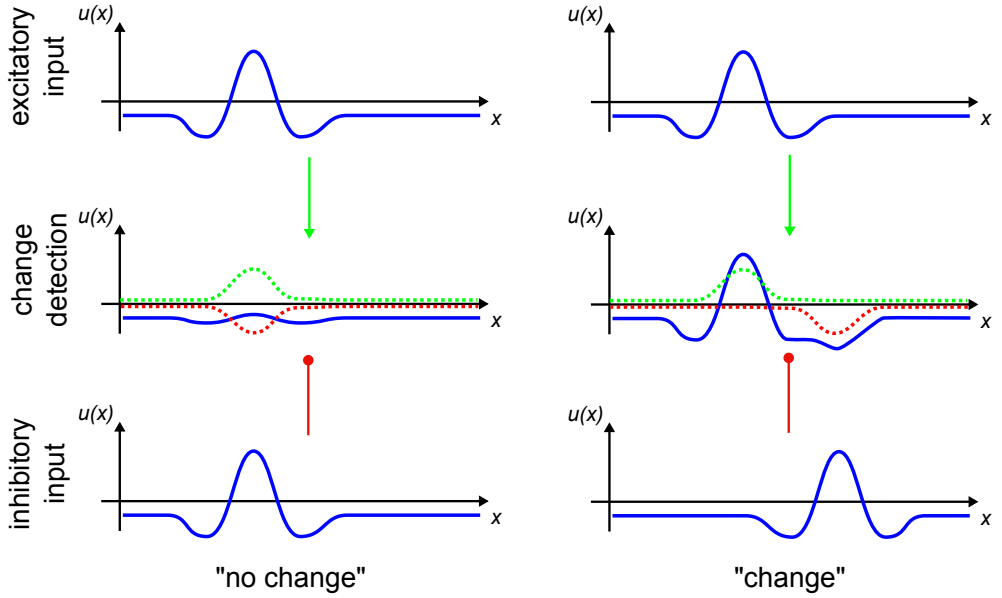

Figure 3.11: Change detection in DFs can be achieved by projecting both excitatory and inhibitory input into a change detection field. If the location of peaks in both inputs match, no peak is created in the third field. If peaks appear at different positions, the change detection field builds up a peak as well.

spectively.

The *no match* node

$$\begin{aligned}\tau \dot{u}_{\text{nom}}(t) &= -u_{\text{nom}}(t) + h \\ &\quad + c_{\text{nom,inh}}\sigma(u^{\text{pd}}_{\text{inh}}(t)) + c_{\text{nom,exc}}\sigma(u^{\text{pd}}_{\text{exc}}(t)) \\ &\quad + c_{\text{nom,mat}}\sigma(u^{\text{pd}}_{\text{mat}}(t))\end{aligned} \tag{3.14}$$

is activated through sufficient supra-threshold activity in the *match* field, measured by its peak detector $u^{\text{pd}}_{\text{mat}}$. The competing *match* node

$$\begin{aligned}\tau \dot{u}_{\text{mat}}(t) &= -u_{\text{mat}}(t) + h \\ &\quad + c_{\text{mat,inh}}\sigma(u^{\text{pd}}_{\text{inh}}(t)) + c_{\text{mat,exc}}\sigma(u^{\text{pd}}_{\text{exc}}(t)) \\ &\quad - c_{\text{mat,nom}}\sigma(u_{\text{nom}}(t))\end{aligned} \tag{3.15}$$

signals a match. Both nodes receive additional input from the peak detectors of the excitatory and inhibitory fields, $u^{\text{pd}}_{\text{exc}}$ and $u^{\text{pd}}_{\text{inh}}$. These inputs are strong enough to drive the *match* node above the detection threshold. Through inhibition, the *no match* node suppresses the *match* node. Figure 3.12 shows a graphical representation of the connectivity. Note that the activation of

these nodes does not explicitly state the degree of match, although a certain tolerance of expectation can be expressed in the width of the peak in the inhibitory field.

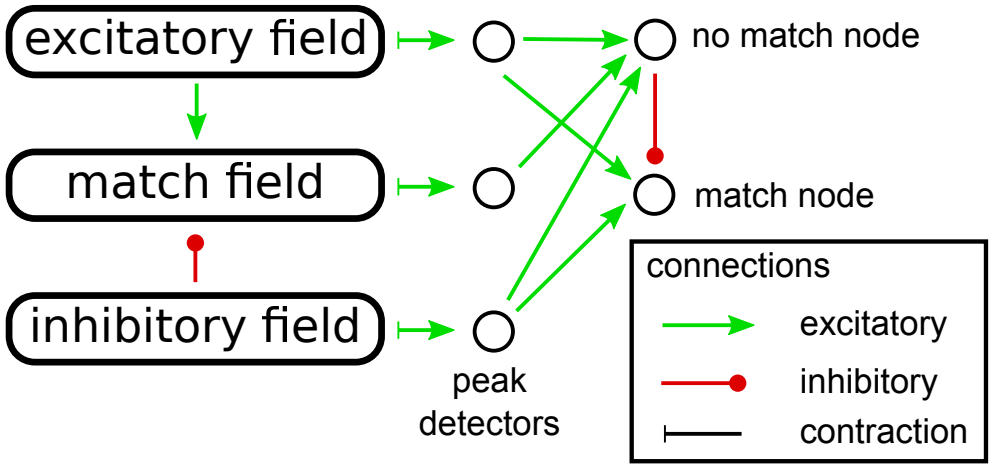

Figure 3.12: Match detection between input and expectation is expressed by two nodes signaling the match and no match conditions brought about by the coupling to peak detectors of all involved fields.

3.2.3 Reference Frame Transformations

Spatial information exists in different reference frames. Spatial coordinates may be expressed as position on the retina, a combination of eye and head position called gaze, body-centered coordinates, allocentric coordinates (an arbitrary, but fixed coordinate system that is independent of body pose), and motor configurations (e.g., the joint configuration to place the hand at this spatial position), among others. In DFT, reference frame transformations are implemented with transformation fields with a specific coupling structure. Sandamirskaya and colleagues [152] present a field architecture that allows to transform retinal positions to body-centered coordinates and vice versa, taking into account the current gaze configuration as a parameter of the transformation. Here, expansions are used to create a dynamic link of gaze and spatial positions. A special diagonal contraction is used to read out the transformed spatial position. More complex transformations require complex weight matrices that relate the two coordinate frames to each other. To learn more complex transformations, a framework implementing autonomous learning is required [151].

Zibner and Faubel [195] describe an algorithmic shortcut for transforming a camera image into a scene-centered representation and vice versa. This method requires two matrices that describe the internal and external transformations that map a point in the world onto a camera pixel. The internal matrix describes intrinsic camera properties, which are parameters of a pinhole camera model. The pinhole camera model assumes that light reflected from an object's surface passes a singular pinhole before hitting an image plane placed behind the pinhole. Resulting parameters are the focal distance between pinhole and image plane and the two-dimensional position of the pinhole projected onto the image plane. The external transformation summarizes the translations and rotations applied to the coordinate frame of the image plane in relation to a world coordinate frame. This matrix is defined through the position and orientation of the camera in space, which can be extracted from the configuration of a robot's position and state of degrees of freedom using forward kinematics. The combination of internal and external transformation map a three-dimensional scene observed by a camera onto an allocentric reference frame that is invariant under camera movement. Note that this transformation suffers perspective distortions whose amplitude depend on the position relative to the camera.

3.3 Assembly and Simulation of DFT Architectures

DFT architectures are assembled from elementary building blocks and larger groups such as the match detector described above. Large systems of differential equations are the result, with each building block having an impact on other blocks it connects to, be it directly or through transition. Differential equations pose two challenges for computer science. First, how can these architectures be assembled and parameterized, and second, how can these equations be solved to yield experimental results?

3.3.1 Numerical Approximation of Dynamics

DFT architectures are continuously linked to sensory input and are expected to produce motor output at a steady rate. Thus, one cannot analytically solve the differential equation. Instead, an approximate, iterative solution is calculated that uses a small time step and takes into account the current state of sensory input. Approximation introduces errors in the numerics, with error size increasing with size of the chosen time step. This results in a trade-off between computation time and error size. Having a small time

step decreases error size, but the dynamics have to be evaluated more often, increasing the computational demands. Choosing a larger time step increases error size, but at the same time fewer evaluations require less computational power.

There exist several approaches to numerical approximation, for example Runge-Kutta methods, and forward and backward Euler methods, each suited for certain categories of differential equations with varying amounts of computational complexity by requiring a number of evaluations of the differential equation per approximation and memory demands by using several buffered recent approximations fur the current approximation. The dynamical systems used in DFT are characterized by stable fixpoints, which counteract the numerical errors introduced by approximation. With this in mind, the forward Euler method is sufficient and requires only a single evaluation of the differential equation using the current approximation, which keeps computational and memory demands low. For a time step Δt, a differential equation of form

$$\tau \dot{u}(t) = -u(t) + h + c\sigma(u(t)) + s(t), \tag{3.16}$$

which resembles the activation of dynamic nodes in Equation 3.3 is approximated as

$$u(t + \Delta t) \approx u(t) + \frac{\Delta t}{\tau}(-u(t) + h + c\sigma(u(t)) + s(t)) \tag{3.17}$$

to determine activation u at the next time step $t + \Delta t$, given the current approximation u and the change during the time step Δt, estimated from u, the constant h, and input s at time step t. Approximation errors increase proportional to the size of the time steps, with larger step sizes leading to numerical oscillations and instabilities, which have a significant impact on the behavior of the approximated neural dynamics (see Figure 3.13 for examples on how step size affects relaxation and Figure 3.14 for an example of how errors affect overall behavior of a non-linear dynamical system).

Iterative approximations require a time step to calculate the next state from the current state. Measuring the elapsed time between two consecutive approximations is one way of providing the required timing. The computational load of a processor may however lead to fluctuating time measurements, as the same computation necessary for the approximation requires various amounts of time. A fluctuation of computation time translates into fluctuations of the approximation error. To keep these fluctuations to a minimum, a minimal step size can be defined. If computation of one approximation takes less time than defined by this minimum, the remaining

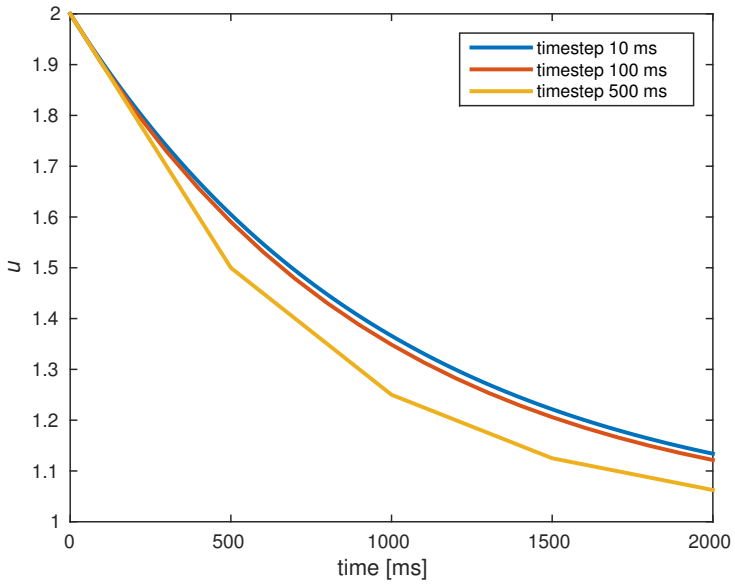

Figure 3.13: This figure shows the relaxation of the dynamical system $\tau\dot{u}(t) = -u(t) - 1 + 2\sigma(u(t))$ for $\tau = 1\,\mathrm{s}$ and a starting value of $u(0) = 2$ using an iterative approximation with different step sizes. With increases in step size, the approximated solution deviates more from the exponential decay of the dynamical system. This distorts the relaxation rate defined by τ, which is an issue if multiple dynamical system are coordinated through time.

time is filled with a pause in computation, freeing computational power for other resources. The approximation is then calculated using the minimal step size. If, however, computation takes longer than the minimal step size (which should be the exception), the step size is adapted to the elapsed time for a single approximation.

3.3.2 Efficient Output, Lateral Interactions, and Projections

Calculating the lateral interaction and projections between DFs requires to evaluate a sigmoid function (see Equation 3.2) at every sampling point of the discretized field activation. This requires an evaluation of the exponential function for each sampling point and a floating point parameter. Approximations of the sigmoid function save computational time by avoiding exponentiation. A computationally fast approximation of the sigmoid func-

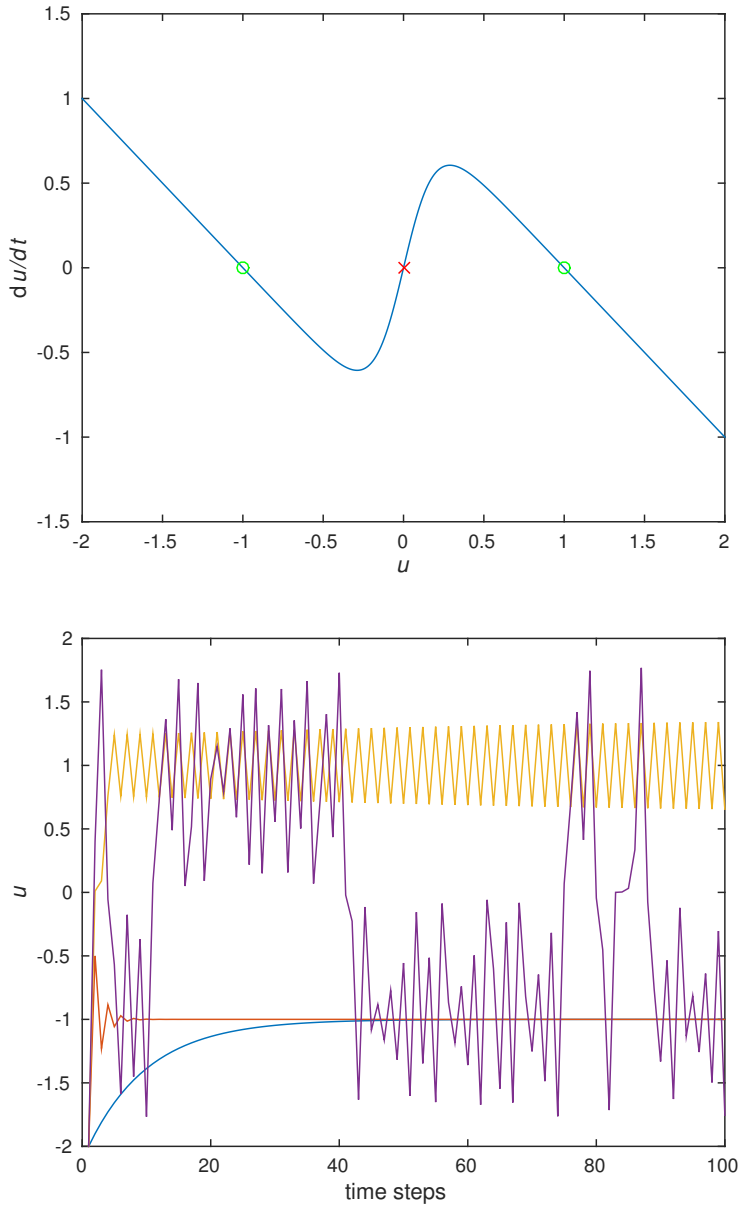

Figure 3.14: The top plot shows a phase plot of a non-linear dynamical system defined in Figure 3.13 with two attractors marked with green circles and a repellor marked with a red x. The bottom plot shows approximations of the evolution of u for a start value of -1 and different step sizes. Sufficiently small step sizes (blue line) reflect the exponential relaxation to the attractor at -1. With increasing step size, approximation errors induce oscillations. The error manifests as damped oscillation around the attractor at -1 (red line), initial switch to a different attractor and subsequent undamped oscillation around this attractor (yellow), and oscillations between the attractor states (purple line). Even larger step sizes lead to numerical instability. Approximation errors thus have a direct impact on the behavior of neural dynamics.

tion is

$$\sigma_{\beta,\text{fast}}(u(\boldsymbol{x},t)) = 0.5\left(1 + \frac{\beta u(\boldsymbol{x},t)}{1+\beta|u(\boldsymbol{x},t)|}\right), \tag{3.18}$$

which uses the computationally less demanding absolute function instead of an exponentiation.

Dynamic fields use kernels to determine how supra-threshold activation affects the dynamics of the lateral layer and other DFs receiving input from this layer. Excitatory connectivity covers a limited region of the field dimensions, while inhibitory connections may span the whole field (global inhibition). The weight matrix of lateral and projection kernels is identical for every field site. Through this homogeneity, calculating lateral interactions and projections can be considered as filtering the sigmoided field activation with the kernel. Thus, I use the convolution to determine interactions.

Convolutions are computationally demanding operations, especially for high field dimensionality and large kernel sizes. Each differential equation describing a dynamic field contains at least one convolution, the lateral interaction, with additional convolutions originating in projections from other fields. This poses a challenge for numerical approximation of the overall architecture, as multiple convolutions have to be calculated in each time step of the numerical simulation. However, the structure of the interaction kernels of DFs offers some simplifications of the convolution that keep the computational load low.

First, global inhibition can be calculated separately from lateral interactions, which reduces the kernel size. This calculation requires an integral of supra-threshold activation over $\boldsymbol{x} = (x_1, \ldots, x_n)$, weighted with the strength of global inhibition, $c_{\text{gi}} \leq 0$,

$$f_{\text{gi}}(\boldsymbol{x},t) = c_{\text{gi}} \int \cdots \int \sigma(u(\boldsymbol{x},t))dx_1 \cdots dx_n. \tag{3.19}$$

Second, the remaining interaction kernels are made up of Gaussians. For multi-dimensional fields, the kernel can be separated into one-dimensional Gaussians, as their tensor product again yields the kernel. The convolution for an n-dimensional field can be replaced with n one-dimensional convolutions with the one-dimensional components of the kernel, with a subsequent summation of convolution results.

Third, convolutions can be calculated in frequency space after applying the discrete Fourier transformation to sigmoided field activation and the kernel. In this space, convolution is a point-wise complex-valued multiplication of the two operands. An inverse Fourier transformation brings the product

back to the original space. I use efficient algorithms for Fourier transformations (coined fast Fourier transformations), which further benefit from Fourier transformations being separable for multi-dimensional fields and kernels not changing over time.

3.3.3 Parameterization and Tuning

Each elementary building block, be it a DF or a dynamic node, adds a set of parameters to the resulting differential equation of each DFT architecture: resting level, steepness of the threshold function, time scale, strength of noise, and in the case of fields, strength of global inhibition and lateral kernel, as well as sizes of the excitatory and inhibitory component for each dimension. Each connection between elementary building blocks adds additional parameters, such as connection weights. Numerical approximation adds more parameters, such as amount of sampling points discretizing the continuous field metrics, cut-offs for kernel matrices to keep them small for convolutions, re-sampling between neural layers, choice of Fourier transformation algorithms and border padding for convolutions, and sampling size of time steps, among others.

The sheer amount of parameters forms a contrast to the few regimes DFs and dynamic nodes operate in. In fact, the input into a dynamic field or node as well as the desired operational regime yield restrictions on the intervals of parameters. For example, the range of resting level of a field or node is determined by the inputs these receive and the level at which field or node undergo the detection decision. If n sources project sigmoided activation to a field or node, the resting level may be chosen as -0.5 to trigger the detection decision if any of the sources is above threshold. If the resting level is instead chosen to be $-0.5+(1-n)$, all inputs must be above threshold for a detection decision. Note that these examples correspond to the logical expressions OR and AND, respectively (see Section 3.2.1). As another example, consider lateral kernel widths and amplitude for fields operating in a working memory regime. The widths of the kernel are restricted by the width of the input this field receives, as there is a kernel size that maximizes interaction for a given input size (see also Figure 3.8). Kernel amplitude depends on the field's resting level, as peaks have to be sustained after localized input has vanished. Thus, lateral interaction has to be larger than the resting level to keep the field above threshold and sustain working memory.

To assist with the assembly of DFT architectures, parameters can be set to good default values wherever possible. In addition, easy access to parameters during assembly as well as at runtime further support the modeler in assembling an architecture.

3.3.4 Groups of Elementary Building Blocks

Elementary building blocks may form groups providing higher-level functionality, such as change or match detection and behavioral organization. Groups assure that the tuning necessary for a group's functionality is encapsulated; other parts of an architecture influence a group only through clearly defined interfaces, that is, projecting input to dedicated fields and nodes of the group. A software implementation of groups may draw upon the principles of object-oriented programming, especially encapsulation.

3.3.5 The DFT Software Framework cedar

The previously mentioned aspects of DFT architectures are captured in the open-source C++ software framework *cedar*[1] [103], which wraps architecture assembly and simulation in a graphical user interface (see Figure 3.15). See Appendix D for details.

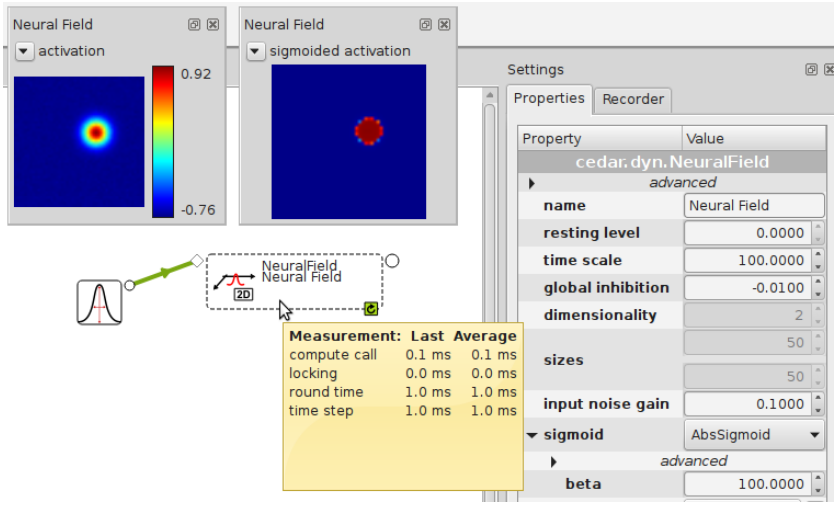

Figure 3.15: A screenshot of *cedar* showing part of the graphical user interface. Two components (Gaussian input, two-dimensional DF) are connected. Plots of all components, as well as parameters and additional statistics are accessible during simulation of the dynamics.

[1] http://cedar.ini.rub.de/

3.4 Robotic Platforms

Within the software framework *cedar*, the neuro-dynamic architectures presented in this work are connected to the robotic platforms CAREN, CoRA, and NAO. Visual sensors are the primary source of perception, while robotic actuators such as head joints, arms, and hands execute movements generated by the architectures.

3.4.1 CAREN

The cognitive autonomous robot for embodiment and neural dynamics, or CAREN for short, is a custom-made anthropomorphic, stationary robotic platform (see Figure 3.16). Its trunk is mounted on a table top. The trunk supports a seven degrees of freedom (DoF) Kuka light weight robot (LWR4) attached to a Schunk PR110 rotary module, ending in a seven DoF Schunk Dextrous Hand (SDH), and a two DoF camera head consisting of a Schunk PW90 (a pan-tilt unit) and exchangeable visual sensors. Sensor mounts are either a Microsoft Kinect or a rack with three Sony RGB cameras—two high-resolution Sony XCD-SX90CR (up to 1280x960 px) and one high-frequency Sony XCD-V60CR (up to 90 Hz). All hardware components of CAREN are accessible from within the *cedar* software framework (see Appendix D). The table top in front of CAREN serves as the visual scene. Its uniform white color puts low demand on the perceptual discrimination between background and foreground.

3.4.2 CoRA

The cooperative robotic assistant (CoRA) is the predecessor of CAREN, featuring an eight DoF arm made up of Schunk PowerCube rotary modules and ending in a two-finger gripper, and a pan-tilt camera head with two Sony cameras. All hardware components of CoRA are accessible in *cedar*. CoRA is also mounted on a table top, using the same setting as CAREN.

3.4.3 NAO

NAO is a humanoid robotic platform made by Aldebaran[2] (see Figure 3.17). The whole body features 25 DoFs, covering two legs, two arms with hands, and a head. The head features two RGB cameras, arranged vertically. Additional sensors are available, but not relevant for this work. All hardware components of NAO are accessible from within the *cedar* software framework.

[2] www.aldebaran.com

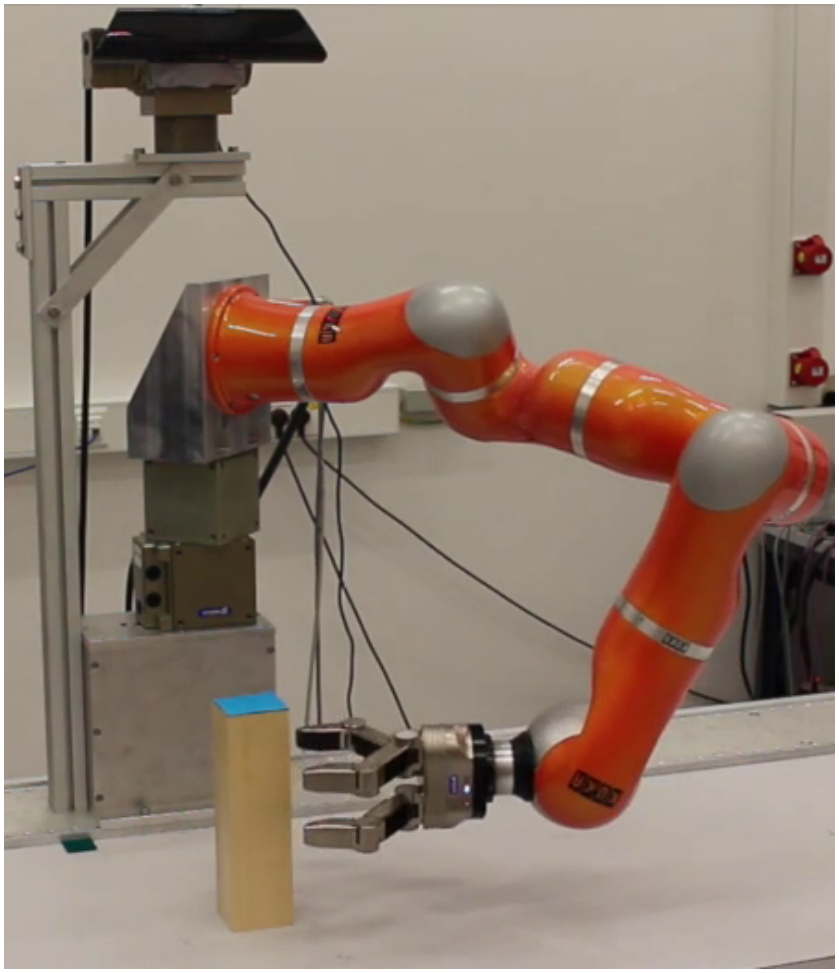

Figure 3.16: The CAREN platform mounted on a table top. In this figure, a Microsoft Kinect is mounted on the head joints.

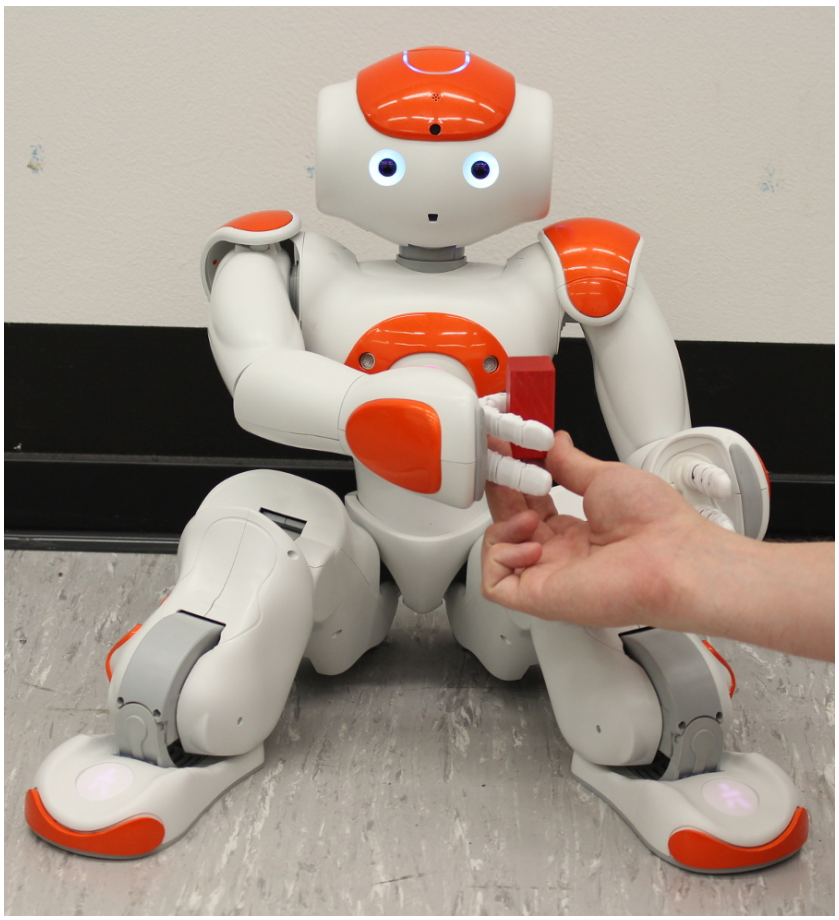

Figure 3.17: The NAO platform reaching for a colored object.

Chapter 4

Scene Representation

In this chapter I present my model of scene representation. It uses the principles and building blocks of DFT presented in Chapter 3 and is constrained by the behavioral signatures discussed in Chapter 2. Central concepts are an attentional bottleneck that introduces sequentiality in an otherwise parallel processing pathway, a saliency extraction pathway that can be influenced in parallel by top-down cues, a working memory representation of the scene that accumulates feature estimates of inspected objects, a match detector mechanism that compares estimates or expectations with the currently selected candidate, and a query mechanism that re-instantiates accumulated scene knowledge given cues.

I split up scene representation into three behavioral components: (1) creating internal representations of relevant objects by means of *visual exploration*; (2) re-evaluating and updating internal representations in accordance with changes in the observed scene by continuously *maintaining* the resulting representations; (3) re-instantiating attention to memorized objects to access their properties by applying task-specific cues to a *query* or finding objects not contained in memory via *visual search*. The architecture is evaluated on the robotic platform CAREN, covering the behaviors visual exploration, maintenance, and visual search/query.

The following description is an integration and refinement of several aspects previously investigated in publications. A core building block of this architecture are three-dimensional DFs, whose properties and their suitability in the context of scene representation are evaluated in my master thesis [194]. The fundamental structure of scene representation in the context of robotics is covered in [195, 196], providing solutions for the challenges arising out of embodiment, for example using cameras as input and dealing with a limited field of view and a moving head. Maintenance in the form of multi-item tracking [197] and the use of multiple feature channels in the

context of queries [198] are further discussed in separate publications. Earlier publications use the robotic platform CoRA as means of evaluation.

4.1 Architecture

In this section, I first introduce the fields and projections that make up the scene representation architecture for visual exploration, maintenance and query, before moving on to a closer description of the organization of these three behaviors and the emergent autonomy. I refine the previously published version of this architecture by adding additional feed-forward feature channels and top-down feature cue feedback. I leave out the reference frame transformations between different levels of the architecture, since I do not cover overt attention shifts (i.e., head movements) in this thesis and focus mainly on the autonomy of the behaviors. For the sake of clarity, I focus on a single feature channel, color, in the following descriptions and figures whenever possible, although scene representation assumes a richer description of object features (such as size and a label identifying objects [198]). The full architecture uses three distinct feature channels: color, size, and aspect-ratio.

4.1.1 Camera Input

My scene representation architecture processes real-time camera input capturing a scene in front of the robot. The subsequent pathways assume a three-channel RGB image covering a two-dimensional allocentric reference frame such as the table plane in front of the stationary robots CAREN and CoRA. I produce this input in three different ways: (1) by using a Sony firewire camera and applying a perspective transformation to the camera image, knowing the internal camera matrix and the external transformation in relation to the table coordinate frame (see Zibner and colleagues [196] for details); (2) by using a Kinect RGBD sensor and transforming the point cloud data into a bird's eye projection onto the table plane (with the help of the software framework Point Cloud Library (PCL [146]), specialized on point cloud processing; see also Knips and colleagues [92] for details); (3) creating artificial images and assuming that these images are already in the right coordinate frame (see experimental setup in Section 4.2.5 for details). The motivation behind being able to switch the sensor driving my architecture is threefold: analyzing generalization across multiple sensor surfaces without having to alter the tuning of the architecture; demonstrating robustness to noise when using highly volatile sensor input such as the one obtained from

the Kinect; being able to compare my architecture to models that use ideal (i.e., noise-free and normalized) input and evaluating conclusions drawn from this paradigm in comparison to real sensor input.

4.1.2 Feed-forward Feature Maps

Three feed-forward features are used in my architecture: color, size, and aspect ratio. For each feature, a feed-forward *early space-feature field* uses the detection decision to ensure that only feature values at salient positions are represented. This is a continuous, single-scale variation of the feature maps of Itti and colleagues' feature map pyramid [84]. I implement the normalization operation present in the feature maps and subsequent processing stages by applying global inhibition across the spatial dimensions for slices taken along the feature dimension both in the input layer s_{esf}, as well as the dynamic fields representing these inputs. This inhibition is implemented by a chain of operations, consisting of a *contraction* onto the feature dimension f, a weighting with a one-dimensional inhibitory kernel w_{inh} and a subsequent *expansion* back to the three-dimensional space,

$$s_{\text{inh}}(x, y, f, t) = \int w_{\text{inh}}(f - f') \left(\iint s_{\text{esf}}(x', y', f', t) dx' dy' \right) df'. \tag{4.1}$$

If multiple peaks represent similar feature values at different spatial positions, inhibition is stronger and thus decreases peak strength of peaks representing similar feature values. This results in a competitive advantage for peaks in feature regions with fewer peaks. See Figure 4.1 for an example.

The input for the three different feature channels is created as follows. For color, the camera image is first translated into the HSV[1] color space. The hue channel $\text{hue}(x, y, t)$ is split up into $R \in \mathbb{N}^+$ maps covering fixed intervals of the full hue range $[0, H)$. Only regions of saturation above threshold θ in the saturation channel $\text{sat}(x, y, t)$ contain non-zero entries. The resulting stack of maps,

$$s^{\text{col}}_{\text{bup}}(x, y, r, t) = \begin{cases} c_{\text{sat}} \sigma_\theta(\text{sat}(x, y, t)) & \text{if } \lfloor \frac{\text{hue}(x,y,t)}{H} R \rfloor = r \\ 0 & \text{else,} \end{cases} \tag{4.2}$$

is input to the three-dimensional early space-color field.

Size is determined by a battery of center-surround filters of different size applied to the thresholded saturation, leading to a stack of size maps,

$$\begin{aligned} s^{\text{siz}}_{\text{bup}}(x, y, k, t) \;&=\; [w_k^+ * \sigma_\beta(\text{sat})](x, y, t) \\ &\quad -[w_k^- * \sigma_\beta(\text{sat})](x, y, t), \end{aligned} \tag{4.3}$$

[1] **h**ue, **s**aturation, **v**alue

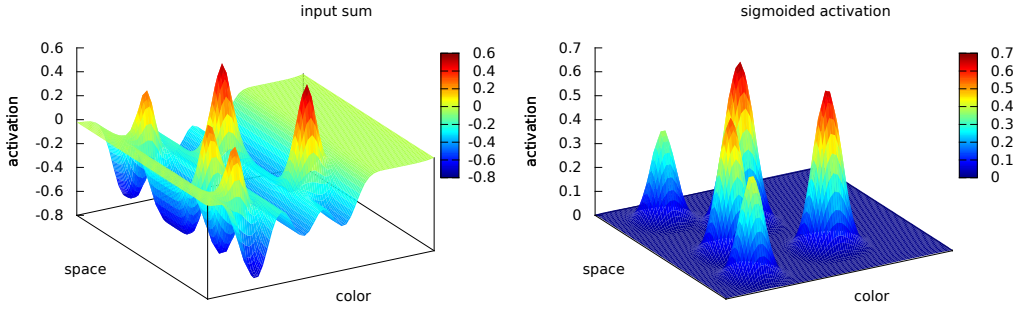

Figure 4.1: Five uniform Gaussian inputs are placed along the two metrical dimensions space and color on the left. Input strength is decreased relative to the amount of activation for each color through global inhibition along the spatial dimension in the form of negative ridges. The sigmoided activation of a dynamic field representing salient colors in space is stronger for colors that appear less often in the input array.

which is represented in a three-dimensional early space-size field. w_k^+ are the excitatory filter modes representing the center response, while w_k^- are the inhibitory filter modes representing the off-center response.

Aspect ratio is extracted from ellipsoids fitted to clusters in the camera image given by the detection decision of a field receiving the thresholded saturation channel. Clusters are first transformed into contours [168], before fitting minimal area rectangles to these contours. This fit is based on the rotating calipers approach [177]. The resulting rectangles define the aspect ratio by the ratio of major axis length, l_{maj} to minor axis length, l_{min}. A three-dimensional activation pattern,

$$s_{\text{bup}}^{\text{rat}}(x, y, a, t) = \begin{cases} 1 & \text{if sat}(x, y, t) \text{ has rectangle and } \frac{l_{\text{maj}}}{l_{\text{min}}} = a \\ 0 & \text{else,} \end{cases} \tag{4.4}$$

is generated from the spatial arrangement of rectangle ratios. Aspect ratio in the interval $[1, A]$, with A signifying the largest represented aspect ratio, is represented in a three-dimensional early space-aspect-ratio field.

All *early space-feature fields* (see Figure 4.2) receive their respective stack of maps as input. In addition, a top-down feature cue $u_{\text{cue}}^{\text{F}}$ might highlight a certain range of the represented feature space, with F being a placeholder for the feature channel. Additional top-down cues, for example spatial biases, are not discussed here, but follow the same structure as the feature cue. The

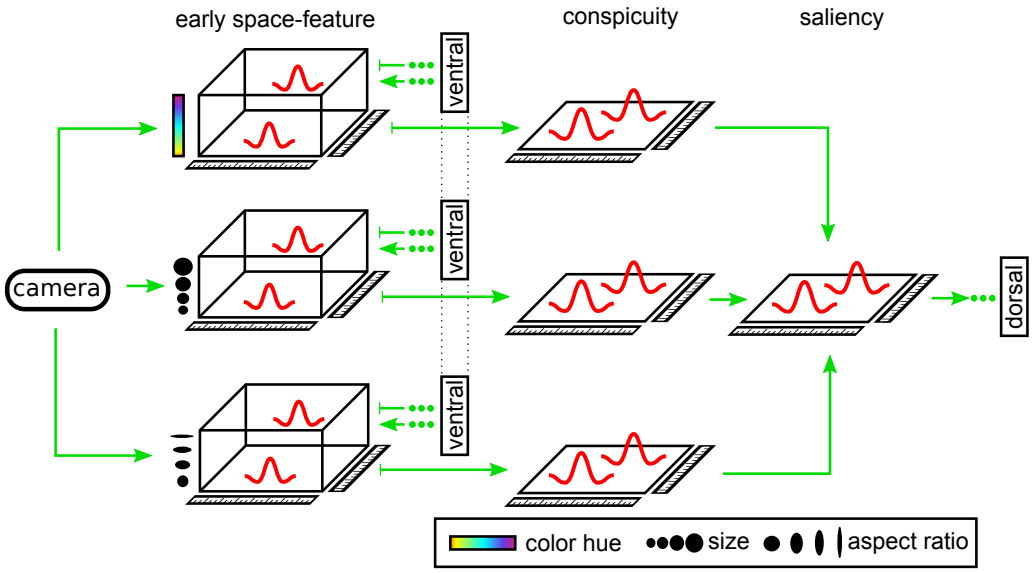

Figure 4.2: The bottom-up pathway first enters *early space-feature fields*. Summed over feature, their output enters *conspicuity fields*. A *saliency field* integrates the contributions of all *conspicuity fields*. Summed over space and weighted with the current attentional focus, features are locally extracted and forwarded to the scene representation. Top-down feature cues may influence the level of saliency along each feature channel.

resulting field equation,

$$\begin{aligned} \tau \dot{u}^{\mathrm{F}}_{\mathrm{esf}}(x,y,f,t) \;=\; & -u^{\mathrm{F}}_{\mathrm{esf}}(x,y,f,t) + h \\ & + [w^{\mathrm{F}}_{\mathrm{esf}} * \sigma(u^{\mathrm{F}}_{\mathrm{esf}})](x,y,f,t) \\ & + s^{\mathrm{F}}_{\mathrm{bup}}(x,y,f,t) \\ & + [w^{\mathrm{F}}_{\mathrm{esf,cue}} * \sigma(u^{\mathrm{F}}_{\mathrm{cue}})](f,t), \end{aligned} \tag{4.5}$$

combines the feed-forward input and the top-down influence. Global inhibition in the interaction kernel $w^{\mathrm{F}}_{\mathrm{esf}}$ implements a normalization operator. The global inhibition stretches along each slice of spatial dimensions of the *early space-feature fields* (compare Equation 4.1). If there are multiple peaks coding for similar metrical values in multiple spatial positions, inhibition decreases their contribution to the overall saliency.

4.1.3 Saliency

The connections from camera image to saliency resemble the dorsal pathway of human visual processing, containing spatial estimates of object hypotheses,

but loosing the feature description.

An integral along the feature dimension of each *early space-feature field* enters a conspicuity field over space for this feature channel,

$$\begin{aligned}\tau\dot{u}^{\mathrm{F}}_{\mathrm{con}}(x,y,t) &= -u^{\mathrm{F}}_{\mathrm{con}}(x,y,t)+h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{con}} * \sigma(u^{\mathrm{F}}_{\mathrm{con}})](x,y,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{con,esf}} * \int \sigma(u^{\mathrm{F}}_{\mathrm{esf}}(x,y,f,t))df](x,y,t) \\ &\quad +c_{\mathrm{con,cue}}\sigma(u^{\mathrm{F}}_{\mathrm{cue}}(t)),\end{aligned} \tag{4.6}$$

which operates in a multi-peak regime and restricts the input further through its global inhibition and sigmoided output function (see Figure 4.2). The output of each conspicuity field then enters the saliency field,

$$\begin{aligned}\tau\dot{u}_{\mathrm{sal}}(x,y,t) &= -u_{\mathrm{sal}}(x,y,t)+h \\ &\quad +[w_{\mathrm{sal}} * \sigma(u_{\mathrm{sal}})](x,y,t) \\ &\quad +\sum_{\mathrm{F}}[w_{\mathrm{sal,con}} * \sigma(u^{\mathrm{F}}_{\mathrm{con}})](x,y,t),\end{aligned} \tag{4.7}$$

which integrates their contributions. Saliency is given by peak strength and width (see Section 3.1.4). The resting level of u_{sal} is chosen to allow for peaks at locations at which some of the *early space-feature fields* do not pierce the detection threshold, as long as there is enough evidence for object presence given by the remaining ones. With this in mind, saliency is strongest at locations that contain a unique set of features or match well with the current top-down feature cues. Saliency at one location is lower if feature values are similar to those at other locations. Due to global inhibition, feature values may not be strong enough to push the *early space-feature fields* through the detection threshold at all.

4.1.4 Feature Extraction

Feature extraction represents the ventral pathway of human visual processing, since it contains only feature descriptions and no explicit spatial information of their origin.

A second pathway splits from the three-dimensional fields over space and feature and represents the feature description of an area of the input, modulated by top-down feedback specifying the attentional focus. Extracted features are represented in one-dimensional *feature extraction fields* (e.g., a

color field, see Figure 4.3, D),

$$\begin{aligned}\tau\dot{u}^{\mathrm{F}}_{\mathrm{fex}}(f,t) &= -u^{\mathrm{F}}_{\mathrm{fex}}(f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{fex}} * \sigma(u^{\mathrm{F}}_{\mathrm{fex}})](f,t) \\ &\quad + \iint \sigma(u_{\mathrm{atn}}(x,y,t))\sigma(u^{\mathrm{F}}_{\mathrm{esf}}(x,y,f,t))dxdy.\end{aligned} \tag{4.8}$$

The sigmoided output of the attention field u_{atn} feeds back to the feature extraction, defining the local region at which feature values are extracted. This results in a feature description of the currently focused spatial location in all *feature extraction fields* $u^{\mathrm{F}}_{\mathrm{fex}}$. The feature extraction fields are one part of a group of fields implementing feature match detection (see Section 3.2.2 for further details).

4.1.5 Attention, Working Memory, and Inhibition of Return

Figure 4.3 shows an overview of the fields and couplings of the part of the architecture following the bottom-up processing depicted in Figure 4.2. The *attention field*,

$$\begin{aligned}\tau\dot{u}_{\mathrm{atn}}(x,y,t) &= -u_{\mathrm{atn}}(x,y,t) + h \\ &\quad +[w_{\mathrm{atn}} * \sigma(u_{\mathrm{atn}})](x,y,t) \\ &\quad +[w_{\mathrm{atn,sal}} * \sigma(u_{\mathrm{sal}})](x,y,t) \\ &\quad +[w_{\mathrm{atn,meb}} * \sigma(u_{\mathrm{meb}})](x,y,t) \\ &\quad -[w_{\mathrm{atn,lwm}} * \sigma(u_{\mathrm{lwm}})](x,y,t) \\ &\quad +c_{\mathrm{atn,iio}}\sigma(u^{\mathrm{io}}_{\mathrm{int}}(t)) + c_{\mathrm{atn,iqo}}\sigma(u^{\mathrm{qo}}_{\mathrm{int}}(t)),\end{aligned} \tag{4.9}$$

is a central component of this architecture, which has incoming and outgoing connections to several other fields. It receives input from the bottom-up saliency extraction u_{sal}, an excitatory spatial bias from ongoing queries in the *space-feature query fields* $u^{\mathrm{F}}_{\mathrm{sfq}}$, which is integrated in a memory bias field u_{meb}, and an inhibitory spatial bias from the *looking working memory* u_{lwm}. Due to strong global inhibition in its lateral interaction kernel, the *attention field* performs single-peak selection decisions on its excitatory inputs, effectively bringing a single location into the attentional foreground of the architecture. Selection decisions are triggered by resting level boosts originating in intention nodes of the behavioral organization for *exploration* ($u^{\mathrm{io}}_{\mathrm{int}}$ denoting the intention to inspect an object) and *query* ($u^{\mathrm{qo}}_{\mathrm{int}}$ denoting to pick a single candidate object for query). The *attention field*'s recent activation

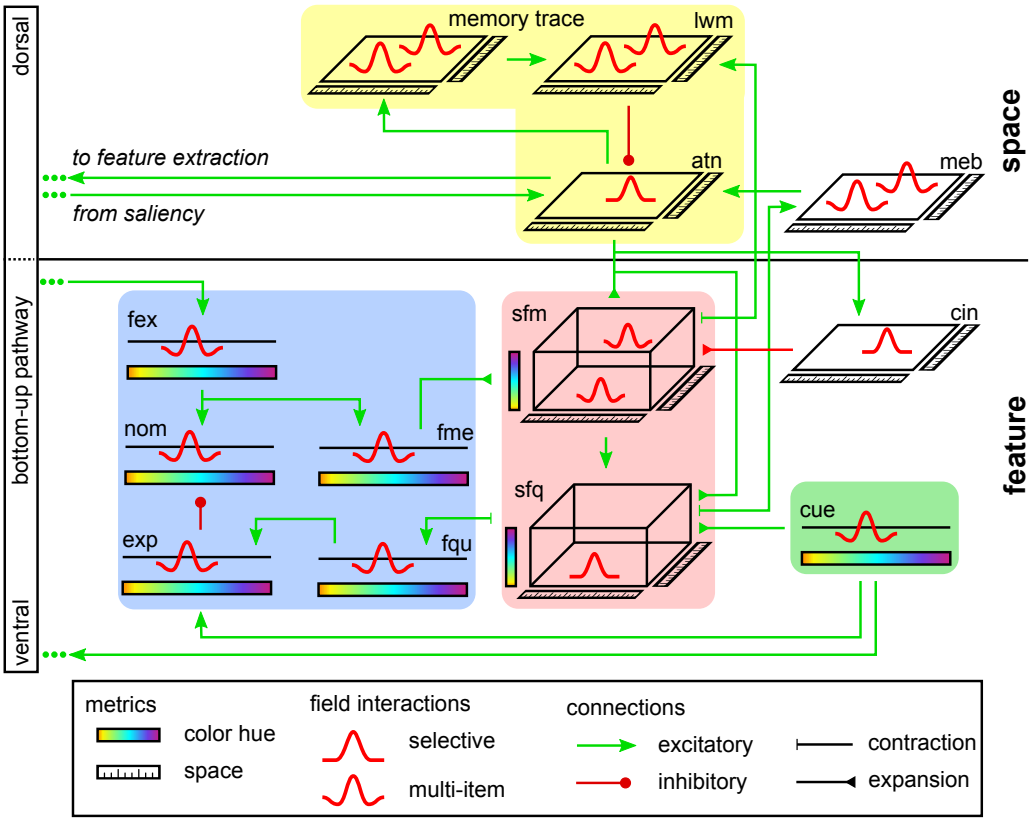

Figure 4.3: The fields of the scene representation architecture can be divided into a spatial pathway (yellow) and a feature pathway (blue), which are combined in working memory (red). External cues may enter the architecture (green). For each feature channel, there exists a copy of all fields in the lower part of the figure, marked "feature". Nodes realizing the behavioral organization of the architecture and connections to and from them are not shown here.

history is carried along in a memory trace [196],

$$\begin{aligned} \tau \dot{p}_{\text{atn}}(x,y,t) &= \lambda(x,y,t)(-p_{\text{atn}}(x,y,t) + \sigma(u_{\text{atn}}(x,y,t))) \\ \text{with } \lambda(x,y,t) &= \lambda_{\text{bui}}\sigma(u_{\text{atn}}(x,y,t)) \\ &\quad +\lambda_{\text{dec}}(1-\sigma(u_{\text{atn}}(x,y,t))), \end{aligned} \tag{4.10}$$

with a fast build-up rate λ_{bui} and a slower decay rate λ_{dec}. It adapts quickly to peaks and keeps inspected locations in memory over several attentional fixations. Note that projections from the memory trace to other DFs do not use a sigmoided output function. Its influence on other fields thus is of a graded nature.

The memory trace of the *attention field* and a spatial readout of the space-feature working memory (consisting of a combination of all *space-feature working memory fields*) are input to a *looking working memory field*,

$$\begin{aligned} \tau \dot{u}_{\text{lwm}}(x,y,t) &= -u_{\text{lwm}}(x,y,t) + h \\ &\quad +[w_{\text{lwm}} * \sigma(u_{\text{lwm}})](x,y,t) \\ &\quad +[w_{\text{lwm,pre}} * p_{\text{atn}}](x,y,t) \\ &\quad +\sum_{\text{F}}[w^{\text{F}}_{\text{lwm,sfm}} * s^{\text{F}}_{\text{ctr}}](x,y,t) \\ \text{with } s^{\text{F}}_{\text{ctr}}(x,y,t) &= \int \sigma(u^{\text{F}}_{\text{sfm}}(x,y,f,t))df, \end{aligned} \tag{4.11}$$

which contains transient peaks of recently inspected and memorized locations. In addition to the input from the memory trace of the attention field, the looking working memory also receives contracted input from all space-feature working memory fields, effectively making working memory peaks of locations with active space-feature links more persistent. The output of this field projects inhibition back to the *attention field*, thus decreasing the saliency of already inspected locations. This resembles a top-down inhibition of return mechanism (see the model of Itti and colleagues [84]).

A selection decision in the attention field fills the *feature extraction field*s with feature estimates describing the currently selected spatial location (see Equation 4.8). The output of each *feature field* passes through an associated *feature memorization field*,

$$\begin{aligned} \tau \dot{u}^{\text{F}}_{\text{fme}}(f,t) &= -u^{\text{F}}_{\text{fme}}(f,t) + h \\ &\quad +[w^{\text{F}}_{\text{fme}} * \sigma(u^{\text{F}}_{\text{fme}})](f,t) \\ &\quad +[w^{\text{F}}_{\text{fme,fex}} * \sigma(u^{\text{F}}_{\text{fex}})](f,t) \\ &\quad -c^{\text{F}}_{\text{mem,hme}} u^{\text{F}}_{\text{hme}}(t). \end{aligned} \tag{4.12}$$

These fields are inhibited if there is a current entry in working memory at this location, which is expressed through the activation of a *has memory node* $u^{\mathrm{F}}_{\mathrm{hme}}$. The *has memory node* is active whenever the contraction of a *space-feature memory field* to space has a peak at the currently attended location.

The activation in the *attention field* and the *feature memorization field*s are both input to three-dimensional *space-feature working memory fields*,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{sfm}}(x,y,f,t) &= -u^{\mathrm{F}}_{\mathrm{sfm}}(x,y,f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{sfm}} * \sigma(u^{\mathrm{F}}_{\mathrm{sfm}})](x,y,f,t) \\ &\quad +[w_{\mathrm{sfm,atn}} * \sigma(u_{\mathrm{atn}})](x,y,t) \\ &\quad +[w_{\mathrm{sfm,fme}} * \sigma(u^{\mathrm{F}}_{\mathrm{fme}})](f,t) \\ &\quad -[w^{\mathrm{F}}_{\mathrm{sfm,cin}} * \sigma(u^{\mathrm{F}}_{\mathrm{cin}})](x,y,t) \\ &\quad +c_{\mathrm{sfm,mef}} u^{\mathrm{F}}_{\mathrm{mef}}(t),\end{aligned} \tag{4.13}$$

whose interaction kernels are set up for self-sustained peaks. The space-feature working memory for a feature F is only active if its corresponding memorize feature task node $u^{\mathrm{F}}_{\mathrm{mef}}$ is active (for more details, see Section 4.1.7). The projections of the expanded lower-dimensional inputs from the attention field and feature memorization fields intersect at the spatial position and feature estimate of the currently inspected location, thus creating a working memory of the link between space and feature.

A conditional inhibition field,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{cin}}(x,y,t) &= -u^{\mathrm{F}}_{\mathrm{cin}}(x,y,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{cin}} * \sigma(u^{\mathrm{F}}_{\mathrm{cin}})](x,y,t) \\ &\quad +[w_{\mathrm{cin,atn}} * \sigma(u_{\mathrm{atn}})](x,y,t) \\ &\quad +c^{\mathrm{F}}_{\mathrm{cin,nom}} \sigma(u^{\mathrm{F}}_{\mathrm{nom}}(t)),\end{aligned} \tag{4.14}$$

may delete working memory peaks at the currently inspected location if the *no match node* of the match detector $u^{\mathrm{F}}_{\mathrm{nom}}$ signals an outdated memory with respect to the current visual input.

The outputs of the *space-feature working memory fields* enter corresponding *space-feature query fields*,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{sfq}}(x,y,f,t) &= -u^{\mathrm{F}}_{\mathrm{sfq}}(x,y,f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{sfq}} * \sigma(u^{\mathrm{F}}_{\mathrm{sfq}})](x,y,f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{sfq,sfm}} * \sigma(u^{\mathrm{F}}_{\mathrm{sfm}})](x,y,f,t) \\ &\quad +[w_{\mathrm{sfm,atn}} * \sigma(u_{\mathrm{atn}})](x,y,t) \\ &\quad +[w_{\mathrm{sfm,cue}} * \sigma(u^{\mathrm{F}}_{\mathrm{cue}})](f,t)\end{aligned} \tag{4.15}$$

and are matched with the spatial input of the *attention field* or cues from their corresponding feature cue field. These query fields operate in a more selective regime, building up a peak if working memory contains an entry at the attended location. The output is reduced to a feature readout (summing up along both spatial dimensions) and represented in a *feature query field*,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{fqu}}(f,t) &= -u^{\mathrm{F}}_{\mathrm{fqu}}(f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{fqu}} * \sigma(u^{\mathrm{F}}_{\mathrm{fqu}})](f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{fqu,sfq}} * \iint \sigma(u^{\mathrm{F}}_{\mathrm{sfq}})dxdy](f,t) \\ &\quad +c^{\mathrm{F}}_{\mathrm{fqu,ies}} u^{\mathrm{es}}_{\mathrm{int}}(t),\end{aligned} \tag{4.16}$$

which is only active if the explore scene behavior is active, passing on the cued feature to its corresponding *feature expectation field*

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{exp}}(f,t) &= -u^{\mathrm{F}}_{\mathrm{exp}}(f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{exp}} * \sigma(u^{\mathrm{F}}_{\mathrm{exp}})](f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{exp,fqu}} * \sigma(u^{\mathrm{F}}_{\mathrm{fqu}})](f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{exp,cue}} * \sigma(u^{\mathrm{F}}_{\mathrm{cue}})](f,t),\end{aligned} \tag{4.17}$$

thus projecting an expectation of feature estimates into the match detector. The feature expectation may also be shaped by top-down feature cues $u^{\mathrm{F}}_{\mathrm{cue}}$ during visual search and query.

The *no match fields* receive inputs from their respective *feature extraction* and *feature expectation fields*, following the definition in Section 3.2.2,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{nom}}(f,t) &= -u^{\mathrm{F}}_{\mathrm{nom}}(f,t) + h \\ &\quad +[w^{\mathrm{F}}_{\mathrm{nom}} * \sigma(u^{\mathrm{F}}_{\mathrm{nom}})](f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{nom,fex}} * \sigma(u^{\mathrm{F}}_{\mathrm{fex}})](f,t), \\ &\quad +[w^{\mathrm{F}}_{\mathrm{nom,exp}} * \sigma(u^{\mathrm{F}}_{\mathrm{exp}})](f,t).\end{aligned} \tag{4.18}$$

I do not name these fields *match fields*, since supra-threshold peaks in these fields signal a mismatch between extracted and expected features values. Thus, *no match* is a more suitable name in this context.

4.1.6 Cues and Query

For each *space-feature query field*, a corresponding *feature cue field*,

$$\begin{aligned}\tau \dot{u}^{\mathrm{F}}_{\mathrm{cue}}(f,t) &= -u^{\mathrm{F}}_{\mathrm{cue}}(f,t) + h + s^{\mathrm{F}}_{\mathrm{cue}}(f,t) \\ &\quad +[w^{\mathrm{F}}_{\mathrm{cue}} * \sigma(u^{\mathrm{F}}_{\mathrm{cue}})](f,t)\end{aligned} \tag{4.19}$$

may contain peaks at feature estimates originating in other cognitive processes (i.e., language processing). Each *feature cue field* is connected to its corresponding *space-feature query field*, resulting in a peak if input from the working memory matches the current cue. In addition, each *feature cue field* projects to the *feature expectation field*, implementing an expectation for the feed-forward feature extraction through inhibitory coupling to the *no match field* (see Section 3.2.2 and Equation 4.18). If the feature is also involved in the saliency extraction pathway (e.g., color), the *feature cue field* projects its activation to the corresponding *early space-feature field* (as it is the case for color).

Both the top-down retuning of saliency and the emergence of peaks in the *space-feature query fields* influence the *attention field* in its selection decisions. The top-down retuning uses a graded effect, as cued feature values lead to more localized activation at matching sites. The influence from the *space-feature query fields* passes through a *memory bias field*

$$\begin{aligned} \tau \dot{u}_{\text{meb}}(x,y,t) &= -u_{\text{meb}}(x,y,t) + h \\ &\quad + [w_{\text{meb}} * \sigma(u_{\text{cin}})](x,y,t) \\ &\quad + \sum_{\text{F}} [w^{\text{F}}_{\text{meb,sfq}} * \int \sigma(u^{\text{F}}_{\text{sfq}}) df](x,y,t) \\ &\quad - \sum_{\text{F}} c^{\text{F}}_{\text{meb,fqu}} \sigma(u^{\text{pd,F}}_{\text{cue}}(t)) + c^{\text{qs}}_{\text{meb,int}} \sigma(u^{\text{qs}}_{\text{int}}(t)), \end{aligned} \tag{4.20}$$

whose resting level depends on the amount of active feature cue field peak detectors $u^{\text{pd,F}}_{\text{cue}}$ and the state of the query intention node $u^{\text{qs}}_{\text{int}}$. The excitatory dependence on the intention node state assures that the *space-feature query fields* do not have an influence on attention during exploration. The inhibitory dependence on feature cue field peak detectors lowers the resting level for each expected contribution of any *space-feature query field*. In this way, u_{meb} only undergoes a detection decision at locations at which there is enough evidence for a match along all queried feature channels. In conjunction searches, this highlights fewer candidates in contrast to the retuned bottom-up saliency pathway, as detection decisions in u_{meb} only happen for candidates that match all feature channels, while bottom-up retuning increases the saliency of partially matching candidates as well.

4.1.7 Behavioral Organization

I now take a closer look at the behavioral organization of *exploration*, *maintenance*, and *query*. For this, I introduce a number of intention and CoS nodes, following the principles introduced in Section 3.2.1. The behaviors may affect

an arbitrary amount of feature channels (e.g., memorizing sizes or querying a specific color). To cover this, each behavior's condition of satisfaction and dissatisfaction nodes are parameterizable to listen to all currently active feature channels. Each channel is connected to these nodes in the same way, using a match detector as driving force of the behaviors.

Exploration

The *exploration* behavior is characterized by a hierarchical structure (see Figure 4.4). On the scene level, an *explore scene* intention node

$$\begin{aligned}\tau \dot{u}^{\text{es}}_{\text{int}}(t) &= -u^{\text{es}}_{\text{int}}(t) + h + s^{\text{es}}_{\text{tsk}}(t) \\ &\quad + c_{\text{es}}\sigma(u^{\text{es}}_{\text{int}}(t)) - c_{\text{es,qs}}\sigma(u^{\text{qs}}_{\text{int}}(t))\end{aligned} \tag{4.21}$$

expresses the intention to continuously explore the current scene and create working memory representations of inspected objects. This node is activated once it receives task input $s^{\text{es}}_{\text{tsk}}$. The *explore scene* node can be suppressed by other competing behaviors (e.g., to fixate the current state of working memory or to have exclusive access to shared resources such as the attention field). In Equation 4.21, the query behavior can potentially suppress the exploration once its intention node $u^{\text{qs}}_{\text{int}}$ on the scene level becomes active. In addition, the exploration intention activates the object level by projecting its supra-threshold output to an *inspect object* intention node

$$\begin{aligned}\tau \dot{u}^{\text{io}}_{\text{int}}(t) &= -u^{\text{io}}_{\text{int}}(t) + h + c_{\text{io}}\sigma(u^{\text{io}}_{\text{int}}(t)) \\ &\quad + c_{\text{io,es}}\sigma(u^{\text{es}}_{\text{int}}(t)) - c_{\text{io,io}}\sigma(u^{\text{io}}_{\text{cos}}(t))\end{aligned} \tag{4.22}$$

representing the inspection of a single object. This node in turn boosts the *attention field*, which picks the currently most salient location in the dorsal stream. The *inspect object* node also boosts its CoS node

$$\begin{aligned}\tau \dot{u}^{\text{io}}_{\text{cos}}(t) &= -u^{\text{io}}_{\text{cos}}(t) + h + c_{\text{io}}\sigma(u^{\text{io}}_{\text{cos}}(t)) \\ &\quad + c_{\text{io,io}}\sigma(u^{\text{io}}_{\text{int}}(t)) - c_{\text{io,mef}}\sum_{\text{F}}\sigma(u^{\text{F}}_{\text{mef}}(t)) \\ &\quad + c_{\text{io,mat}}\sum_{\text{F}}\sigma(u^{\text{F}}_{\text{mat}}(t))\end{aligned} \tag{4.23}$$

whose resting level is modified by the amount of active *memorize feature* nodes

$$\begin{aligned}\tau \dot{u}^{\text{F}}_{\text{mef}}(t) &= -u^{\text{F}}_{\text{mef}}(t) + h + s^{\text{F}}_{\text{tsk}}(t) \\ &\quad + c^{\text{F}}_{\text{mef}}\sigma(u^{\text{F}}_{\text{mef}}(t)).\end{aligned} \tag{4.24}$$

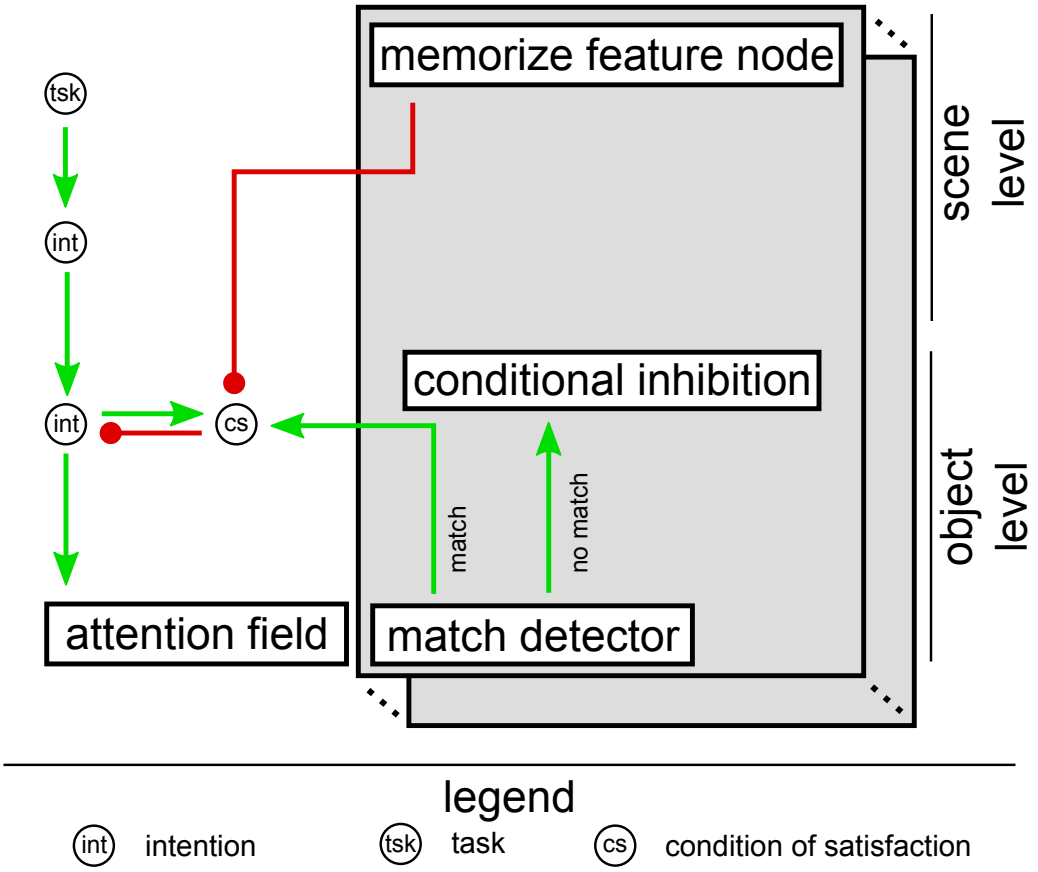

Figure 4.4: Behavioral organization for exploration and maintenance. The gray part exists for each feature channel.

They define the set of features to be memorized in working memory and also preactivate the *space-feature working memory fields* to be able to store space-feature links of all specified feature dimensions. The inhibitory influence of the *memorize feature* nodes is canceled out by excitatory input of the match nodes $u^{\mathrm{F}}_{\mathrm{mat}}$ along the currently active feature channels. $u^{\mathrm{io}}_{\mathrm{cos}}(t)$ thus becomes active once all match nodes signal a successful creation of working memory space-feature links. Memorize feature nodes are activated through task input $s^{\mathrm{F}}_{\mathrm{tsk}}$.

The activation of the CoS node is the consequence of a chain of instabilities. The attentional focus onto a single salient location establishes inputs for the match detector of each feature channel. If there is no current working memory representation to compare with the feed-forward feature extraction, the *feature memorization field* is not inhibited by the *has memory node* and forwards the feature extraction into the *space-feature working memory field*, creating a space-feature link as described in Section 4.1.5. This provides an expectation in the match detector. As soon as both feed-forward and memory inputs are present in the match detector, the *match node* becomes active, signaling the successful creation of the link. The *match node* is connected to the CoS node of the *inspect object* behavior. The resting level of the CoS node is lowered according to the amount of active *memorize feature nodes*. This ensures that the *match node* of every active feature has to be active to push the CoS node over its detection threshold. If this condition is met, the intention of the *inspect object* behavior is inhibited, effectively releasing the current object from fixation and turning off the CoS node. The reverse detection in the *attention field* also removes the inputs to the match detector of each feature channel. The behavior is now ready to be activated again, as long as the task input is still active and no competing behavior suppresses the *explore* intention on the scene scope. Through the *looking working memory*, the recently visited and memorized items are less likely to be picked again by the re-activation of the attention field. Note that there is no CoS node on the scene scope, as exploration is a continuous behavior. The continuity of object inspections is also a prerequisite for maintenance operations for existing space-feature links.

Maintenance

Maintenance comprises mechanisms of updating the working memory according to changes in the scene. Tracking of changes in position and deletion of removed items were previously demonstrated and discussed by Zibner and colleagues [196, 197]. These mechanisms operate in parallel on the whole input area and require no attentional selection. Thus, they require no explicit

behavioral organization. In the present extension, I add the autonomous updating of changes in the feature description of an item. These changes may be related to the pose, for example, changing the orientation of an object and exposing a previously non-accessible view of the item with changes in feature values (e.g., flipping a multi-colored item from a 'blue' side to a 'red' side). Changes may also be induced by replacing an unattended item by another item at the same spatial position, resulting in change blindness.

I focus on the behavioral processes described in Section 4.1.7 for an inspection of a previously visited item whose feature description has changed since the last visit. The decay of attention's memory trace affects the *looking working memory*, whose diminishing inhibition allows to put the previously visited item back into the foreground of the inspection behavior after several fixations. Since there already is a working memory representation of this item, the *space-feature query fields* immediately deliver input to the *feature expectation fields* through selections in *space-feature query fields* and *feature query fields*, creating an inhibitory expectation input in the *no match fields*. For an unchanged item, the CoS nodes of the inspection behavior turn on, since no peak is created in any of the *no match fields*, resulting in active *match nodes*. For a changed item, this is no longer true, as the expectation does not match the bottom-up extraction of the current item. One or more active *no match nodes* prevent the inspection behavior's completion.

The activation of a *no match node* in any feature channel triggers a conditional inhibition in the *space-feature working memory* associated with this feature channel. The projection from any conditional inhibition field $u^{\mathrm{F}}_{\mathrm{cin}}$ to its associated space-feature working memory is strong enough to delete the working memory link at the currently inspected position defined by the activation peak in the attention field. As a consequence, the space-feature query field loses its peak as well, which removes the input into the match detector and subsequently deboosts the *no match node* and the conditional inhibition. At the same time, the missing working memory representation of the foreground item is detected by the *has memory node* $u^{\mathrm{F}}_{\mathrm{hme}}$, which no longer inhibits the *memorize feature field*, which forwards the current feature value to working memory. A new link is created, which again is used as estimation input into the match detector. The *match node* of this feature channel now turns on, signaling the partial fulfillment of the CoS of object inspection. If this maintenance operation is finished in all mismatching feature channels, the CoS node of the inspection behavior turns on and fixation is released. As a consequence of the maintenance, the inspection behavior dwells longer on the location of changed items, but correctly updates the internal representation of the inspected item.

Query and Visual Search

The goal of a query is to establish a selection decision in the *attention field* picking the item that matches the given cues. Visual search uses the same behavioral organization, but assumes that the search operates on a completely unrepresented scene. The query behavior is a reactive behavior that becomes active once cue input arrives at the scene representation architecture. Figure 4.5 shows the nodes and connectivity involved in this behavior. Peaks in *feature cue fields* are detected by associated peak detectors, which activate a *query intention* node

$$\begin{aligned}\tau \dot{u}^{\mathrm{qs}}_{\mathrm{int}}(t) &= -u^{\mathrm{qs}}_{\mathrm{int}}(t) + h + c_{\mathrm{qs}}\sigma(u^{\mathrm{qs}}_{\mathrm{int}}(t)) \\ &\quad + c_{\mathrm{qs,cue}}\sigma\left(\sum_{\mathrm{F}} \sigma(u^{\mathrm{pd,F}}_{\mathrm{cue}}(t))\right)\end{aligned} \tag{4.25}$$

on the scene scope, which inhibits any other ongoing behaviors (such as the exploration behavior) to gain exclusive access to the *attention field*. The query intention projects onto an intention node on the object scope

$$\begin{aligned}\tau \dot{u}^{\mathrm{qo}}_{\mathrm{int}}(t) &= -u^{\mathrm{qo}}_{\mathrm{int}}(t) + h + c_{\mathrm{qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{int}}(t)) \\ &\quad + c_{\mathrm{qo,qs}}\sigma(u^{\mathrm{qs}}_{\mathrm{int}}(t)) - c_{\mathrm{qo,qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{cod}}(t)),\end{aligned} \tag{4.26}$$

which represents the intention to sample a single item from the current saliency configuration. This connectivity resembles the hierarchy of intentionality already used in the *exploration* behavior. This *query object* node boosts the *attention field* once it becomes active. The *no match* nodes of all cued features couple back into a CoD node

$$\begin{aligned}\tau \dot{u}^{\mathrm{qo}}_{\mathrm{cod}}(t) &= -u^{\mathrm{qo}}_{\mathrm{cod}}(t) + h + c_{\mathrm{qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{cod}}(t)) \\ &\quad + c_{\mathrm{qo,qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{int}}(t)) + c_{\mathrm{qo,nom}}\sum_{\mathrm{F}} \sigma(u^{\mathrm{F}}_{\mathrm{nom}}(t)),\end{aligned} \tag{4.27}$$

which inhibits the intention node if any of the no match nodes in the active feature channels detects a mismatch between cue and currently selected candidate. The *match* nodes excite a CoS node of the behavior

$$\begin{aligned}\tau \dot{u}^{\mathrm{qo}}_{\mathrm{cos}}(t) &= -u^{\mathrm{qo}}_{\mathrm{cos}}(t) + h + c_{\mathrm{qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{cos}}(t)) + c_{\mathrm{qo,qo}}\sigma(u^{\mathrm{qo}}_{\mathrm{int}}(t)) \\ &\quad + c_{\mathrm{qo,mat}}\sum_{\mathrm{F}} \sigma(u^{\mathrm{F}}_{\mathrm{mat}}(t)) - c_{\mathrm{qo,cue}}\sum_{\mathrm{F}} \sigma(u^{\mathrm{pd,F}}_{\mathrm{cue}}(t)),\end{aligned} \tag{4.28}$$

which requires all *match nodes* of queried features to be active in order to pierce the detection threshold. If any of the given cues do not match the feedforward extraction of features for the currently selected item, the CoD node

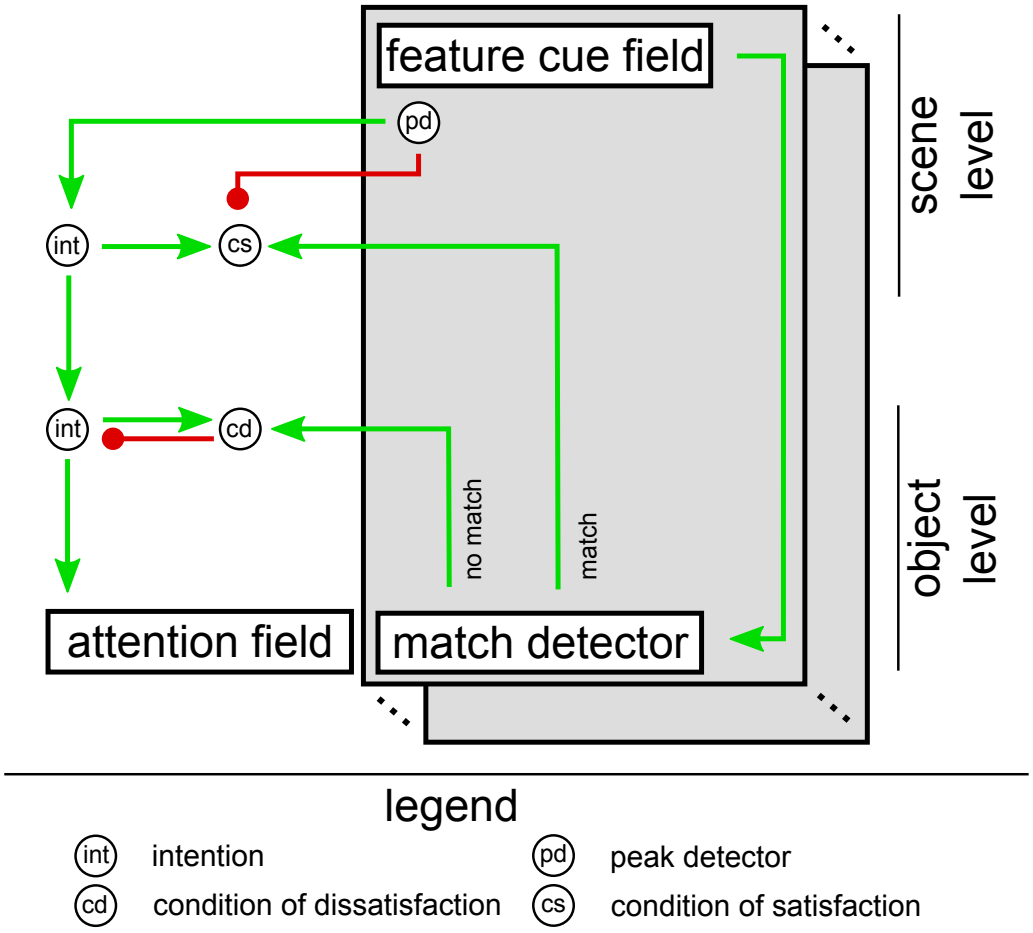

Figure 4.5: Behavioral organization for query. Gray part exists for each feature channel.

is turned on and inhibits the intention node for single item query. Through the detection of dissatisfaction, the *attention field* is released and ready for switching to another item once the single item intention node reactivates. Once the selected item fits all given cues (i.e., all *match nodes* of cued features are active), the CoS node activates and stops the loop. The query behavior does not automatically turn itself off when the CoS is reached. It assures that the attentionally selected item fits the query cues and can be used in other parts of cognitive processing. Taking away the cue inputs automatically turns off the query behavior, thus re-enabling other competing behaviors such as the exploration. Note that the query behavior might fail if there is no item in the current scene that fits all given cues. In this case, intrinsic motivational mechanisms (such as frustration) or other contingency behaviors (such as checking back with the person who initiated the query if all given cues are indeed correct or if the item is maybe missing from the scene) may take over. These are not modeled in the present work.

4.1.8 Exemplary Instabilities

After describing the behavioral organization of the three behaviors in the previous section, I give an exemplary chain of instabilities occurring during these behaviors. As an example, I assume a sequence of two fixations, the first to a novel object of blue color and small size, the second to a changed object—also blue, but big—with a mismatching representation in working memory (e.g., green and small), followed by an interrupting query with a combination of two cues, which reinstantiates the queried object with the second fixation. Figure 4.6 contains a sketch of the initial scene and the content of the internal representation in the top row.

The following instabilities occur during the exploration of the first, novel object. First, the inspect node gets activated, which in turn results in an attentional selection of a single, most salient location. Since the big, blue object is already represented in working memory and was likely inspected recently, the novel object is more salient. Through feature extraction, a complete description of the inspected location is represented as peaks in the *feature fields*, which then passes through the *feature memorization fields* and is linked to the spatial position in working memory. The newly created working memory peaks induce peaks in the *space-feature query fields* and consequently in the *feature expectation fields* as well. Now, the expected feature values match the feed-forward representation in the *feature extraction fields*, leading to the dissolution of peaks in the *no match fields*. The match node becomes active, therefore reaching the CoS of the inspection behavior. The intention node is pulled below threshold by the CoS node, which in turn

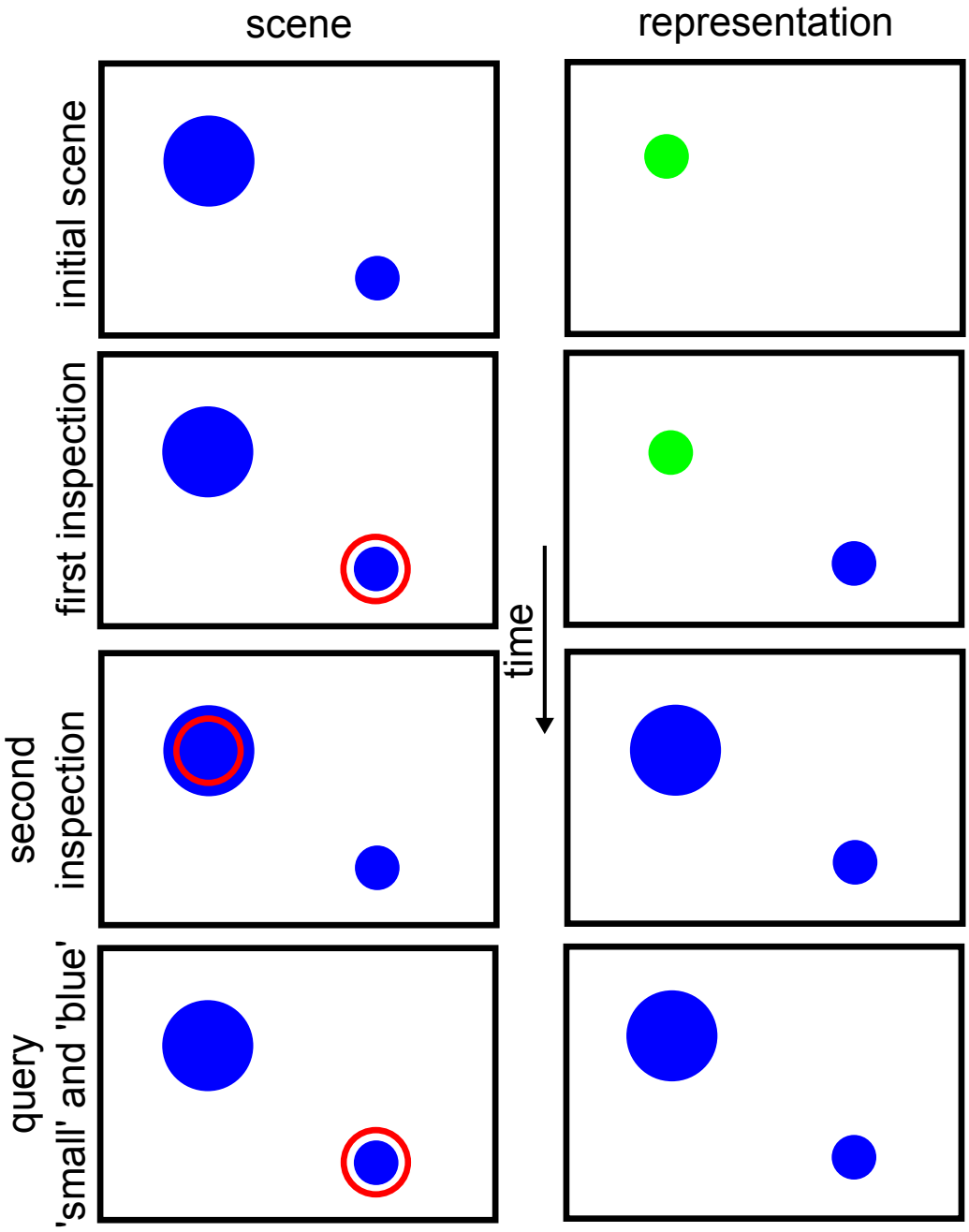

Figure 4.6: This figure shows an exemplary sequence of fixations in a visual scene in the left column and the corresponding content of the internal representation in the right column. Rows are snapshots of time-continuous behavior. The red circle marks the focus of attention. See text for details.

releases the *attention field* from its boosted state. The reverse detection in the *attention field* turns off the match node, as there is no current feed-forward feature input into the match detector. See Figure 4.6, second row from top, for a sketch of the internal representation after first inspection.

Parallel to this chain of instabilities, the memory trace of the *attention field* and the newly created space-feature links of the working memory overlap in the *looking working memory field*, whose inhibitory coupling to the *attention field* decreases the saliency of the recently inspected location. This has an effect on the re-activating inspection behavior: due to the lowered saliency of already inspected locations, the inspection behavior now picks a new location in the input stream, which is either not represented in working memory or was not visited for a certain amount of time, determined by the decay rate of the memory trace of the *attention field*.

The second fixation brings an already represented object into the foreground. The attentional focus reinstantiates the feature description of the object from working memory, which is compared to the currently extracted feed-forward feature description using the match detector. Since this object changed in color and size after the last fixation onto it, the *no match nodes* of the match detectors for color and size are activated and inhibit the associated *match nodes*. In turn, conditional inhibition triggered by the *no match nodes* deletes the outdated working memory peak in the *space-color* and *space-size working memory fields*. As a consequence, the *has memory nodes* turn off, activating the *memorize feature* fields. A new peak is created in each *space-feature working memory field*, which in turn produces an updated input into the *feature expectation* fields and the match detectors. The *no match nodes* no longer receive input from the *no match fields* and do not inhibit the *match nodes*, which consequently turn on and activate the CoS node of the inspection behavior. At this point, the internal representation has adapted to the change in the scene. See Figure 4.6, third row from top, for a sketch of the internal representation after second inspection.

A combined cue of the color 'blue' and the size 'small' enter the architecture through the respective *feature cue* fields. Their peak detectors turn on and activate the query behavior, which inhibits the exploration behavior through its suppression node. The query intention activates the intention node on the object scope, which boosts the *attention* field. The saliency input influencing the *attention* field's selection decision is retuned given the current cues and the content of working memory. The attention field may pick the blue big object first, since object size is an intrinsic influence on saliency. The features color and size are compared in the respective match detectors. For color, the *match* node is activated, signaling the satisfaction of the cue constraint. For size, a mismatch is detected in the *no match field*,

activating the *no match* node and subsequently the CoD node of the object query behavior, releasing the fixation on the mismatching object. With the reverse detection of the *attention* field, all match detector nodes turn off. The looking memory reduces the saliency of this object, leaving other candidates more salient for the next re-activation of the intention node.

With the second selection decision of the attention field, triggered by the re-activating intention node, the small blue object is picked. Both color and size are compared in the match detectors and the *match* nodes turn on. The CoS node of the query turns on since the summed activation of all *match* nodes is strong enough to push it over the detection threshold. The queried object is kept in the attentional focus of the architecture as long as the cues are not deactivated. Subsequent behaviors connected to the CoS node of the query behavior can now apply operations to this object. See Figure 4.6, bottom row, for a sketch of the internal representation after successful query.

4.2 Experiments

The following sections present experiments that take a closer look at the dynamics of the behaviors *exploration*, *maintenance*, and *query*. I use the to evaluate the architecture's capabilities to produce the behavior described in the example given in Section 4.1.8.

4.2.1 Exploration

In this experiment, the exploration behavior is evaluated on scenes containing a varying amount of objects from a pool of household objects (dishes, toys, tools, food packages, hygiene products, office supplies; see Figure 4.7 for an overview of used objects and Figure 4.8 for example scenes). Some scenes are made visually challenging by adding background clutter such as smaller objects or textured paper. Each scene is recorded as a video with a length of around 1:30 min, which is input to the scene representation (see the data set in Appendix A). My architecture explores each scene for 30 seconds and builds up internal representations, which are continuously recorded. For a qualitative evaluation, the placement of attention and the activation pattern of the behavioral organization are also recorded. To quantitatively evaluate the internal representation, I define two measures: *precision* and *coverage*.

Precision measures how closely the internal representation reflects the sensory input. Sources of error include drifts of working memory peaks over time and discretization errors due to limited sampling of the field dimensions. I use two variants – *continuous* and *discrete* – for measuring precision to

Figure 4.7: This figure shows the object set used in the experiments. Note that I use the wooden blocks on the left as clutter, as they are relatively small in comparison to the other objects.

evaluate the suitability of each measure.

Continuous precision For every sample point of field activation in space-feature working memory, the represented feature value $v_{\text{rfv}}(x, y)$ is compared to the mean feature value $v_{\text{mfv}}(x, y)$ at the same position in the feed-forward input. The mean feature value is generated by temporal averaging. The mean error

$$f_{\text{pre}}(t) = \frac{\iint v_{\text{err}}(x, y, t)\sigma(\int \sigma(u_{\text{sfm}}(x, y, f, t))df)dxdy}{\iint \sigma(\int \sigma(u_{\text{sfm}}(x, y, f, t))df)dxdy} \tag{4.29}$$

is calculated by integrating the error between represented and mean feature values, $v_{\text{err}}(x, y, t) = |v_{\text{rfv}}(x, y, t) - v_{\text{mfv}}(x, y, t)|$, of all regions with supra-threshold activation (integrated along f and thresholded again) and normalizing with the integral of these regions. $f_{\text{pre}}(t) \approx 0$ corresponds to a high precision, whereas larger values indicate a loss in precision. For cyclic metrics such as color hue, mean and error calculations have to be replaced by variants suitable to capture the properties of circularity. See Appendix C for details.

Discrete precision For every peak in space-feature working memory, a peak position is determined. Represented feature values are compared to

Figure 4.8: This figure shows example scenes assembled from the object set shown in Figure 4.7. Different variations of scene complexity are shown. The top-left example shows a scene with uniform background. The top-right example introduces a local patterned background. The bottom-left example shows a scene cluttered with small objects. The bottom-right example combines patterned background and object clutter.

mean feature values only at peak positions. The resulting error is normalized by the amount of peaks. See Algorithm 1 for pseudocode. This measure can be applied to a search window around the peak positions instead to increase robustness against outliers in the mean feature values by only considering the minimal error of the whole search window.

The continuous precision measure does not take into account that each region may contain multiple distinct feature values, while the internal representation only stores a single value. Objects with complex shape and color scheme thus lead to a decrease in precision. The discrete precision measure only takes a single mean feature value into account. This method treats the single-peak nature of the internal representation more fairly, but is prone to picking outliers from the mean feature values. Using a search window reduces

Algorithm 1 Mean feature error for discrete precision.

$p \leftarrow 0$
$i \leftarrow$ number of peaks in memory
for all peaks in memory **do**
  $p_{\text{wm}} \leftarrow$ center of peak
  $v_{\text{mfv}} \leftarrow$ memory feature value at p_{wm}
  $v_{\text{rfv}} \leftarrow$ mean feature value at p_{wm}
  $p \leftarrow p + |v_{\text{rfv}} - v_{\text{mfv}}|$
end for
return $\frac{p}{i}$

this influence, but distorts the error in other ways.

Coverage is a measure of how much information about a scene is available in the internal representation. Similar to precision, *continuous* and *discrete* coverage measures can be defined.

Continuous coverage A *continuous* measure of coverage can be derived from comparing the amount of supra-threshold activation in space-feature working memory to the amount of activation in the saliency field u_{sal},

$$f_{\text{cov}}(t) = \frac{\iiint \sigma(u_{\text{sfm}}(x, y, f, t))dxdydf}{\iint \sigma(u_{\text{sal}}(x, y, t))dxdy}. \tag{4.30}$$

The resulting value $f_{\text{cov}}(t)$ is normalized with the activation level of the input stream.

Discrete coverage Instead of comparing activation levels directly, one can determine the amount of peaks in memory and compare this number with the number of peaks in the saliency pathway. An additional indicator of coverage is the mean distance between peak positions in memory and saliency pathway. For each memory peak, the closest saliency peak is found, with the distance between them serving as contribution to this indicator. See Algorithm 2 for pseudocode computing discrete coverage and mean peak distance.

The continuous coverage measure does not take into account that the activation pattern in the saliency pathway and working memory differ in shape. Saliency peaks adhere to the object shapes, whereas the shape of working memory peaks is strongly influenced by the Gaussian interaction kernel. Elongated objects thus cover different areas in the saliency map and working memory, which in turn leads to errors in the continuous coverage measure. The discrete coverage measure exhibits similar problems with elongated objects, as they tend to induce more than one saliency peak per object.

Algorithm 2 Discrete coverage and mean peak distance.

$d \leftarrow 0$
$i \leftarrow$ number of peaks in memory
$j \leftarrow$ number of peaks in saliency map
for all peaks in memory **do**
  $p_{\text{wm}} \leftarrow$ center of peak
  $p_{\text{sal}} \leftarrow$ center of closest saliency peak
  $d \leftarrow d + |p_{\text{wm}} - p_{\text{sal}}|$
end for
return $\frac{i}{j}$, $\frac{d}{i}$

Results

The recorded placement of attention over the 30 second exploration interval gives a qualitative insight into the exploration behavior. Attention lands mostly on the objects placed in the scene (see examples in Figure 4.9). In some cases, parts of the background (Figure 4.9, top row) and different parts of an object (Figure 4.9, middle row) are attended. Since attention operates on a proto-object level, this is likely to happen.

An exemplary timeline of node activations of the behavioral organization is shown in Figure 4.10. An active *inspect object* intention node boosts attention, which in the end produces the inputs to each match detector. Only after every match node is active, the *inspect object* CoS node may become active as well. The intention node is subsequently inhibited, which not only deactivates the CoS node, but also removes attention and thus the inputs to the match detectors. Due to the length of the chain of fields and nodes this event has to traverse, the match detectors stay on longer than the CoS.

The advantages and disadvantages of continuous and discrete evaluation measures are examined by a quantitative analysis of the internal representation of 15 scenes. Figure 4.11 shows the resulting precision measurements for the color channel over time. The continuous precision measure exhibits the highest mean feature error. The mean error decreases for discrete precision measures with single values and search window. For all three precision measures I observe that the mean error remains quasi-constant, which excludes memory drift along the feature dimension as a source of error.

The coverage measures both show an increase of coverage over time (see Figure 4.12, top and middle). The mean peak distance remains quasi-constant at a subpixel value, which translates to a mean displacement of approximately 7.2 mm on the table plane. This suggests that there is no significant drift along the spatial dimensions over the recorded time period.

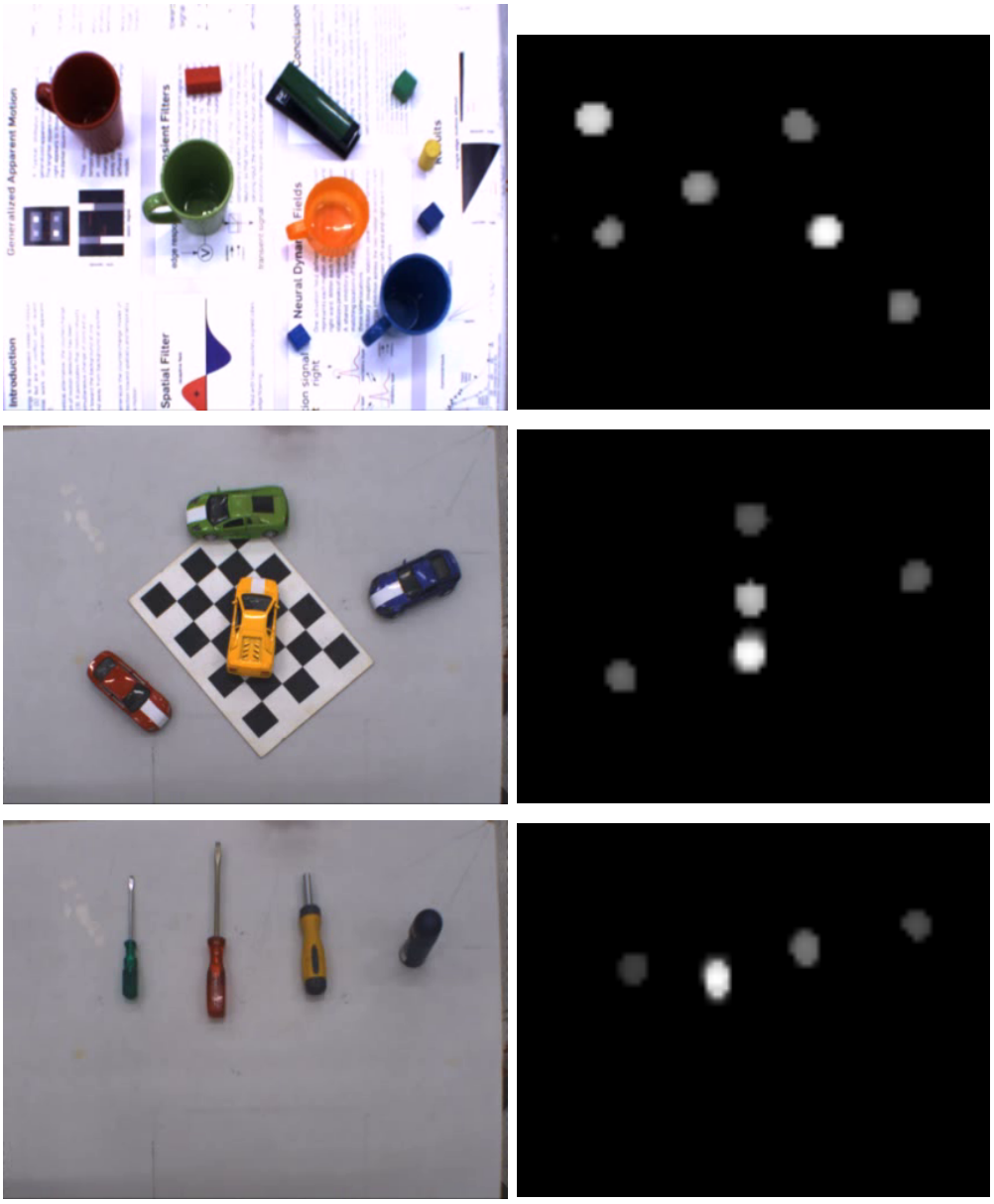

Figure 4.9: This figure shows the input images in the left column and accumulated fixations after 30 seconds of free viewing in the right column for three example images.

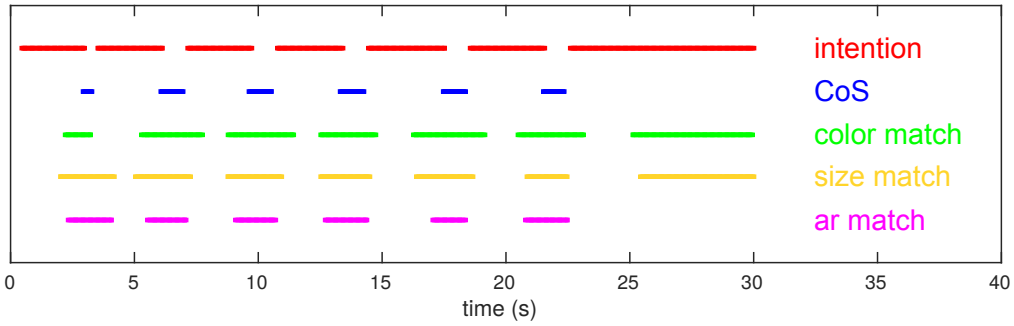

Figure 4.10: The activation of intention node, condition of satisfaction node and the three match detector nodes of the *inspect object* behavior is plotted here over time. Bars denote activation above 0.8.

Discussion

The exploration experiment shows how the comparison of memorized feature values and values extracted from the input stream at the current attentional focus are the drive for autonomous exploration of a scene. The repeated release from fixation signaled by the CoS node of the inspect behavior and the accumulating looking working memory generate sequences of fixations covering multiple objects in the scene. Objects are inspected in order of their saliency, with small objects rarely entering working memory. The proto-object nature of representation leads to multiple space-feature links for single large objects and space-feature links appearing for regions considered as background clutter. A more sophisticated saliency computation considering depth and shape cues may improve the consistency between the proto-object and object level.

4.2.2 Maintenance of Features

In the previous experiment, scenes remained static during the exploration sequence. The internal representation may become invalid if feature values change either through interactions with the scene (e.g., turning an object over to reveal a previously unseen appearance) or drift and dissolution of working memory. On attentional re-entry, these changes lead to a mismatch and subsequent maintenance of working memory. In this experiment, the architecture is exposed to videos of scenes with objects of different color. After a time interval that allows initial exploration of the scene, objects in the scene are replaced with other objects (see Figure 4.13). The internal representation of replaced objects is now outdated. The duration of fixations and the activation pattern of the behavioral organization are recorded for

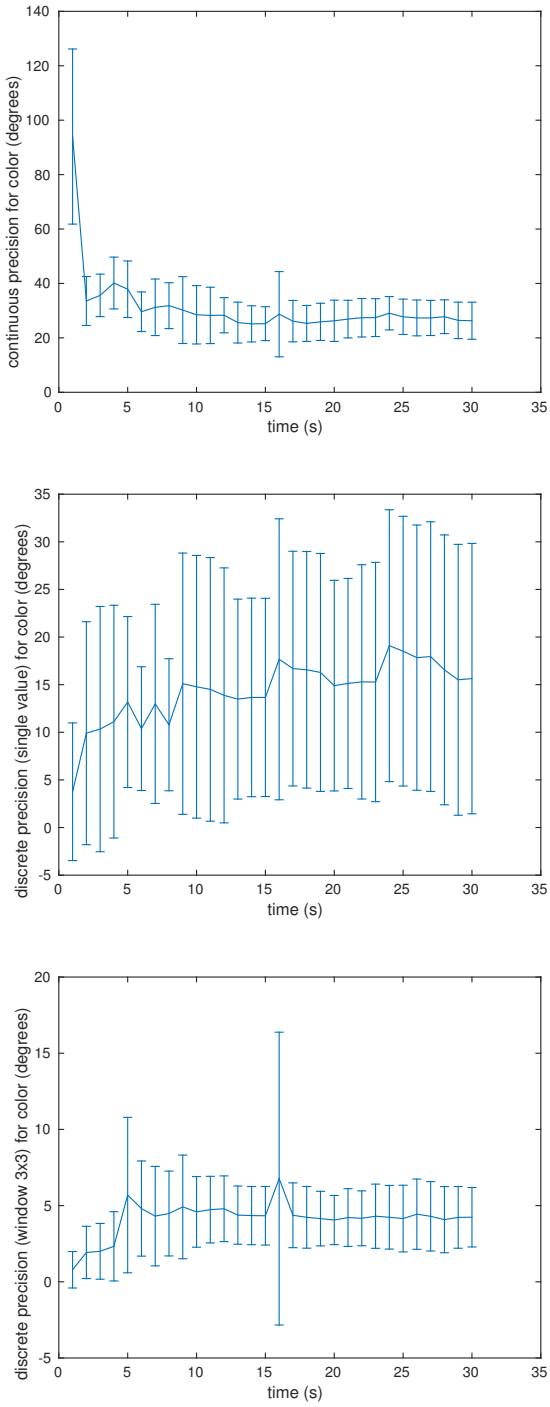

Figure 4.11: Continuous and discrete color hue precision over time, starting after the first item is stored in memory. Different measures result in different mean feature errors. See text for details.

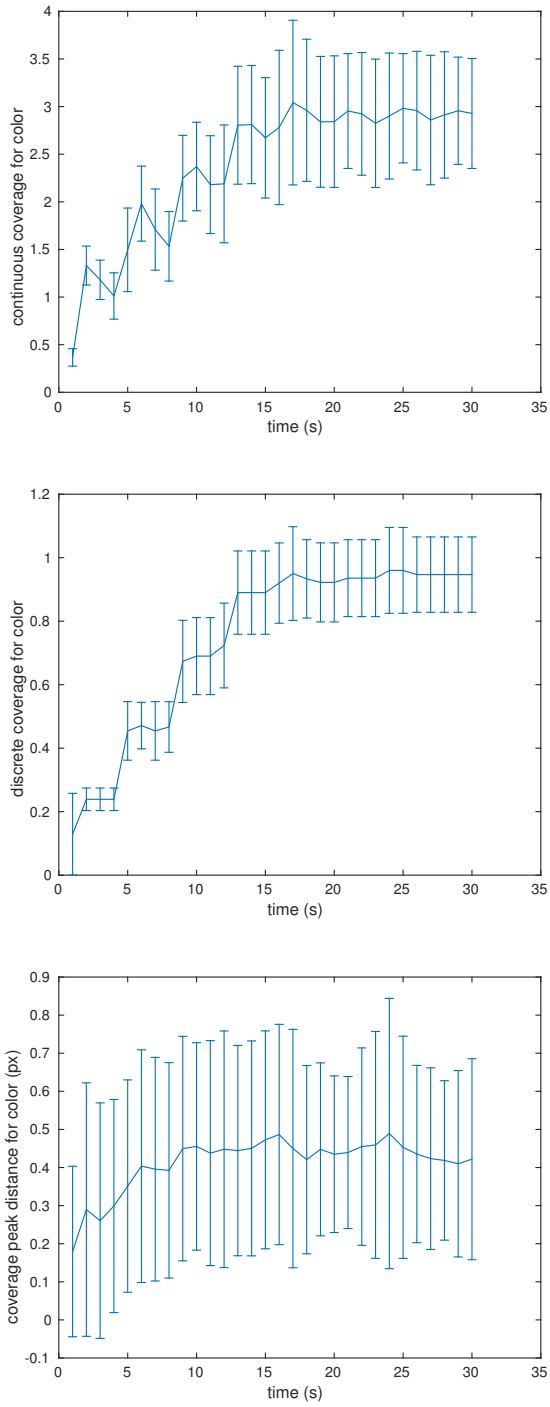

Figure 4.12: Mean continuous and discrete coverage are plotted on the top and in the middle, respectively. The bottom plot shows mean peak distance over time.

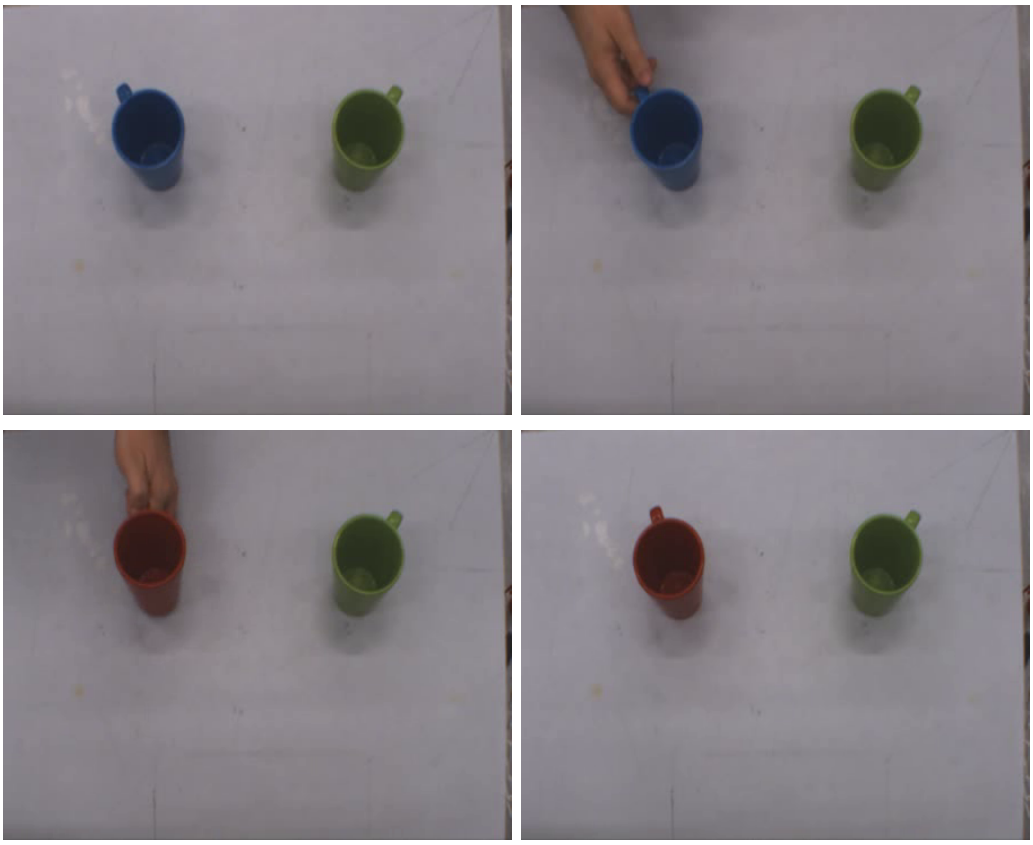

Figure 4.13: These images are snapshots from the maintenance of features experiment. Objects in the scene are manually replaced by other objects with different feature values.

evaluation.

Results

Once fixation returns to a replaced object, a mismatch between memory and current input is detected by the no-match detector (see Figure 4.14). The memory is deleted at this location. If no memory exists, a new peak is created using the current feature input. Now the match node ends the inspection behavior and fixation is released. This process takes considerably more time than the no-change case, which is reflected in the fixation duration (see Figure 4.15).

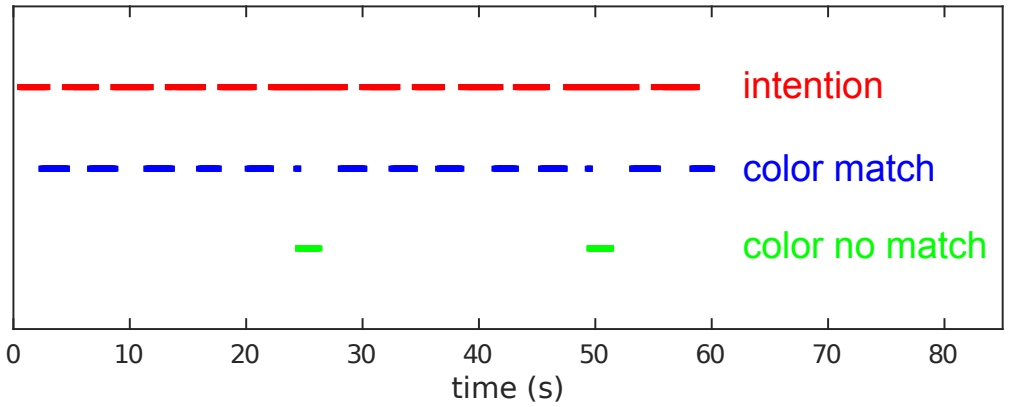

Figure 4.14: Activation of intention (red), match (blue) and no-match (green) nodes for the *inspect object* behavior. The no-match node is activated twice, once for each changed item.

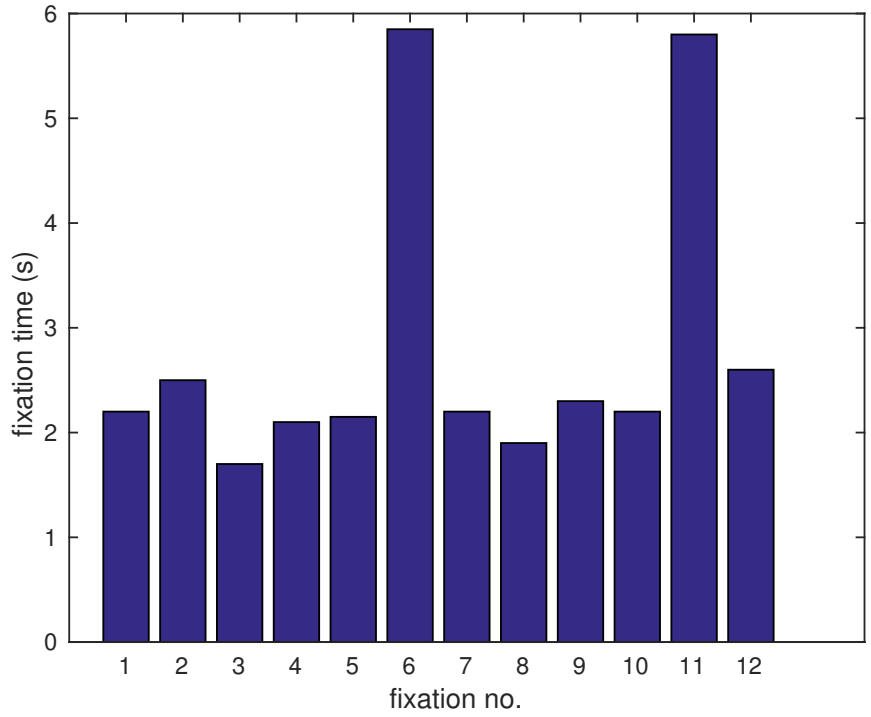

Figure 4.15: Duration of fixations for the example input of Figure 4.13. The first two fixations store the scene in working memory. All subsequent fixations re-evaluate the working memory given the current input. On the first fixation of each changed item, the fixation duration is significantly larger.

Discussion

The increased fixation duration during change detection is in alignment with psychophysical evidence by Hollingworth and colleagues [78] and a review by Rensink [136] who differentiates change detection in implicit (i.e., no conscious awareness of change despite behavioral signatures of its detection) and explicit perception. In the present architecture, change detection is a transient signal represented by the supra-threshold activation of the no-match node of any feature match detector. Since the activation of any no-match node automatically triggers a feature maintenance operation, the no-match node is deactivated after conditional inhibition deletes the working memory entry. If the nervous system is aware of the transient activation of the no-match node, a subsequent change response may happen at chance, as the working memory representation is meanwhile deleted or may default to a 'no change' response, as a matching representation emerges during the maintenance operation.

4.2.3 Maintenance of Positions

The spatial position of objects is the binding dimension of this scene representation architecture. Changing an object's position effectively prevents straight-forward access to the memorized features of said object, as there is no memory at the new position of the object and the history of object movements is not memorized. If an object is in the attentional focus, new working memory is created during object movement as attention leaves the region of existing working memory. If, however, attention is currently not focused on the moving object, working memory is lost. To mend this, working memory can be linked to current saliency input, which is expanded to the three-dimensional space-feature fields. The resulting tubes of activation are an active contribution to the self-sustained regime of the working memory peaks. If objects move, the tube input adapts to the positional changes. Existing working memory peaks track this change, with feature values being moved around according to the object movements. No explicit attention is necessary for this maintenance operation, as saliency is available without attentional focus. The left column of Figure 4.16 shows a time series of a dynamic scene, in which three colored robots move around.

The introduced dependence on saliency input has a direct consequence for the maintenance of working memory links that are not supported by saliency input. Input may be missing due to occlusion or objects getting out of the view of a moving camera head. To keep objects in working memory without support by saliency, localized resting boosts may replace this excitatory

input. For example, having an increased resting level for all regions outside of the current camera view on the observed scene keeps space-feature links alive if objects move out of view. If the location of such objects re-enters the camera view, saliency input seamlessly replaces the resting level boost. If no saliency input exists at this location (e.g., due to a removal of the object), the working memory links loose an essential contribution to the self-sustained regime. As a consequence, such links automatically dissolve. See [196] for an experiment probing this maintenance mechanism.

Results

The middle and right columns of Figure 4.16 show slices of activation in space-color working memory of a preliminary version of the scene representation architecture (presented in [197]). Peaks at 'red' and 'blue' are tracked according to the current camera input. The expanded saliency input influences field activation at all three salient locations, but remains below threshold if no working memory exists for a given object.

Discussion

Tracking of object identity is possible using a spatial updating signal that does not depend on attentional selection of objects. Tracking has limits arising from the decrease in overall activation of moving activation peaks and accumulating inhibition for each additional working memory peak (discussed in more detail in [197]). This is in alignment with psychophysical evidence of human tracking capabilities [31, 131, 132], showing a similar decrease in performance for increasing number of tracked items and increasing speed of objects. The trade-off between quasi-binary location tracking (i.e., objects are either targets or distractors) and identity tracking is not explored in the present experiment, but may be explained with the distributed nature of the representation. To decide if an object is a target or a distractor, only a single space-feature working memory entry is sufficient, while remembering the identity of an object may require a full coverage of all represented feature channels or even high-dimensional markers such as identity labels linked to space. Loosing parts of the feature description is more likely than loosing the full memory entry for a given object.

Pylyshyn [131] discusses that a decrease in tracking the identity of objects originates in identity swaps of two target objects passing close to each other, which is more likely for smaller distances between objects. Fewer swaps are observed for target-nontarget pairs. The interaction between represented target objects resonates well with representing objects as activation peaks

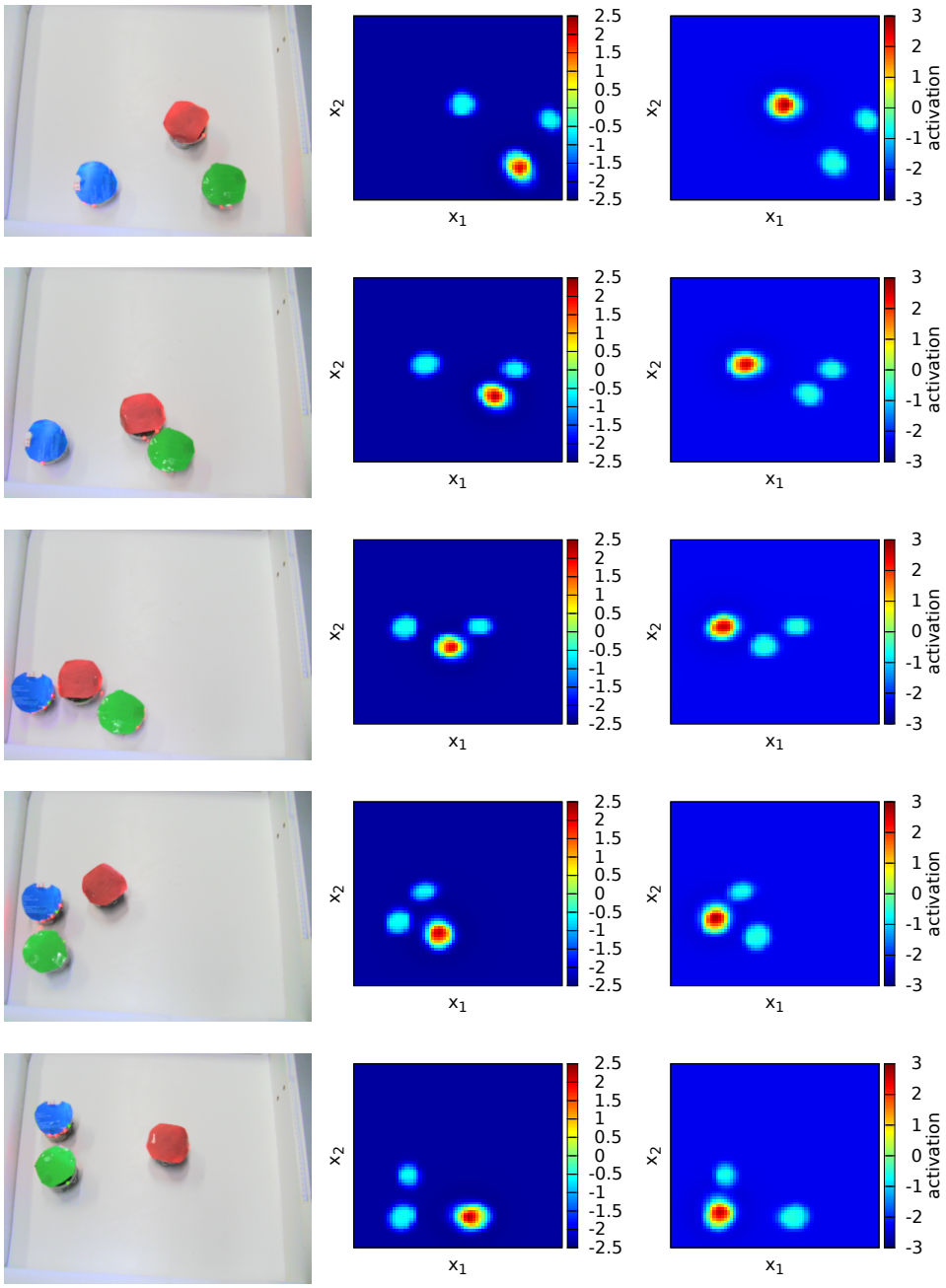

Figure 4.16: This figure shows the camera input of a scene containing three colored moving robots (left column) and two slices of space-color working memory activation (middle column: red, right column: blue) over time (from top to bottom). Working memory tracks the positional changes of the memorized colors. The green object never entered working memory and thus is not tracked.

along spatial positions. Activation of space-feature links may spill over to other close-by sites if peak distance is small.

4.2.4 Query

Queries use available working memory and the current input stream to efficiently localize a target object given cues describing its characteristics. Using the scenes assembled in Section 4.2.1, this experiment evaluates the query behavior for scenes with several objects (see Figure 4.8). Both single and combined cue paradigms are evaluated in this experiment. For single cues, performance of bottom-up query, top-down query, and a combination of both is evaluated on the previously introduced data set of static scenes by giving a color cue ('red', 'yellow', 'green', 'blue') and recording the activation of the CoS and CoD nodes of the query behavior for the first fixation after cue. Top-down and combination queries are preceded by 15 seconds of free exploration. After exploration, the bottom-up saliency is turned off for the top-down only case and IoR is reset. The following four cases may occur:

1. an object of cued color is in the scene and query CoS becomes active during first fixation

2. an object of cued color is not contained in the scene and query CoD becomes active during first fixation

3. an object of cued color is in the scene and query CoD becomes active during first fixation

4. an object of cued color is not contained in the scene and query CoS becomes active during first fixation

The first two cases are considered as correct behavior. The second two cases are considered errors. The autonomy of the query behavior creates a sequence of fixations as long as no matching object entered the attentional focus or the currently focused object does not match the cue. Thus, Cases 3 and 4 may, over time, exhibit desired behavior if the query is allowed to continue onto new candidates or revise the classification for the focused object.

To demonstrate the combined cue query behavior I use scenes in which the given set of cues is not contained at the start of the experiment. This probes if the behavior is able to correctly detect the condition of dissatisfaction for each inspected object. At a given point in time, a cue is removed. Now the behavior should be able to find a matching object and halt the behavior with an active condition of satisfaction node.

Table 4.1: Evaluation of first fixations during bottom-up color query.

		search result	
		CoS	CoD
Object	present	40%	23.33%
	absent	1.67%	35%

Results

Cues affect bottom-up and top-down saliency differently. Bottom-up retuning increases the peak width of possible target candidates, top-down queries rely on detection decisions (see Figure 4.17 for exemplary saliency activation). Quantitative results for the single feature color queries are listed in Table 4.1 for bottom-up search and in Table 4.2 for top-down search. Bottom-up search correctly fixates a target if it is present or a distractor if no target is present in 75% of first fixations. 23.33% of first fixations land on distractors although a target is contained in the scene. False alarms for targets in arrays in which the target object is missing happen in 1.67% of first fixations. Split up into target present and absent trials, bottom-up search performance finds the target object in 63.16% of first fixations. Since the search continues after the CoD node signals a mismatch, the target object may be found during subsequent fixations. Purely top-down search shows a similar performance. 76% of all trials either successfully fixate a target object or time out due to an absence of targets. 20% of first fixations run into a time-out although at least one target object is present. The CoS node activates in 3.33% of all fixations despite having no target object in sight. Table 4.3 shows that using a combination of bottom-up and top-down search improves overall performance.

Figure 4.18 shows an exemplary time course of activation for a combined cue query. At first, all selected candidates match at most one of the given cues, which leads to a repeated selection once the condition of dissatisfaction in the mismatching feature channel is activated. The query behavior is able to select a suitable candidate after one of the cues is removed. This is expressed through an activated condition of satisfaction node.

Discussion

Single feature queries find the target on first fixation in around two-thirds of target present trials, using bottom-up or top-down influences separately. For bottom-up search, the intrinsic saliency of objects plays a role in fixation

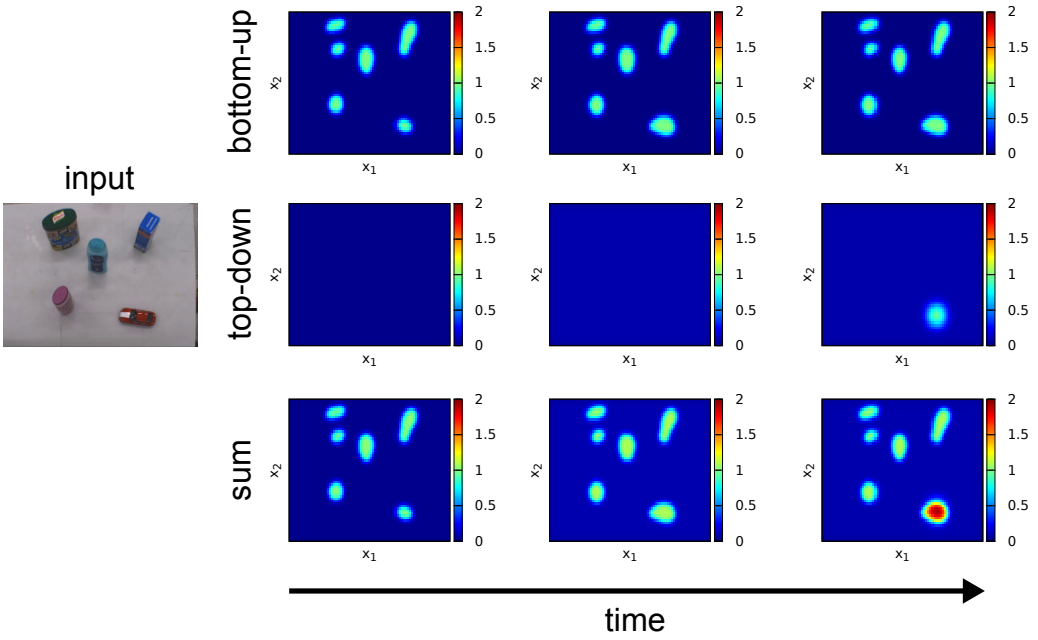

Figure 4.17: Saliency retuning during single-feature query for the color *red* affects the bottom-up (top row) and top-down (middle row) saliency contribution, highlighting the position of the red car. The combination of bottom-up and top-down saliency (bottom row) shows a competitive advantage for the target object in comparison to the distractor objects.

Table 4.2: Evaluation of first fixations during top-down color query.

		search result	
		CoS	time-out
Object	present	43.33%	20%
	absent	3.33%	33.33%

Table 4.3: Evaluation of first fixations during color query.

		search result	
		CoS	CoD
Object	present	48.33%	15%
	absent	3.33%	33.33%

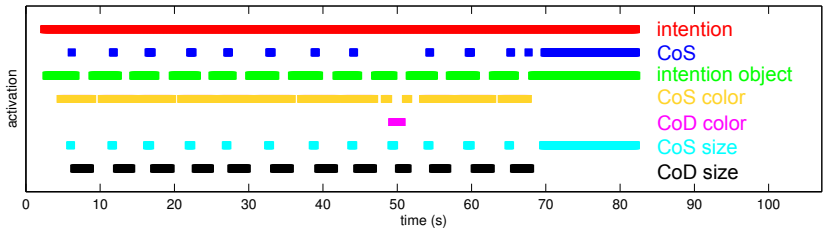

Figure 4.18: Activation of all involved nodes. The CoD nodes turn on as long as the cues do not match any object in the scene. Once the color cue is removed, the query behavior settles onto a matching object.

placement. If distractors are significantly more salient than target objects, the first fixation picks a distractor instead of the target. This phenomenon is encountered in psychophysical experiments as well, as size cues for small target objects compete with the intrinsically higher saliency of large distractor objects (see [113], experiment 3). For top-down searches, intrinsic saliency decides whether a working memory entry is created in the 15 seconds of free exploration. For both conditions, the width of the given color cue also factors into performance. Since cues are categorical with peak width encoding a range of possible matching grades of cued color, target objects whose features lie in the weaker regions of the cue peak are highlighted less strongly, while at the same time salient distractors whose color is close to the cue receive an additional boost. A combination of bottom-up and top-down search yields better performance on first fixation, as intrinsic saliency of distractors is less influential if a working memory link biases attention towards a less salient, but better matching target object.

4.2.5 Visual Search

To investigate if the proposed architecture shows a parallel or serial visual search behavior, I replicate an experiment done by Hamker [62]. In this experiment, random image displays containing one target object and several distractors are presented to a visual search model. The distractors may share one feature value with the feature values of the target along two feature dimensions. If this is not the case for all distractors, this is a feature search. If some distractors share the feature values of one or the other dimension, this becomes a conjunction search. Cues of the target feature values are given to the model. Over the trials, the dissimilarity of feature values not shared with the target is varied gradually. The conjunction search images consist of one target, two distractors sharing the target feature value for each

feature dimension (four distractors partially matching the search cue in total) and one dissimilar distractor. The feature search images contain one target and five dissimilar distractors. For each trial, Hamker measures the time to first fixation and whether the target is fixated. With increasing similarity between feature values of target and distractors, his model takes longer to decide and makes more errors. The same is true if feature search is compared to conjunction search with conjunction search being slower and making more errors.

I produce similar arrays for my architecture, consisting of colored circles and ellipses, with color and aspect ratio being the two feature dimensions through which the target is defined. All images are of size 640×480px. The circles have a radius of 20px and thus an area of $400\pi\text{px}^2$. The ellipses have the same area, but a variable length of the axes, resulting in different aspect ratios. Target and distractor color hue values differ in a varying amount of degrees. The target and distractor positions are drawn from a uniform distribution. Positions that overlap with previously placed items or are only partly visible (i.e., placed at the image borders) are discarded and redrawn. This has a subtle effect on the uniform distribution, as subsequent items are more likely to be placed on the borders and corners of the valid region of the image. Since my architecture has no center bias, this is not a restricting factor. Image generation allows to create single feature search arrays as well as arrays for conjunction search. Figure 4.19 shows sample input images for color, aspect-ratio, and conjunction arrays. Once a valid image is generated, I let the query behavior pick an item and record the time it took for successfully establishing the selection decision and also monitor if the field selects the target or any distractor. I vary the feature distances in feature and conjunctive tasks and the amount of top-down influence available to the selection process: the condition *bottom-up only* (BUO) uses only intrinsic bottom-up saliency with no retuning. The condition *retuned bottom-up* (RBU) feeds feature cues defining the target features into the bottom-up processing pathway. The last condition, *bottom-up and top-down* (BAT) combines the bottom-up votes with the query results from the scene memory, which also receives the target feature cues. Scene memory is initially filled during 30 seconds of free exploration preceding the query.

Results

Figure 4.20 shows the percentage of first fixations that land on the target object. The left column shows the percentages sorted by the three search conditions. For feature searches, performance is above chance level (which is at 16.7% for arrays of six objects). Bottom-up retuning and top-down

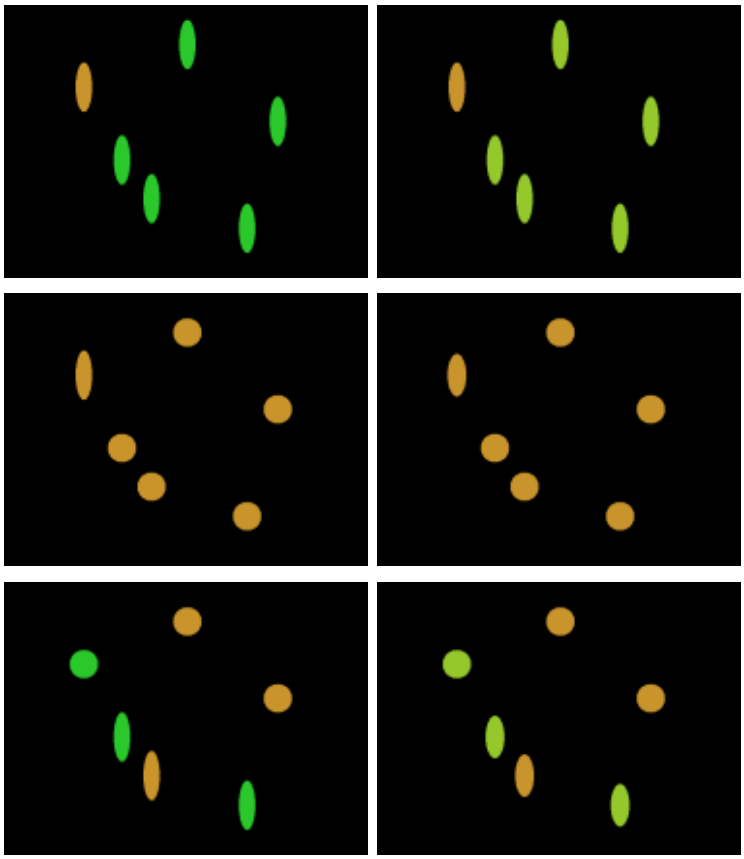

Figure 4.19: The rows show exemplary inputs to the architecture for color searches (top), aspect-ratio searches (middle), and conjunction searches (bottom). The left column has a large target-distractor feature distance in the queried channel(s), while the right column has a low target-distractor feature distance. In all examples, the target object is an orange ellipse, while green ellipses and circles of any color are considered distractors.

influence further improve performance. In color searches, performance in arrays with large target-distractor difference is better than in arrays with small difference. This is not the case for aspect-ratio searches, pointing at a change in intrinsic saliency as the target shape is altered (see Figure 4.19). Purely bottom-up performance in conjunction searches is at chance level. Cues significantly increase performance in conjunction searches, but performance remains below the one in cued feature searches (see Figure 4.20, right column). Working memory does not further improve feature searches, as performance is already at around 100% for retuned bottom-up searches. For color searches, using working memory decreases search performance if target-distractor feature distance is small. For conjunction searches, there is a slight performance increase compared to the retuned bottom-up condition.

Discussion

Hamker [62] hypothesized that the two modes of operation observed in visual search—parallel and serial—are, in fact, the outcome of search array design and a machinery with parallel biasing and sequential selection of candidates. In this experiment, the conditions yield a broad spectrum of performance based on the target-distractor distance and the type of search (feature or conjunction). Intrinsic saliency achieves above-chance performance in the two feature searches, which can be attributed to the normalization operation along the saliency pathway. Since all distractors share the same feature value, their intrinsic saliency is lower than the saliency of the target object. In feature search, performance saturates if an additional feature cue specifies the target object. Any additional biasing originating in the scene memory cannot further increase performance. Feature cues and scene memory have a clear effect on conjunction search performance. Target-distractor similarity has an impact in all three search array types. For smaller target-distractor distances, cues and scene memory queries highlight distractors as well as the target. The sequential attention process thus selects the target object with lower probability. The aspect-ratio feature channel shows a less clear effect of target-distractor feature distance. An explanation of this effect may lie in a change in intrinsic saliency, as target objects with aspect ratio closer to 1 better fit the symmetric lateral interaction and projection kernels along the processing pathway. The correlation of intrinsic saliency and values along certain feature dimensions is also observed in experiments [113].

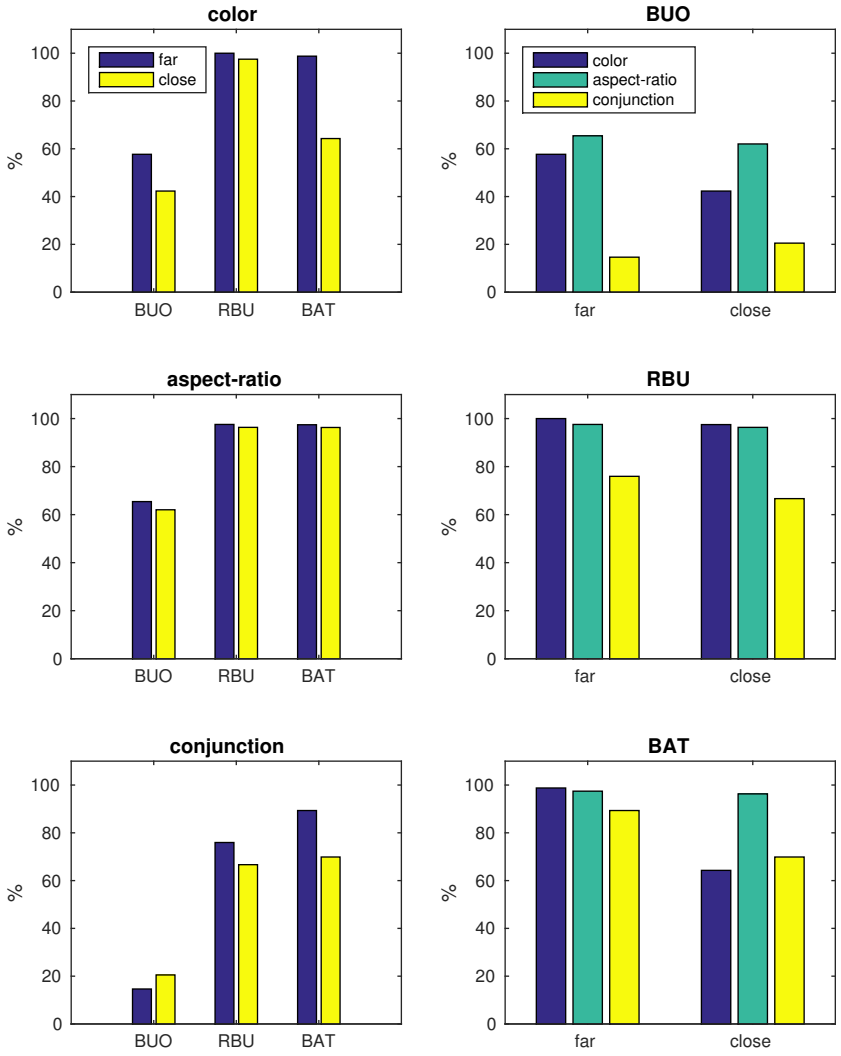

Figure 4.20: The left column of this plot shows the percentage of first fixations that land on a target object for the three types of searches (color, aspect-ratio, and conjunction search). The right column shows the same percentages, but sorted by cuing conditions (bottom-up only, retuned bottom-up, bottom-up and top-down) and target-distractor feature distance (far, close).

4.3 Discussion

Probing the behavioral characteristics of the architecture showed a general alignment with the behavioral signatures of human scene representation. The exploration behavior autonomously inspects salient regions in each scene and builds up a representation that, over time, covers an increasing amount of salient regions while keeping the representational error in feature space constant. The maintenance behavior detects feature inconsistencies while attention rests on an object and autonomously replaces the outdated representation. Changes in spatial position are tracked to a certain degree without the need of attentionally selecting an object, while simultaneously carrying along the feature representations. Queries for target objects specified by feature cues are executed using both bottom-up and top-down influences. Attentional selection of candidate objects autonomously compares cues and object features and selects a new candidate if mismatch is detected. The internal representation improves the performance of queries, as it relies on detection decisions as means to highlight candidate objects, while bottom-up retuning introduces graded changes in activation patterns that have to traverse through several sigmoid functions and have to compete with differences in activation induced by intrinsic saliency.

4.3.1 Comparison to Other Models

I closely follow the structure of Hamker's model [62, 64] by implementing saliency processing and visual search in neural substrate. Additionally, my model features working memory of the scene and autonomous behaviors that manage its content. I will first review the model by Hamker before pointing to individual differences.

Feature values are represented as maps over space in V4, which are retuneable given a cue originating in PFC and passing through TE (or IT [62]), before reaching V4. The activation of all feature maps in V4 is integrated in a perceptual map which is attributed to either PP [62] or FEF [64]. A premotor or decision map in FEF picks one candidate location as a plan for an eye movement (while not necessarily executing a saccade, but using the same neural substrate, as supported by the premotor theory of attention [141] with controversy [164]). This selection projects back into TE and V4, resulting in a feature description of the object in the attentional focus. A comparison of feature extraction and expectation is matched in PFC and ultimately leads to a release from fixation if the match fails.

My model follows the same dorsal structure: from feature maps (early space-feature fields), retuneable by expectations (feature cue fields), to salien-

cy maps (conspicuity and saliency fields), coalescing in a single decision map (attention field). The ventral pathway of feature extraction through decision feedback (feature fields) and comparison of expectation and extraction to detect matches (match detector) also bears resemblance to the model by Hamker.

The top-down influence on the early space-feature fields in my architecture is additive, which is a DFT-conform implementation of a gain control of map response. If there is no matching peak for the given cue, the additive input does not lead to the emergence of a peak, which in turn does not alter the saliency of all non-matching location. If there already is a peak, the additive boost strengthens the peak and increases the global inhibition in the whole field, effectively reducing the saliency of other competing sites. The saliency of a matching location is thus increased. If multiple items are highlighted by a cue, the inhibition across each feature slice lowers the saliency of these items. Non-matching sites may still have a higher level of activation due to differences in intrinsic bottom-up saliency and inhibitory normalization through increased activation at several locations of the map. Such phenomena cannot be observed in idealized artificial arrays and are an insight from using real camera input. My model does not extract a feature description of the foreground item through spatial feedback into the early space-feature fields directly, as in Hamker's earlier model [62], but uses readouts of the early space-feature fields weighted with the activation of the attention field and summed over space. This is not a qualitative difference to the mechanism proposed by Hamker.

The match between extracted and expected feature values (see [63] for a detailed description) is achieved through additive inputs into *no match fields* in my architecture. Hamker's model uses a measure of similarity to determine the activation level of the match neurons, while my model features competing nodes that explicate the match and no-match condition given the activity in the *no match field*. While this is again no qualitative difference, my model re-uses the match detector setup during scene exploration. The peak width and amount of peaks in cue fields defines match precision. Having a small, single-peak cue expresses a precise search for a specified feature value, while broader cue peaks are of a categorical nature that tolerates small deviations from the feature value specified with the peak center.

My architecture adds a second source of top-down influence on saliency—the bias originating in space-feature working memory. As Hamker pointed out in reference to an earlier model by Hoffman [74], visual search is split up into two stages: a parallel influence of feature attention and a subsequent serial processing of candidates picked from the first stage's map (in alignment with *feature integration theory* [180]). As Hoffman points out, the first

stage produces maps with a low signal-to-noise ratio, which is attributed as a source for the steep search slopes usually found in conjunctive searches, in which partial matches of cue and distractor objects' features raise the saliency of non-target objects. My architecture increases the distance between target and distractor saliency by not only implementing a retuning of bottom-up saliency (which also highlights distractor objects, as pointed out by Navalpakkam and Itti [116], and thus does not speed up conjunctive search), but also adding a bias from working memory.

From the comparison to Hamker's models and the experiment presented in Section 4.2.5, I support the statement of Duncan and Humphreys [35] that there is only one mode of operation for visual search, which is serial in nature. Through an appropriate choice of search arrays and other procedural details, visual search might appear to be parallel in nature [180] or not being tuneable by cues [113], but these are degenerate cases of the serial paradigm. For the parallel case, target objects are unique, that is the distance of saliency between target and distractors is high, thus having a competitive advantage in selection of the attention field (i.e., the first fixation on the display is on the target). Cases in which top-down retuning seems to fail might be caused by an overall optimal array for search (i.e., a cue adds additional saliency to already sufficiently salient targets) or distraction through using cue-matching distractors [116].

A related model based on DFT is presented by Fix and colleagues [49, 50]. The basic model presented in [50] features a saliency map, which feeds both a focus map and working memory, which is only updated at focused positions. A memory anticipation process predicts the content of working memory given a saccadic eye movement. With this group of dynamic fields, the model achieves an autonomous exploration behavior for a given visual scene through a switching mechanism, biased by inhibition of return. The working memory of the scene is purely spatial in the basic model. Since the saliency extraction path is not included in the model, there is no way of introducing feature cues to retune saliency.

An extension of the basic model [49] introduces such feature cues. They are of a conceptual nature (two colors and two orientations) and represented by discrete nodes instead of continuous fields. They influence their respective feed-forward filters located in the sensory pole. At the same time, feature extraction at the current focus of attention is realized in the same fashion as in Hamker's model by projecting back activation of the focus field to the feature maps. This extension sketches a visual search behavior, with strong emphasis on the premotor theory of attention [141]: the focus field is used to execute covert shifts of attention, which are made overt if the comparison of extracted feature values matches the cues given to the model. My model

can be seen as an extension of this model, which covers its behaviors of visual exploration and search and adds space-feature working memory and an explicit representation of behavioral organization. Overt attention shifts are not covered by my model, but can be connected to the CoS node of the query behavior to yield a similar behavior of only executing a saccade if the target fits the given cues.

A combination of visual search and scene representation in an integrated model is presented by Navalpakkam and Itti [116]. Their model consists of the following stages: first, the relevance of locations in the image is determined given a task description, which defines what to look for in a given scene. This is achieved by a similar retuning process of low-level feature maps as in the models presented above. The attentional focus of a winner-take-all selection is then recognized using the same low-level features (corresponding to the matching operation of Hamker's model). The level of task relevance of recognized objects is subsequently stored in a dedicated map if relevance exceeds a threshold. Additionally, the visual features are also stored and links to symbolic representations are created. Using low-level features for bottom-up processing, top-down cuing, object representation, and recognition is in alignment with my model. The storage of low-level visual features is not included in their implementation. My model uses space-feature working memory and I argue that this sort of memory helps in reinstantiating previously inspected objects, which is not necessary for Navalpakkam's and Itti's model, since there exists a symbolic representation of objects linked to a low-level feature description.

The authors discuss the efficiency of conjunction searches given their model and in comparison to work by Rao and colleagues [133]. They conclude that using separate maps of low-level features and normalization operations as basis for saliency not only highlights the target object, but also partly-matching distractors. Through normalization, the target object has no competitive advantage and thus an efficient search is not possible, requiring serial inspections of all matching items. Rao's model does not include this restriction, as all locations are weighted with the Euclidean distance between cued and perceived features across multiple scales. With this approach, a target object specified through a conjunction of cues quickly (i.e., in the first fixations) becomes most salient and is consequently selected by the attention process. This contradicts empirical evidence [180].

Visual search can also be modeled using a probabilistic approach (reviewed in [15], see also exemplary work by Oliva and colleagues [123]). Saliency is considered a combination of local, bottom-up characteristics (e.g., the likelihood of a feature value appearing in an image) and top-down priors such as spatial likelihood of target objects (e.g., cars typically being on

streets) and feature likelihood (e.g., apples being likely green or red). Applying the logarithm to the localized probabilities of each contribution transforms them into a saliency-like map that is a combination of bottom-up and top-down influences (for details, see [15]). As discussed in Section 3.1.4, dynamic fields may express the degree of fit to a given cue in a graded activation patterns, which is to some degree comparable to probabilities. Using additive activation patterns that influence selection decisions is comparable to integration by using probabilities for each contribution. While the mapping onto neural substrate is straight-forward for DFs, probabilistic approaches have to be explicitly embedded in neural processes. Borji and Itti [15] discuss neural networks that implement exemplary probabilistic computations. The organization of these processes, however, is not discussed.

4.3.2 The Nature of Visual Search

Wolfe defines a set of phenomena [191], which should be considered by all comprehensive models of visual search. Here, I discuss these phenomena from the saliency and serial processing viewpoint, explaining them with inherent properties of the saliency processing.

Phenomenon A declares targets to be harder to find with increasing set size of distractors. This is valid from the saliency perspective for cues producing no increase in saliency (for example, a letter T between differently oriented letters L), as there is no way of retuning the bottom-up saliency, leaving simply more candidates for the serial processing and increasing search time.

Phenomenon B describes the discrepancy of search time between *target present* and *absent* trials. This also resonates with the saliency viewpoint, as the serial process of comparing a candidate to a cue in present trials will take on average a number of fixations half the amount of items to find the target, whereas *absent* trials need fixations on all items to be sure that the target object is not present.

Phenomenon C highlights the influence of feature similarity between target and distractor items. Search is harder if target and distractors are closer in feature space. Retuneable saliency gives an explanation for this phenomenon as well: if feature cues are presented as Gaussian distributions along the feature space, retuning the saliency maps for a given feature value also highlights the distractors with similar feature values, adding more candidates to the serial processing queue. The example given by Wolfe for Phenomenon C uses size as target feature. From the saliency perspective, this has an additional influence on bottom-up saliency, which further facilitates the search.

Phenomenon D states that the more heterogeneous distractor objects are in feature space, the harder the search for a target object gets. This is covered by the saliency viewpoint with normalization operations that decrease the saliency of non-unique features in relation to more unique ones. Similar distractors are not as interesting as nearly unique ones. A sequential processing then has to detect the uniqueness of a target in relation to distractors that may appear twice in the array.

Phenomenon E describes flanking distractors, that is distractor feature values deviating in two directions from the target feature make it harder to find the target than distractors deviating only in one direction of the feature space (for example, a target with vertical orientation and distractors that either all deviate clockwise from vertical orientation or deviate both clockwise as well as counter-clockwise). This again can be accounted for by Gaussian cue distributions and normalization along similar items. Having distractor items only from one side of the feature space in relation to the target's feature value decreases the saliency of the distractor objects.

Phenomenon F summarizes search asymmetries found by researches (e.g., searching a bar with orientation 0° between distractors of orientation 15° or vice versa, with the latter being the harder case). Rosenholtz discusses that some of these findings have to be attributed to asymmetric experimental setups [143], which can be detected by applying a basic saliency model. Factors inducing experimental asymmetries to search arrays are, among other things, the background saturation and the reference frame of the presentation monitor.

Phenomenon G distinguishes between feature differences on a categorical level: finding, for example, a steeply oriented bar among distractors with shallow orientation in contrast to finding the steepest bar among steep and shallow non-unique distractors. From the saliency perspective, having distractors matching the feature category of the target might reduce the search space onto one of the two categories, but still require a serial search in the target category. It is thus related to feature similarity of Phenomenon C. The concluding phenomenon G describes the guidance through feature cues defining a subset of items in the array. For example, searching for a white letter T among black and white letters L is easier than having target and distractors in black. This translates onto saliency by again highlighting a subset of items in the array (of white color), thus reducing the amount of serial inspections necessary to find the target. The beneficial influence of a preselection of candidates was also found by Egeth and colleagues [37].

As a summary, the saliency viewpoint and two-stage processing covers the set of phenomena in visual search defined by Wolfe.

Wolfe's model of guided search [191] brings him to the conclusion that vi-

sual search is a two-stage process (supporting the hypothesis of Hoffman [74]), with a limiting serial bottleneck. He presents four arguments that are in alignment with the models presented above and my own model. The first argument states that targets might be easy to identify (with a matching operation), but hard to find, due to a lack of tuneable feature maps representing target characteristics. This becomes clear when defining targets through shapes such as letters, which require more complex processing than basic features such as color. The second argument emphasizes the inherently serial mechanism of eye movements, which are not required for executing a visual search, but may lay the foundation (i.e., the neural substrate) for sequential processing of items (as in Hamker's and Fix's models) and improve the performance of the parallel processing of preattentive features. The third argument focuses on binding, that is, combining different feature channels into a consistent item through an attentional window, as proposed by feature-integration theory [180]. From the perspective of representing different features in distinct neural populations—a perspective shared by all models presented above—the need for an attentional mechanism arises naturally to prevent illusory conjunctions. The final argument justifies the need for a bottleneck of serial processing by the observation of change blindness in humans. Changes in the scene are noticed more often if locations of change were visited by the attentional focus before introducing the change (likely creating a working memory representation) and are revisited after the change [78].

4.3.3 The Role of Working Memory in Visual Search and Change Detection

The autonomous creation and maintenance of an internal representation of a scene is the core purpose of the presented architecture. Internal representations have no intrinsic purpose, as the environment itself offers a more detailed and current presentation of target objects. Scene representation thus has to be judged as contribution to other cognitive processes. As discussed at length above, visual perception serves the purpose of efficiently finding objects relevant to the current task. The internal representation supports this task. This support has an evident effect in search scenarios that are challenging (e.g., feature conjunction search). In addition, internal representations for complex features that are not directly extractable from the input stream allow to broaden the search options and speed up search. For example, searching for a pen may be translated to low-level cues such as 'small' and 'elongated' to tune bottom-up saliency and potentially to highlight a

multitude of candidate objects, or to a query directed at a space-category working memory containing links of pens perceived at locations in space. The latter option effectively turns a slow, sequential search, into a search that picks a location containing a pen on first fixation, as long as it has been perceived before. This improvement comes at a price, as outdated working memory may highlight candidate objects that do no longer exist in the visual scene or may have changed since their last attentional selection. This emphasizes once more the need for continuous maintenance of the internal representation.

The maintenance of the internal representation leads to the detection of changes and subsequent adjustments to the representation. While this mechanism is necessary for any kind of internal representation, it has an additional beneficial effect for challenging visual scenarios. The human visual system is equipped to detect motion (for example, the counterchange mechanism [73]), with motion attracting attention [1]. Influence on the attentional process in the form of increased saliency for detected motion is included in several contemporary models of saliency (see [15] for an overview and [107] for an exemplary model). This influence on attention hints at motion being behaviorally significant for a perceiving agent, be it for survival (detecting an impending predator attack), interaction with other agents (hand waving to capture attention), or other reasons. In the absence of direct visual perception of change, this attention-guiding influence is missing (as used for masking change in the change detection paradigm). However, the internal representation detects change on attentional re-entry and a transient change signal is created. This transient signal can in principle be used as a substitute for motion detection, leading to similar behavioral consequences as the direct perception of motion. How change detection and motion detection, for example in form of a DFT model by Berger and colleagues [12], interact to guide attention and influence behavior, is a potential topic of future work.

4.3.4 Giving up: The Condition of Dissatisfaction in Visual Perception

My architecture features a single CoD node, signaling that visual search has detected a mismatch between the cues describing the target object and the features extracted from the currently selected object candidate. This CoD node is necessary since challenging visual scenes increase the likelihood of putting attention on a distractor object. Reaching a state of dissatisfaction is a common theme in visual perception of which the CoD of visual search only covers a small portion. One example is the inability to extract a fea-

ture value from a currently selected object, which prevents the creation of a working memory entry and thus reaching the CoS of inspection. Think of trying to extract a shape that was never seen by the architecture before or extracting color in poor lighting conditions. Should the creation of a memory entry be skipped or should the architecture default to the closest match? Do these decisions depend on a timer and, if so, is the time to reach the CoD learned and adaptive? A second example is a visual search for an object that does not exists. In behavioral experiments, humans performing visual search in an array may report the absence of a target object when every candidate was inspected once. But does this hold in naturalistic scenes? A robot can, in principle, extend the search by moving around a room, opening containers, looking behind or under occluding objects. At which point in time can a visual search be declared as failed? One possible solution for this is to learn the condition of dissatisfaction and links to error recovery behaviors. A starting point for this could be the learning of the condition of satisfaction, as discussed by Luciw and colleagues [105]. In addition, extending the behavior-based approach by motivational drives such as frustration might yield additional sources for the activation of the condition of dissatisfaction. Future work may elaborate on raising the behavior-based approach to an approach incorporating motivation [108].

4.3.5 Saliency and Natural Scenes

In this thesis, saliency and thus the existence of proto-objects is determined by combining the contributions of three feature channels—color, size, and aspect-ratio. While this is a sufficient set to evaluate the autonomy of the presented architecture on table-top scenes with uniform background, natural scenes require a more sophisticated visual input stream to deal with challenging scenarios such as clutter, occlusion, and self-motion. Modeling saliency is in itself a broad research area, with dozens of models covering challenges posed by various data sets (see the exhaustive review by Borji and Itti [15]). The presented architecture is by no means an additional, competitive model for saliency computation, but has to solve this part in a sufficient way to model the processes of scene representation built upon this early processing in the visual pathway. Instead, it should be understood as being compatible to the prevalent use of feature maps contributing to an overall saliency map, with features being not only relevant for saliency extraction, but also for further representation, queries, and visual search. In this way, my architecture is open to expansions by additional feature channels that increase performance in visually challenging scenes. Prototypes expand the saliency pathway by a shape channel based on an object detector [172], using four basic shapes

(circle, cylinder, square, rectangle) as the set of known templates and by motion detection based on optical flow (implementation of an algorithm by Farnebäck [42] included in the image processing library OpenCV). Scene representation may use feature channels for both saliency and representation. Not all features are of a nature that supports both roles. My exploration of additional feature channels in prototypical implementations yields the following three categories: (1) Features that are immediately accessible from the visual stream and can be represented due to their low-dimensional nature and descriptiveness; examples are color, size, aspect-ratio, and shape. (2) Features that contribute to saliency, but whose representation has no purpose; motion is an example. (3) Features that are not directly accessible in the visual stream, that is, they require attentional selection and can thus not contribute to saliency, but whose representation is beneficial for queries, as there is no bottom-up retuning of visual search; representing object recognition labels is an example (discussed in Chapter 5).

Chapter 5

Applications

In this chapter, I apply the principles of scene representation in the context of object recognition and generation of goal-directed reaching movements. The applications serve as further test cases for the architecture developed in this work. In addition, the central role of scene representation and behavioral organization in cognitive processing and movement generation is highlighted.

5.1 Object Recognition

Scene representation covers low-level, pre-attentive features (color, size, and aspect-ratio), which are extracted directly from the sensory stream through filter operations. Binding a set of extracted features to proto-objects allows efficient visual search in scenes. Human vision goes beyond proto-objects, having abstract high-dimensional descriptions of objects, both in the sense of categorical labels ("this is a cup") and unique instantial labels ("this is my personal cup") attributed to them. Labels can be used to specify objects ("please hand me the screwdriver"), which suggests that visual search is equally well suited to deal with such abstract descriptions instead of low-level features.

Creating labels from visual input requires a complex machinery of object recognition, as opposed to the extraction of low-level features. A principle of such machineries is the hierarchical structure of units growing in complexity, as discussed by Riesenhuber and Poggio [140]. In their model, simple cells with narrow receptive fields produce responses to localized features of an input stream. Moving up a hierarchy of more complex cells, responses of simpler cells are combined to both broaden the receptive field and making cell responses more specialized (e.g., cells listening to a specific combination of more simpler cells detecting oriented edges to form a corner detector). At

the top of the hierarchy, cells respond to the whole input array and may produce categorical or instance responses. Cells at the top exhibit invariance to pose parameters, as their receptive fields cover the whole input array. Sophisticated hierarchical structure and training may introduce additional pose information. One such machinery is a hierarchical network of nodes based on slow feature analysis [53]. Here, object recognition happens invariant of the pose of a target object, while at the same time creating pose estimates (position and angle). Challenges for such machineries arise in complex scenes, in which multiple objects are present and background is not homogeneous.

Can scene representation enhance the capabilities of object recognition machineries and, if so, how? Attention is a shared resource between visual inspection and recognition [34] and saliency may guide recognition to interesting regions of the visual input [147]. Attention is the key to integration of scene representation and object recognition. The following sections describe how an attentional window provides an object recognition machinery with the necessary information to recognize an object at a given spatial position and how the resulting label is memorized in the same way as low-level features.

5.1.1 Steering Foveal Vision with Attention

Faubel and Schöner [43] present a neuro-dynamic object recognition machinery. Here, object recognition is based on the principle of map-seeking circuits introduced by Arathorn [3]. Estimates of transformation parameters (shift and rotation) tune a transformation chain to bring the current target object into a pose it initially had during training. At the same time, the degree of fit to a battery of training views for all known objects evolves to limit the candidates matching the target object. Pose estimate and match evolve in parallel, which leads to a recurrent circle. Multiple pose and label candidates converge onto single pose estimates and a label candidate over time. Estimates are represented in two-layer DFs, with the second layers being more selective than the first. Estimates and matches are calculated using correlation. Figure 5.1 shows a schematic sketch of the recurrent object recognition machinery for a single pose parameter (rotation).

This machinery can be enhanced by a simulated fovea [44], featuring high resolution in the center and a logarithmic decay of resolution with radial distance to the center. The motivation for introducing a foveal area is the analogy to the human eye and the resulting logarithmic influence of visual input to object recognition. Features extracted close to the foveal center contribute more to recognition than features extracted in the periphery, which are potentially taken from distractors. Movement of the fovea is achieved

with a model of saccadic eye movements by Goldberg [55]. A Hopf oscillator creates ballistic movements with Gaussian-shaped velocity profile once a target for the fovea is defined in a DF, bringing the fovea to the target. If a fixated target object moves, a second influence implementing servoing keeps the fovea on the object. Since the camera image only covers a portion of the visual scene, head movements are executed on a slower time scale, which center the gaze on the attentional foreground. During saccadic eye movements, the perception of the foveal region is switched off. This is in alignment with evidence from neurophysiology (see, for example, [144]) and psychophysics (see, for example, [18]). Although the foveal vision resembles attentional selection, as visual input has decreasing influence on the recognition the further it is away from the center of the fovea, this architecture has no means to sequentially scan a scene and recognize multiple objects.

Learning of the pattern memory happens in a one-shot paradigm. Each object is presented once to the object recognition, with fixed inputs to shift, rotation, and label fields. Patterns are extracted given this pose and are associated to the label. Multiple views per object can be learned as well, but require a label for each pose, which is then associated with object identity. On object sets of size 30 (first 30 objects of COIL-100 [118] or 30 objects of a custom data set), recognition rate is 85 % or better, with rate going up 96 % for two training views per object on COIL-100.

A benefit of using a simulated fovea doing saccadic eye movements and smooth pursuit is the capability to track moving objects with the robot's gaze, even in the presence of distractor objects. For more details, please see [44].

5.1.2 Combining Recognition and Representation

Zibner and colleagues combine the object recognition machinery and principles of scene representation in an integrated architecture [198]. Fusion happens along both the dorsal and ventral pathway. The attention field of the scene representation is connected to the saccadic motor control of the simulated fovea. A peak in this field defines the target of the next saccade, centering the simulated fovea on the attentional blob. Fields in the object recognition machinery are boosted whenever the attention field contains a peak. Recognition thus starts once an object is in the attentional foreground. The second layer of the label field contains the label candidate after convergence of object recognition. This layer is considered as one-dimensional feature input along the ventral stream, similar to color and size. Space-label links are created between the position given by attentional selection and the label candidate originating in the second layer label field of the

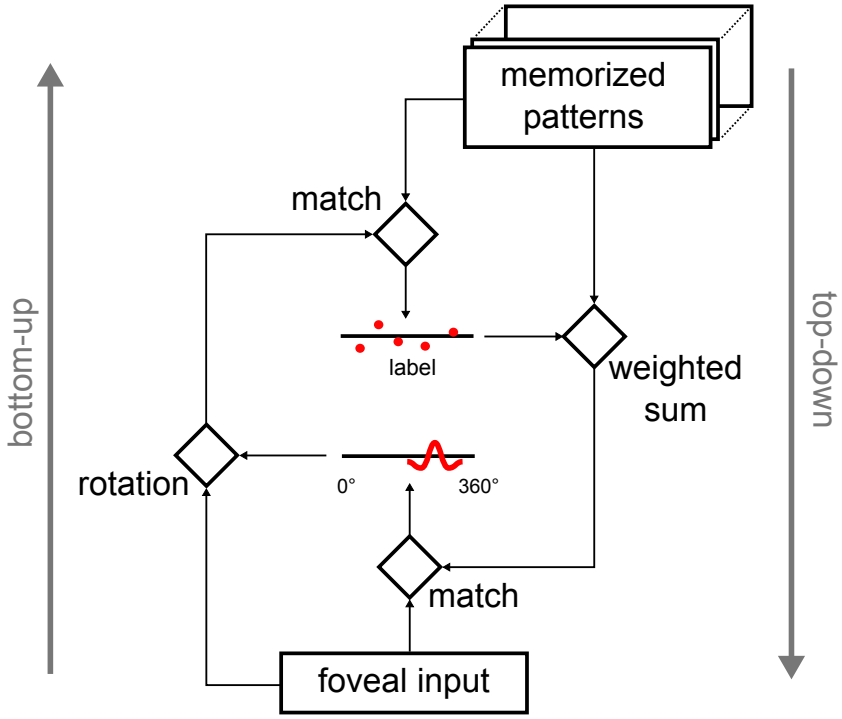

Figure 5.1: This schematic sketch shows the recurrent processing of foveal input to produce an object instance label. Feature distributions extracted from the foveal foreground are rotated with the current rotation estimate along the bottom-up pathway before comparing them with the array of memorized patterns. The degree of match with each pattern influences the activation of each label candidate. This activation is used as a weight to combine every memorized pattern to a weighted sum along the top-down pathway. This sum is matched with the current foveal input to refine the rotation estimate. Recognition and pose estimation run in parallel and converge onto single candidates. Additional stages of pose transformations, such as shifting and scaling, can be cascaded (not shown in this sketch).

object recognition.

The integration of scene representation and object recognition has multiple consequences. Through sequential exploration, the object recognition system is guided to specific parts of the scene and produces label candidates for all objects contained therein. Space-label working memory allows queries with cues that are not directly present in the bottom-up saliency pathway. In addition, labels are carried along for moving objects, which keeps this high-dimensional description available without re-applying the object recognition machinery.

5.1.3 Discussion

Object recognition is a practical application for scene representation. Both components benefit from each other. In terms of behavioral organization, the scene representation component shapes the overall behavior, as object recognition is treated as a black box with a purely feed-forward input-output characteristic from foveal image to recognized label. The recurrent connectivity in the object recognition implies that some form of behavioral organization is necessary, for example to reset the detection decisions in the involved fields between recognitions. Future work may endow this component with a detailed organization of internal processes, for example to give a full account of autonomous label learning.

5.2 Goal-Directed Arm Movements

Humans use their arms and hands to interact with objects contained in scenes. A basic form of movement is the goal-directed reaching movement that brings the hand to a target object. Being able to move the hand to an object establishes a multitude of object interactions, for example picking up food to eat it, moving obstacles out of the way, or grasping a tool to perform intricate handcraft. Visual processing of scenes serves the purpose of extracting motor parameters for the generation of such movements [155]. Object-oriented movements are intensely trained in the first year of life of human infants [173], with vision having an increasing influence in guiding arm movements [185]. Movements generated by adult humans and primates show a number of characteristics that allow insights in the brain's movement generation process.

The tangential velocity profile of the hand exhibits a bell-shaped profile, whose form is invariant under movement distance and speed [5]. This implies that movement planning happens in Cartesian space, rather than joint space [115] or a control of a subset of variables with low variance while others are uncontrolled and thus exhibit a large variance (uncontrolled manifold theory [156]). The view of controlling only relevant variables, while others are left uncontrolled is also prominent in optimal control [176]. The property of the nervous system to control parts of the movement (e.g., hand trajectory), while others are uncontrolled (e.g., joint angles) is evident in behavioral data on coarticulation in movement sequences [66]. Recordings of neural populations confirm this view, as population activation correlates with movement speed and orientation in Cartesian space in primates [26, 54, 114]. A central theme are neural oscillations as implementation of movement gen-

eration [25]. The nervous system quickly adapts to changes in target position [17, 51, 52, 54, 182], even if a change is not consciously perceived [129] (e.g., during a saccadic eye movement).

Arm movements are generated by muscles that move arm segments with 10 degrees of freedom to place the hand at a target position. Muscles are, in a rough simplification, tuneable springs, which allow a certain degree of compliance, that is, they yield to external forces. Each DoF requires a muscle pair, agonist and antagonist, to be able to change its configuration in both directions, as each muscle can only pull in one direction, but cannot generate a force in the opposite direction. A change in joint angle is induced by only one muscle generating a force. If both agonist and antagonist pull at the same time and generated forces cancel each other out, muscle activation has an effect on the stiffness of the joint, decreasing the compliance of the associated joint. The type of control employed by the nervous system to generate muscle-based movement ranges from control of force, to muscle length, stiffness, and mechanical impedance [165]. The control signal sent from the brain may either be complex, for example time series of force profiles, or simple, for example a desired resting length to which muscles contract. The latter view is called equilibrium point hypothesis [45, 46]. Depending on the degree of detail put in modeling arm muscles, equilibrium point trajectories are considered as ramps with constant slope [59] or more complex, N-shaped functions [56], with evidence for the influence of timing between agonist and antagonist shifts [102]. Independent of the shape of equilibrium point trajectories, the brain also has to translate arbitrary positions into a reference frame suitable for movement generation [47].

The generation of goal-directed movements is modeled with a variety of approaches. The vector integration to end-point (VITE) model by Bullock and Grossberg [21] and its implementation in neural dynamics [20] calculates the difference between hand position and target position. The difference vector pointing towards the target is combined with a GO signal, which takes the form of a ramp function, generating movement of the arm and updating the internally represented hand position. With an appropriate choice of function to generate the time course of the GO signal, the resulting tangential velocity profile is bell-shaped. An adaptation of VITE by Strauss and Heinke [167] using DFT replaces the GO signal with an asymmetric interaction kernel. The bell-shaped velocity profile is generated by a peak traveling from the center of a motor field outwards to a point defined by the current target, with a subsequent return to the motor field's center. Rokni and Sompolinsky [142] present a model that uses a neural oscillator with modulatable frequency and amplitude that drives a number of integrators that in turn translate the oscillatory pattern in muscle movement. Each oscillatory cycle describes one

movement segment. The oscillator is implemented as a neural population covering the circular phase space and asymmetric connection weights along the metric, which leads to traveling patterns of supra-threshold activation. The frequency of this oscillation is controlled by additive input to the neural population. Generated movements exhibit properties of human arm movements. Based on the assumption that the cortex directly controls muscle activation [175], Todorov and Jordan [176] formulate a model of movement generation that uses feedback-driven optimal control. The model deals with redundancy in terms of only correcting deviations from the relevant task.

Here, I present a neuro-dynamic process model that takes the properties of human arm movement into account. Based on attentional selection of a visual target and an internal representation of the hand position, a neural oscillator creates virtual equilibrium point velocities in Cartesian space to reach the selected target. These velocities are translated to joint space and shift the equilibrium point of each muscle. Muscles are simulated using damped harmonic oscillators. They create movement if their equilibrium point differs from their current length. An efference copy of generated equilibrium point shifts is used to update the representation of initial hand position in-between movements. Behavioral organization controls the switch between phases of equilibrium point movements and postural control.

5.2.1 A Neuro-Dynamic Architecture of Reaching Movements

Figure 5.2 shows a schematic overview of the architecture. A target in the workspace of the robot is visually perceived, using a saliency operation and attentional selection. The target position is transformed into an allocentric Cartesian reference frame. The target then enters the *reaching target* field (see Figure 5.2, A). The *initial hand position* is represented in a second field covering the same spatial reference frame (see Figure 5.2, B). The peaks in both fields are convolved to form a movement plan. With this operation, the target position is expressed in relative coordinates of the hand position, with the hand being in the middle of the resulting plan. The movement plan parameterizes a neural oscillator (see Figure 5.2, C). All positions in the oscillator are associated with preferred movement directions and amplitudes. The neural oscillator consists of two layers, one excitatory and one inhibitory, with both layers receiving the plan as input. The inhibitory layer evolves on a slower time scale than the excitatory layer, which leads to a brief emergence of a peak in the excitatory layer, which is subsequently suppressed by the inhibitory layer. The oscillator thus undergoes a one-shot transient oscilla-

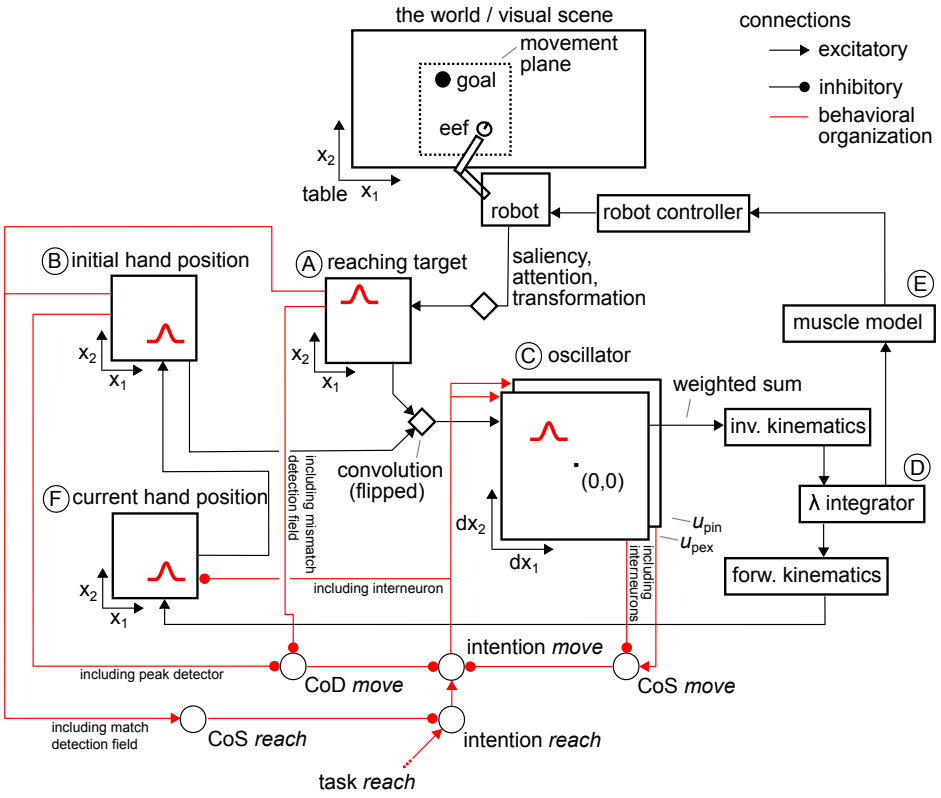

Figure 5.2: This figure shows the movement architecture. Here, field representations of target and initial hand position drive a neural oscillator, which in turn shifts the equilibrium points of the muscles (denoted as λ). During postural control, the equilibrium points are used to update the initial hand position.

tion. Figure 5.3 shows an example of such a two-layer oscillator. Weighted with the preferred directions and amplitudes, the activation of the excitatory layer is integrated to form a velocity vector for the equilibrium points, still in Cartesian space. This velocity is translated to joint space using inverse kinematics. The resulting velocity vector in joint space influences an *equilibrium point integrator* (see Figure 5.2, D). The integrator forwards its current state to the muscles (see Figure 5.2, E), which in turn generate movement if their current state differs from the equilibrium point. A second pathway feeds the integrator state back to the representation of hand position by first entering a *current hand position* field (see Figure 5.2, F), which is de-boosted during movements. During postural control, this field is boosted and updates the initial hand position field.

The behavioral organization of this architecture consists of a hierarchy of

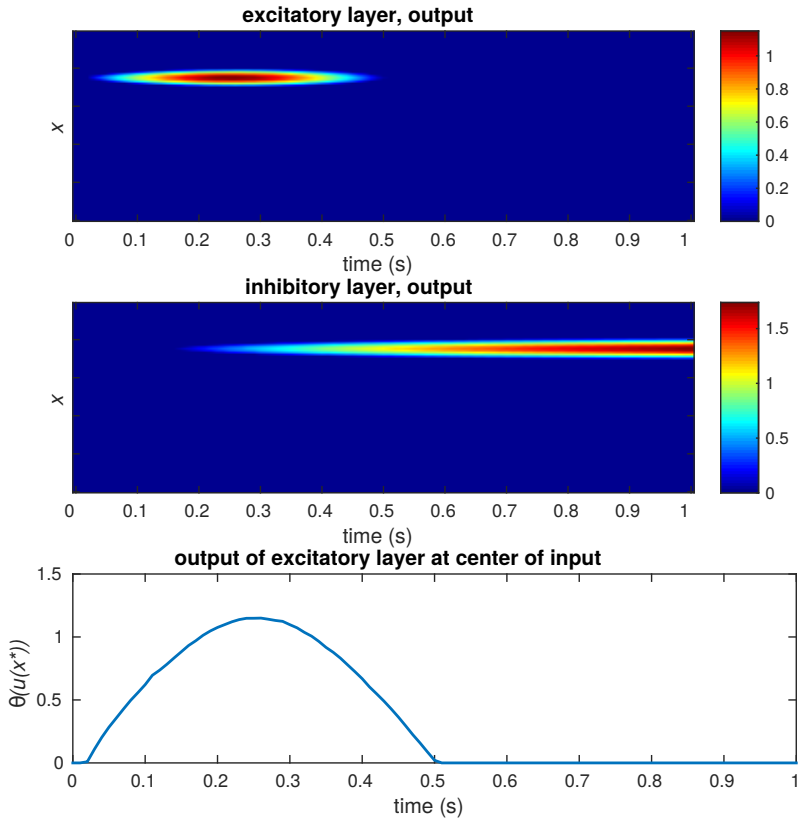

Figure 5.3: This figure shows the thresholded activation of excitatory (top) and inhibitory (middle) layers of a one-shot oscillator. Both layers receive Gaussian-shaped localized input centered at $x^* = 25$. The bell-shaped activation profile of the excitatory layer at this location is shown at the bottom.

ECUs. On a lower level, the intention to *move* the equilibrium points boosts the resting level of both layers of the neural oscillator, while at the same time decreasing the resting level of the current hand position field (using an inhibitory inter-neuron). The current hand position field goes through a reverse detection decision, which removes the input to the initial hand position field. Through lateral interactions, the initial hand position is kept in working memory. The condition of satisfaction of the move behavior is reached when the excitatory layer of the neural oscillator contains no peak and the inhibitory layer has formed a peak. These two conditions are monitored by inter-neuron peak detectors. Reaching the condition of satisfaction releases the current hand position field from inhibition, effectively updating

the initial hand position. Loosing the peak in the initial hand position field or changes in the target field trigger the condition of dissatisfaction of move, as the condition of satisfaction cannot be reached anymore. On a higher hierarchical level, a *reach* behavior activates the low-level move behavior by projecting the activation of its intention node to the intention node of move. The intention node of reach in turn is activated by task input. The condition of satisfaction of reach is triggered by a match detection between initial hand position and target position. Reaching the CoS deactivates the move behavior on the lower level by taking away the input from the intention node of reach.

5.2.2 On-Line Updating as an Emergent Property of Integration

The continuous coupling to the sensory input through saliency and attentional selection, the delay of movement generation introduced by the muscle model, as well as the autonomy of the behavioral organization of this architecture lead to an emergence of on-line updating. The attention field along the visual processing pathway tracks movements of the target. Changes in target position are detected on both levels of the movement architecture. The move behavior's condition of dissatisfaction may be triggered if the target position changes during the shift of the equilibrium points, entering the postural phase and updating the initial hand position with the current state of the equilibrium point integrator. A subsequent shift is executed as the condition of dissatisfaction deactivates. If a change in target position happens after the equilibrium point already moved to the target position, the higher-level reach behavior looses its condition of satisfaction as initial hand position and target position no longer match. The intention node of reach reactivates and activates the lower level move behavior, executing corrective shifts of the equilibrium points as long as initial position and target position do not match.

In the two-step paradigm used by van Sonderen and colleagues [182] and Flash and Henis [52], the architecture performs smooth blends of movements directed at the initial target and the updated position. Trajectories generated by the movement generation architecture for the two studies are shown in Figure 5.4, top for [182] and bottom for [52]. The tangential velocity profiles feature two peaks, with the relation of their amplitude determined by the inter-stimulus interval between presentation of the initial target and shift to the updated position.

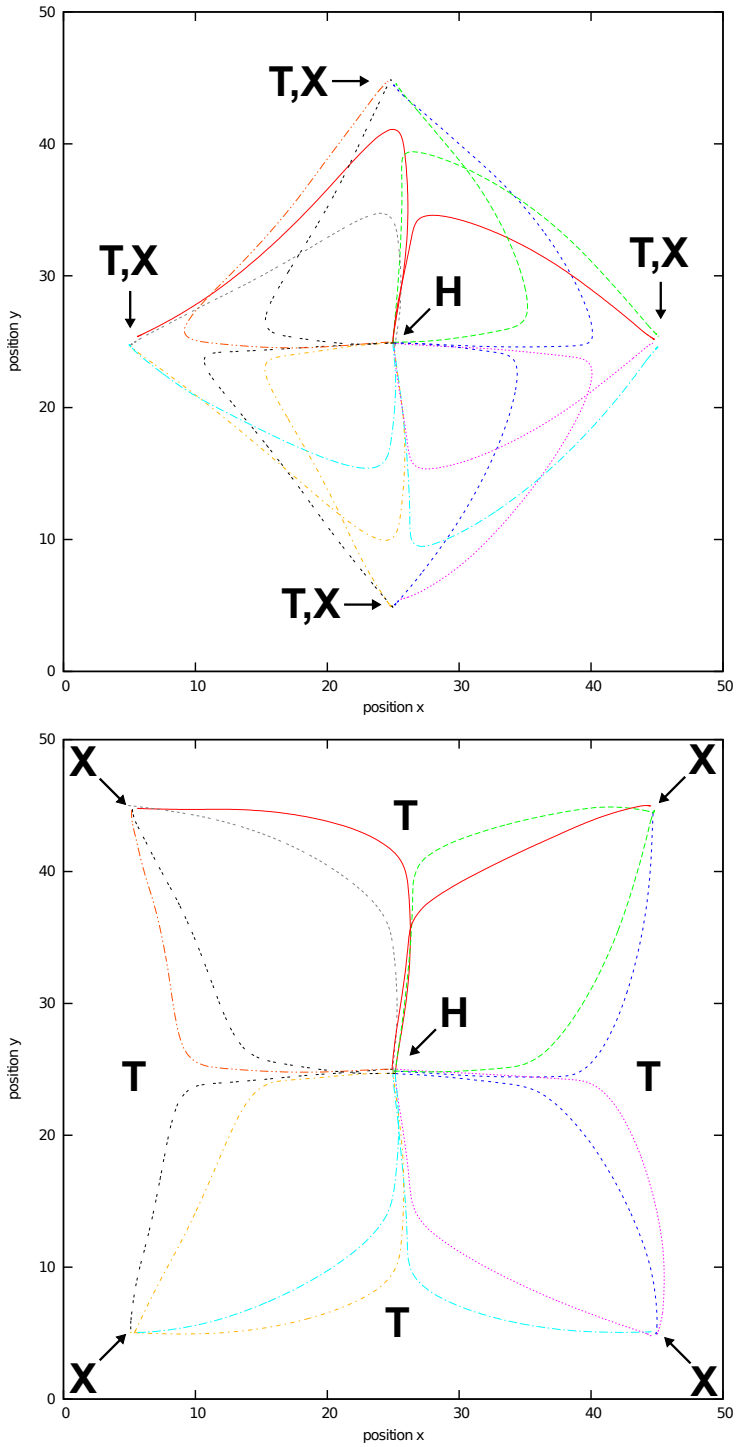

Figure 5.4: Moving the target to a new position during movement generation influences the hand trajectories generated by the architecture. Trajectories are a blend of the movement starting at point H towards the initial target (marked with a T) and the updated position (marked with an X).

5.2.3 Discussion

The presented movement generation architecture implements one of the primitives of arm movement for the robotic platforms CAREN and NAO. It moves the hand to a target position. On-line updating of reaching movements emerges as a consequence of continuous coupling to visual perception and behavioral organization of internal processes. The internal organization thus has an impact on overt behavior in challenging scenarios. Additional primitives such as lifting the hand to avoid obstacles are evident in behavioral data [60]. Their generation and integration into the transport component is as of now left open for future work.

The development of reaching movements is a test scenario for the movement generation architecture. Infant development exhibits characteristic phases, for example, an increase in movement units (amount of velocity peaks in reaching movements) and decrease in trajectory straightness [173] and a transition from uncoordinated motor babbling to coordination of eye and hand movements in time [185]. Impairing components of the movement generation architecture shows comparable effects on number of movement units and trajectory straightness [199], implying that the general structure of the architecture is present in infancy, but connectivity has to be slowly adapted based on reaching experiences. The transition from babbling to coordination is a property of behavioral organization, as preconditions between elementary behaviors develop during infancy in parallel to an overall improvement of connectivity [200].

The movement generation architecture assumes that any difference between virtual and factual position of the arm and hand is transient in nature, with the hand ending up at the target position defined by the equilibrium positions of all involved muscles. This is not necessarily true under load conditions, as additional forces distort the arm position. Humans exhibit load compensation [58], but it is not clear if this is a property of high-level planning (requiring fast sensory feed-back) or low-level servoing [165]. To fully account for distortions in the open loop control currently implemented with this movement architecture, some calibration mechanism has to be added, which takes care of adapting to differences between internal representations and factual arm configurations. This is evident from a developmental perspective, as a continuous adaptation to a growing body takes place during childhood and might still be at work in adults to adapt to changes in muscle strength caused by changes in the muscle structure.

The current state of the architecture has no account for how different movement speeds can be realized and controlled by the nervous system. The speed of the equilibrium point shift depends only on the spatial position of the

target, with a constant movement time determined by the relation between time constants τ of the neural oscillator. Different speeds can be realized by more complex virtual trajectories as discussed by Tekülve [171], with a negative impact on on-line updating performance. An alternative account of movement speeds might be possible by using a more complex muscle model, which takes into consideration the control of muscle co-contraction to increase and decrease muscle stiffness (see [59]). With increased stiffness, the delay induced by the muscle model decreases. The hand trajectory thus follows the internal virtual movement more closely, effectively increasing the speed of the executed arm movement. Decreasing the delay between internal trajectory and overt arm movement, however, leaves less time for adapting to changes in the target, which can be observed in experiments, comparing the adaptation to target changes of slow and fast moving participants [17].

Chapter 6

General Discussion and Conclusion

In this thesis I have presented a neuro-dynamic architecture for scene representation whose core components I have explored in several publications during my time as a doctoral student. Here, I integrated my research into a congruent architecture of scene representation and added mechanisms to organize the sequential structure of the three involved behaviors exploration, maintenance, and query/visual search. The experiments I presented as well as the embedding of results in behavioral data and existing models show that my neural process account delivers the essential functionality of a scene representation. Sequences of behaviors are generated autonomously and robustly. The architecture operates on real sensory inputs, perceives real scenes, and updates its internal representation on-line. My architecture shares with previously published neurally motivated models the sensitivity to the statistics of the visual input as already highlighted in the experiments and discussion in Chapter 4. Its added capacity to autonomously organize memory enables a new channel of top-down influence that accelerates visual search in challenging scenes and makes change detection possible in the absence of attention-grabbing events such as motion cues.

Scene representation always serves a behavioral purpose. To emphasize this, I have integrated the neural process model of scene representation into larger architectures that use the same neuro-dynamic framework to implement object recognition and generation of arm movements. These applications demonstrate that scene representation can be integrated into such larger architectures. I kept the presentation of applications short, as the motivation of these research lines would require theses on their own. I refer interested readers to the associated publications for an in-depth look into these research lines.

Autonomy is the prevalent theme of this thesis. I showed how principles of behavioral organization create sequences of behaviors that autonomously scan visual scenes, conduct maintenance operations, and react to external cues that specify target objects. These sequences emerge from space- and time-continuous neural processes modeled with fields and nodes of neural activation. These processes of behavioral organization reside in the same neural substrate as the scene representation itself. No external control structure violates the neural plausibility of the presented architecture. The driving force of autonomy is the detection of matches and mismatches between representations of extracted feature values and expectations thereof. The role of these matches and mismatches is defined by the context of the ongoing behaviors. They are either used to detect the successful completion of the current behavior, which is used to terminate visual search and repeat the inspection during exploration, or to trigger behaviors on dissatisfaction that maintain working memory or move on to the next candidate in visual search. Similar mechanisms are used in the extension to movement generation. The driving force here is again the comparison of representations, as virtual movements are generated as long as the estimated hand position does not match the current movement target. Behavioral organization is thus essential for the emergence of on-line updating, which is critical in dynamic scenes.

Autonomous generation of behavioral sequences opens up the architecture to neural principles of learning. Behavior generation explores the space of interactions with a scene. The nodes of the behavioral organization representing the conditions of satisfaction and dissatisfaction may serve as signals that trigger learning in cases of success or failure. The developmental and learning processes leading to the structure and connectivity of the discussed architectures were not within the scope of this thesis. Preliminary work inspired by data on infant reaching development resulted in the conclusion that behavioral organization plays a crucial role in early stages of development, as its autonomous nature triggers corrections when errors in connection weights and transformations impact on behavior. Reaching behavioral goals despite existing errors is a trigger for adaptation, from which future repetitions benefit.

In this thesis, I mainly discussed behavioral organization as a mechanism of managing the inner workings of behavioral modules. This can be scaled up to organization of more complex behaviors that make use of multiple behavioral modules. In a master thesis that I supervised, Knips [91] developed an integrated neuro-dynamic architecture of grasping novel objects by combining scene representation, object recognition, and movement generation. Here, behavioral organization not only controls processes within each module, but also manages the sequential activation across modules. Initially, a

grasping target is defined by a color cue, which leads to a visual search and subsequent attentional fixation. Once the condition of satisfaction of visual search is reached, object recognition starts matching a set of known elementary shapes to the target object, while simultaneously estimating its pose. Convergence of these concurrent processes triggers movement generation, which first brings the hand close to the object and opens it before closing the fingers around the object and lifting it up. The performance of the resulting architecture was tested on the robotic platform CAREN [92, 93]. Using the continuous coupling to sensory input as described above, grasp movements dynamically adapt to changes in object pose.

Bibliography

[1] Richard A. Abrams and Shawn E. Christ. Motion onset captures attention. *Psychological Science*, 14(5):427–432, 2003.

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

[3] David W. Arathorn. *Map-seeking circuits in visual cognition: A computational mechanism for biological and machine vision.* Stanford University Press, 2002.

[4] Ronald C. Arkin. *Behavior-based robotics.* MIT press, 1998.

[5] Christopher G. Atkeson and John M. Hollerbach. Kinematic features of unrestrained vertical arm movements. *The Journal of Neuroscience*, 5(9):2318–2330, 1985.

[6] Edward Awh, John Jonides, and Patricia A. Reuter-Lorenz. Rehearsal in spatial working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 24(3):780–790, 1998.

[7] Edward Awh, Edward K. Vogel, and Sei-Hwan Oh. Interactions between attention and working memory. *Neuroscience*, 139(1):201–208, 2006.

[8] Dana Ballard, Mary M. Hayhoe, and Jeff B. Pelz. Memory representations in natural tasks. *Cognitive Neuroscience, Journal of*, 7(1):66–80, 1995.

[9] Paul M. Bays and Masud Husain. Dynamic shifts of limited working memory resources in human vision. *Science*, 321(5890):851–854, 2008.

[10] Paul M. Bays, Emma Y. Wu, and Masud Husain. Storage and binding of object features in visual working memory. *Neuropsychologia*, 49(6):1622–1631, 2011.

[11] Momotaz Begum and Fakhri Karray. Visual attention for robotic cognition: A survey. *Autonomous Mental Development, IEEE Transactions on*, 3(1):92–105, March 2011.

[12] 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. In Alessandro E. P. Villa, Włodzisław Duch, Péter Érdi, Francesco Masulli, and Günther Palm, editors, *Artificial Neural Networks and Machine Learning — ICANN 2012*, volume 7552 of *Lecture Notes in Computer Science*, pages 579–586. Springer Berlin Heidelberg, 2012.

[13] Mårten Björkman and Danica Kragic. Combination of foveal and peripheral vision for object recognition and pose estimation. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 5135–5140. IEEE, 2004.

[14] Nico Blodow, Dominik Jain, Zoltán-Csaba Márton, and Michael Beetz. Perception and probabilistic anchoring for dynamic world state logging. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 160–166. IEEE, 2010.

[15] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):185–207, 2013.

[16] Timothy F. Brady, Talia Konkle, George A. Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

[17] Eli Brenner and Jeroen B. J. Smeets. Fast responses of the human hand to changes in target position. *Journal of Motor Behavior*, 29(4):297–310, 1997.

[18] Bruce Bridgeman, Derek Hendry, and Lawrence Stark. Failure to detect displacement of the visual world during saccadic eye movements. *Vision Research*, 15(6):719–722, 1975.

[19] Rodney A. Brooks. *Cambrian intelligence: the early history of the new AI*, volume 97. MIT Press Cambridge, MA, 1999.

[20] Daniel Bullock, Paul Cisek, and Stephen Grossberg. Cortical networks for control of voluntary arm movements under variable force conditions. *Cerebral Cortex*, 8(1):48–62, 1998.

[21] Daniel Bullock and Stephen Grossberg. Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95(1):49–90, 1988.

[22] Claus Bundesen. A theory of visual attention. *Psychological Review*, 97(4):523–547, 1990.

[23] Timothy J. Buschman and Earl K. Miller. Serial, covert shifts of attention during visual search are reflected by the frontal eye fields and correlated with population oscillations. *Neuron*, 63(3):386–396, 2009.

[24] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011.

[25] Mark M. Churchland, John P. Cunningham, Matthew T. Kaufman, Justin D. Foster, Paul Nuyujukian, Stephen I. Ryu, and Krishna V. Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

[26] Mark M. Churchland, Byron M. Yu, Stephen I. Ryu, Gopal Santhanam, and Krishna V. Shenoy. Neural variability in premotor cortex provides a signature of motor preparation. *The Journal of Neuroscience*, 26(14):3697–3712, 2006.

[27] Paul Cisek. Making decisions through a distributed consensus. *Current Opinion in Neurobiology*, 22(6):927–936, 2012.

[28] Paul Cisek and John F. Kalaska. Neural mechanisms for interacting with a world full of action choices. *Annual Review of Neuroscience*, 33:269–298, 2010.

[29] Paul Cisek, Geneviève A. Puskas, and Stephany El-Murr. Decisions in changing conditions: the urgency-gating model. *The Journal of Neuroscience*, 29(37):11560–11571, 2009.

[30] Melissa W. Clearfield, Evelina Dineva, Linda B. Smith, Frederick J. Diedrich, and Esther Thelen. Cue salience and infant perseverative reaching: Tests of the dynamic field theory. *Developmental Science*, 12(1):26–40, 2009.

[31] Michael A. Cohen, Yair Pinto, Piers D. L. Howe, and Todd S. Horowitz. The what–where trade-off in multiple-identity tracking. *Attention, Perception, & Psychophysics*, 73(5):1422–1434, 2011.

[32] Céline Craye, David Filliat, and Jean-François Goudou. Exploration strategies for incremental learning of object-based visual saliency. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, pages 13–18, 2015.

[33] Bianca de Haan, Paul S. Morgan, and Chris Rorden. Covert orienting of attention and overt eye movements activate identical brain regions. *Brain Research*, 1204:102–111, 2008.

[34] Heiner Deubel and Werner X. Schneider. Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research*, 36(12):1827–1837, 1996.

[35] John Duncan and Glyn W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96(3):433–458, 1989.

[36] Boris Durán, Yulia Sandamirskaya, and Gregor Schöner. A dynamic field architecture for the generation of hierarchically organized sequences. In Alessandro E. P. Villa, Włodzisław Duch, Péter Érdi, Francesco Masulli, and Günther Palm, editors, *Artificial Neural Networks and Machine Learning — ICANN 2012*, volume 7552 of *Lecture Notes in Computer Science*, pages 25–32. Springer Berlin Heidelberg, 2012.

[37] Howard E. Egeth, Robert A. Virzi, and Hadley Garbart. Searching for conjunctively defined targets. *Journal of Experimental Psychology: Human Perception and Performance*, 10(1):32–39, 1984.

[38] Nils Einecke, Manuel Mühlig, Jens Schmüdderich, and Michael Gienger. "Bring it to me" - generation of behavior-relevant scene elements for interactive robot scenarios. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3415–3422. IEEE, 2011.

[39] Lior Elazary and Laurent Itti. A bayesian model for efficient visual search and recognition. *Vision Research*, 50(14):1338–1352, Jun 2010.

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

[41] Wolfram Erlhagen and Gregor Schöner. Dynamic field theory of movement preparation. *Psychological Review*, 109(3):545–572, 2002.

[42] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In Josef Bigun and Tomas Gustavsson, editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 363–370. Springer Berlin Heidelberg, 2003.

[43] Christian Faubel and Gregor Schöner. A neuro-dynamic architecture for one shot learning of objects that uses both bottom-up recognition and top-down prediction. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3162–3169. IEEE, 2009.

[44] Christian Faubel and Stephan K. U. Zibner. A neuro-dynamic object recognition architecture enhanced by foveal vision and a gaze control mechanism. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1171–1176. IEEE, 2010.

[45] Anatol G. Feldman. Functional tuning of nervous system with control of movement or maintenance of a steady posture. II. Controllable parameters of muscles. *Biophysics*, 11(3):565–578, 1966.

[46] Anatol G. Feldman. Once more on the equilibrium-point hypothesis (λ model) for motor control. *Journal of Motor Behavior*, 18(1):17–54, 1986.

[47] Anatol G. Feldman. Space and time in the context of equilibrium-point theory. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(3):287–304, 2011.

[48] Jerome A. Feldman. Four frames suffice: A provisional model of vision and space. *Behavioral and Brain Sciences*, 8:265–289, 1985.

[49] Jérémy Fix, Nicolas Rougier, and Frédéric Alexandre. A dynamic neural field approach to the covert and overt deployment of spatial attention. *Cognitive Computation*, 3(1):279–293, 2011.

[50] Jérémy Fix, Julien Vitay, and Nicolas P. Rougier. A distributed computational model of spatial memory anticipation during a visual search task. In Martin V. Butz, Olivier Sigaud, Giovanni Pezzulo, and Gianluca Baldassarre, editors, *Anticipatory Behavior in Adaptive Learning Systems*, volume 4520 of *Lecture Notes in Computer Science*, pages 170–188. Springer Berlin Heidelberg, 2007.

[51] J. Randall Flanagan, David J. Ostry, and Anatol G. Feldman. Control of trajectory modifications in target-directed reaching. *Journal of Motor Behavior*, 25(3):140–152, 1993.

[52] Tamar Flash and Ealan Henis. Arm trajectory modifications during reaching towards visual targets. *Journal of Cognitive Neuroscience*, 3(3):220–230, 1991.

[53] Mathias Franzius, Niko Wilbert, and Laurenz Wiskott. Invariant object recognition with slow feature analysis. In Véra Kůrková, Roman Neruda, and Jan Koutník, editors, *Artificial Neural Networks — ICANN 2008*, volume 5163 of *Lecture Notes in Computer Science*, pages 961–970. Springer Berlin Heidelberg, 2008.

[54] Apostolos P. Georgopoulos, John F. Kalaska, and Joe T. Massey. Spatial trajectories and reaction times of aimed movements: effects of practice, uncertainty, and change in target location. *Journal of Neurophysiology*, 46(4):725–743, 1981.

[55] Joshua Goldberg. *When, Not Where a Dynamical Field Theory of Infant Gaze*. PhD thesis, Indiana University, 2009.

[56] Hiroaki Gomi and Mitsuo Kawato. Equilibrium-point control hypothesis examined by measured arm stiffness during multijoint movement. *Science*, 272(5258):117–120, 1996.

[57] Nikos Gorgoraptis, Raquel F. G. Catalao, Paul M. Bays, and Masud Husain. Dynamic updating of working memory resources for visual objects. *The Journal of Neuroscience*, 31(23):8502–8511, 2011.

[58] Gerald L. Gottlieb, Qilai Song, Da Hong, and Daniel M. Corcos. Coordinating two degrees of freedom during human arm movement: load and speed invariance of relative joint torques. *Journal of Neurophysiology*, 76(5):3196–3206, 1996.

[59] Paul L. Gribble, David J. Ostry, Vittorio Sanguineti, and Rafael Laboissière. Are complex control signals required for human arm movement? *Journal of Neurophysiology*, 79(3):1409–1424, 1998.

[60] Britta Grimme, John Lipinski, and Gregor Schöner. Naturalistic arm movements during obstacle avoidance in 3D and the identification of movement primitives. *Experimental Brain Research*, 222(3):185–200, 2012.

[61] Pascal Haazebroek, Saskia van Dantzig, and Bernhard Hommel. A computational model of perception and action for cognitive robotics. *Cognitive Processing*, 12(4):355–365, 2011.

[62] Fred H. Hamker. A dynamic model of how feature cues guide spatial attention. *Vision Research*, 44(5):501–521, 2004.

[63] Fred H. Hamker. A computational model of visual stability and change detection during eye movements in real-world scenes. *Visual Cognition*, 12(6):1161–1176, 2005.

[64] Fred H. Hamker. Modeling feature-based attention as an active top-down inference process. *BioSystems*, 86(1–3):91–99, 2006.

[65] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with Microsoft Kinect sensor: A review. *Cybernetics, IEEE Transactions on*, 43(5):1318–1334, 2013.

[66] Eva Hansen, Britta Grimme, Hendrik Reimann, and Gregor Schöner. Carry-over coarticulation in joint angles. *Experimental Brain Research*, 233(9):2555–2569, 2015.

[67] Bridgette M. Hard, Gabriel Recchia, and Barbara Tversky. The shape of action. *Journal of Experimental Psychology: General*, 140(4):586–604, 2011.

[68] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1–3):335–346, 1990.

[69] John M. Henderson. Transsaccadic memory and integration during real-world object perception. *Psychological Science*, 8(1):51–55, 1997.

[70] John M. Henderson. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences*, 7(11):498–504, 2003.

[71] John M. Henderson and Andrew Hollingworth. Eye movements during scene viewing: An overview. In Geoffrey Underwood, editor, *Eye Guidance in Reading and Scene Perception*, chapter 12, pages 269–293. Elsevier Science, 1998.

[72] John M. Henderson and Andrew Hollingworth. High-level scene perception. *Annual Review of Psychology*, 50(1):243–271, 1999.

[73] Howard S. Hock, Gregor Schöner, and Lee Gilroy. A counterchange mechanism for the perception of motion. *Acta Psychologica*, 132(1):1–21, 2009.

[74] James E. Hoffman. A two-stage model of visual search. *Perception & Psychophysics*, 25(4):319–327, 1979.

[75] Andrew Hollingworth. Failures of retrieval and comparison constrain change detection in natural scenes. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):388–403, 2003.

[76] Andrew Hollingworth. Constructing visual representations of natural scenes: the roles of short-and long-term visual memory. *Journal of Experimental Psychology: Human Perception and Performance*, 30(3):519–537, 2004.

[77] Andrew Hollingworth and John M. Henderson. Accurate visual memory for previously attended objects in natural scenes. *Journal of Experimental Psychology: Human Perception and Performance*, 28(1):113–136, 2002.

[78] Andrew Hollingworth, Carrick C. Williams, and John M. Henderson. To see and remember: Visually specific information is retained in memory from previously attended objects in natural scenes. *Psychonomic Bulletin & Review*, 8(4):761–768, 2001.

[79] Todd S. Horowitz and Jeremy M. Wolfe. Visual search has no memory. *Nature*, 394(6693):575–577, 1998.

[80] George Houghton and Steven P. Tipper. A model of inhibitory mechanisms in selective attention. In Dale Dagenbach and Thomas H. Carr, editors, *Inhibitory processes in attention, memory, and language*, chapter 2, pages 53–112. Academic Press, 1994.

[81] Amelia R. Hunt and Alan Kingstone. Covert and overt voluntary attention: linked or independent? *Cognitive Brain Research*, 18(1):102–105, 2003.

[82] Joo-seok Hyun, Geoffrey F. Woodman, Edward K. Vogel, Andrew Hollingworth, and Steven J. Luck. The comparison of visual working memory representations with perceptual inputs. *Journal of Experimental Psychology: Human Perception and Performance*, 35(4):1140–1160, 2009.

[83] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[84] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 1998.

[85] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[86] Jeffrey S. Johnson, John P. Spencer, Steven J. Luck, and Gregor Schöner. A dynamic neural field model of visual working memory and change detection. *Psychological Science*, 20(5):568–577, 2009.

[87] Jeffrey S. Johnson, John P. Spencer, and Gregor Schöner. Moving to higher ground: The dynamic field theory and the dynamics of visual cognition. *New Ideas in Psychology*, 26(2):227–251, 2008.

[88] Michael J. Kane, Bradley J. Poole, Stephen W. Tuholski, and Randall W. Engle. Working memory capacity and the top-down control of visual search: Exploring the boundaries of "executive attention". *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(4):749–777, 2006.

[89] Raymond M. Klein. Inhibitory tagging system facilitates visual search. *Nature*, 334(6181):430–431, 1988.

[90] Raymond M. Klein. Inhibition of return. *Trends in Cognitive Sciences*, 4(4):138–147, 2000.

[91] Guido Knips. Schätzung und Repräsentation von Greifparametern auf Basis neuronaler Felder zur Generierung von robotischen Greifbewegungen in visuellen Szenen. Master's thesis, Ruhr-Universität Bochum, 2013.

[92] Guido Knips, Stephan K. U. Zibner, Hendrik Reimann, Irina Popova, and Gregor Schöner. A neural dynamics architecture for grasping that integrates perception and movement generation and enables on-line updating. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 646–653, 2014.

[93] Guido Knips, Stephan K. U. Zibner, Hendrik Reimann, Irina Popova, and Gregor Schöner. Reaching and grasping novel objects: Using neural dynamics to integrate and organize scene and object perception with movement generation. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pages 311–318, 2014.

[94] Christof Koch and Shimon Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. In Lucia M. Vaina, editor, *Matters of Intelligence*, volume 188 of *Synthese Library*, pages 115–141. Springer Netherlands, 1987.

[95] Kristin Koch, Judith McLean, Ronen Segev, Michael A. Freed, Michael J. Berry II, Vijay Balasubramanian, and Peter Sterling. How much the eye tells the brain. *Current Biology*, 16(14):1428–1434, 2006.

[96] George Konidaris, Leslie P. Kaelbling, and Tomas Lozano-Perez. Symbol acquisition for task-level planning. In *The AAAI Workshop on Learning Rich Representations from Low-Level Sensors*. American Association for the Advancement of Science (AAAS), 2013.

[97] Klaus Kopecz, Christoph Engels, and Gregor Schöner. Dynamic field approach to target selection in gaze control. In *ICANN'93*, pages 96–101. Springer, 1993.

[98] Klaus Kopecz and Gregor Schöner. Saccadic motor planning by integrating visual information and pre-information on neural dynamic fields. *Biological Cybernetics*, 73(1):49–60, 1995.

[99] Benjamin Kühn, Boris Schauerte, Rainer Stiefelhagen, and Kristian Kroschel. A modular audio-visual scene analysis and attention system for humanoid robots. In *The 43rd International Symposium on Robotics (ISR)*, 2012.

[100] Benjamin Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1):191–233, 2000.

[101] Michael F. Land and Mary Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41(25–26):3559–3565, 2001.

[102] Mark L. Latash and Gerald L. Gottlieb. Reconstruction of shifting elbow joint compliant characteristics during fast and slow movements. *Neuroscience*, 43(2–3):697–712, 1991.

[103] Oliver Lomp, Stephan K. U. Zibner, Mathis Richter, Iñaki Rañó, and Gregor Schöner. A software framework for cognition, embodiment, dynamics, and autonomy in robotics: cedar. In Valeri Mladenov, Petia Koprinkova-Hristova, Günther Palm, Alessandro E. P. Villa, Bruno Appollini, and Nikola Kasabov, editors, *Artificial Neural Networks and Machine Learning — ICANN 2013*, volume 8131 of *Lecture Notes in Computer Science*, pages 475–482. Springer Berlin Heidelberg, 2013.

[104] David G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157. IEEE, 1999.

[105] Matthew Luciw, Konstantin Lakhmann, Sohrob Kazerounian, Mathis Richter, and Yulia Sandamirskaya. Learning the perceptual conditions of satisfaction of elementary behaviors. In *Robotics: Science and Systems (RSS), Workshop Active Learning in Robotics: Exploration, Curiosity, and Interaction*, 2013.

[106] Steven J. Luck and Edward K. Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390(6657):279–281, 1997.

[107] Dwarikanath Mahapatra, Stefan Winkler, and Shih-Cheng Yen. Motion saliency outweighs other low-level features while watching videos. In *Proc. SPIE 6806, Human Vision and Electronic Imaging XIII*, pages 68060P–68060P–10, 2008.

[108] Riccardo Manzotti and Vincenzo Tagliasco. From behaviour-based robots to motivation-based robots. *Robotics and Autonomous Systems*, 51(2–3):175–190, 2005.

[109] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.

[110] Elizabeth A. Maylor. Facilitatory and inhibitory components of orienting in visual space. In Michael I. Posner and Oscar Marin, editors, *Attention and Performance: 11th International symposium*, pages 189–204. L. Erlbaum Associates, 1985.

[111] Inna Mikhailova and Christian Goerick. Conditions of activity bubble uniqueness in dynamic neural fields. *Biological Cybernetics*, 92(2):82–91, 2005.

[112] Mortimer Mishkin, Leslie G. Ungerleider, and Kathleen A. Macko. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences*, 6:414–417, 1983.

[113] Cathleen M. Moore and Howard Egeth. How does feature-based attention affect visual processing? *Journal of Experimental Psychology: Human Perception and Performance*, 24(4):1296–1310, 1998.

[114] Daniel W. Moran and Andrew B. Schwartz. Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology*, 82(5):2676–2692, 1999.

[115] Pietro Morasso. Spatial control of arm movements. *Experimental Brain Research*, 42(2):223–227, 1981.

[116] Vidhya Navalpakkam and Laurent Itti. Modeling the influence of task on attention. *Vision Research*, 45(2):205–231, 2005.

[117] Ulric Neisser. *Cognitive Psychology: Classic Edition*. Psychology Press, 2014.

[118] Sammeer A. Nene, Shree Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-100). *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*, 1996.

[119] Hans-Christoph Nothdurft. Saliency effects across dimensions in visual search. *Vision Research*, 33(5–6):839–844, 1993.

[120] Hans-Christoph Nothdurft, Ivan N. Pigarev, and Sabine Kastner. Overt and covert visual search in primates: reaction times and gaze shift strategies. *Journal of Integrative Neuroscience*, 08(02):137–174, 2009.

[121] Andreas Nüchter and Joachim Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[122] Dimitri Ognibene, Christian Balkenius, and Gianluca Baldassarre. Integrating epistemic action (active vision) and pragmatic action (reaching): a neural architecture for camera-arm robots. In Minoru Asada, John C. T. Hallam, Jean-Arcady Meyer, and Jun Tani, editors, *From Animals to Animats 10*, volume 5040 of *Lecture Notes in Computer Science*, pages 220–229. Springer Berlin Heidelberg, 2008.

[123] Aude Oliva, Antonio Torralba, Monica S. Castelhano, and John M. Henderson. Top-down control of visual attention in object detection. In *Image Processing. ICIP 2003. Proceedings of the International Conference on*, volume 1, pages 253–256. IEEE, 2003.

[124] Christian N. L. Olivers, Judith Peters, Roos Houtkamp, and Pieter R. Roelfsema. Different states in visual working memory: When it guides attention and when it does not. *Trends in Cognitive Sciences*, 15(7):327–334, 2011.

[125] Farid Oubbati, Mathis Richter, and Gregor Schöner. A neural dynamics to organize timed movement: Demonstration in a robot ball bouncing task. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, pages 379–386. IEEE, 2014.

[126] Matthew S. Peterson, Arthur F. Kramer, Ranxiao Frances Wang, David E. Irwin, and Jason S. McCarley. Visual search has memory. *Psychological Science*, 12(4):287–292, 2001.

[127] W. A. (Bill) Phillips. On the distinction between sensory storage and short-term visual memory. *Perception & Psychophysics*, 16(2):283–290, 1974.

[128] Michael I. Posner and Yoav Cohen. Components of visual orienting. In Herman Bouma and Don G. Bouwhuis, editors, *Attention and performance: Control of language processes*, volume 10, chapter 32, pages 531–556. Psychology Press, 1984.

[129] Claude Prablanc and Olivier Martin. Automatic control during hand reaching at undetected two-dimensional target displacements. *Journal of Neurophysiology*, 67(2):455–469, 1992.

[130] Andrzej Pronobis, Patric Jensfelt, Kristoffer Sjöö, Hendrik Zender, Geert-Jan M. Kruijff, Oscar Martinez Mozos, and Wolfram Burgard. Semantic modelling of space. In *Cognitive Systems*, pages 165–221. Springer, 2010.

[131] Zenon W. Pylyshyn. Some puzzling findings in multiple object tracking: I. Tracking without keeping track of object identities. *Visual Cognition*, 11(7):801–822, 2004.

[132] Zenon W. Pylyshyn and Ron W. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3(3):179–197, 1988.

[133] Rajesh P. N. Rao, Gregory J. Zelinsky, Mary M. Hayhoe, and Dana H. Ballard. Eye movements in iconic visual search. *Vision Research*, 42(11):1447–1463, 2002.

[134] Babak Rasolzadeh, Mårten Björkman, Kai Huebner, and Danica Kragic. An active vision system for detecting, fixating and manipulating objects in the real world. *The International Journal of Robotics Research*, 29(2–3):133–154, 2010.

[135] Ronald A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1–3):17–42, 2000.

[136] Ronald A. Rensink. Change detection. *Annual Review of Psychology*, 53(1):245–277, 2002.

[137] Ronald A. Rensink, J. Kevin O'Regan, and James J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997.

[138] Alison Rich and Barbara Gillam. Failure to detect changes in color for lines rotating in depth: the effects of grouping and type of color change. *Vision Research*, 40(10–12):1377–1384, 2000.

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

[140] Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[141] Giacomo Rizzolatti, Lucia Riggio, Isabella Dascola, and Carlo Umiltá. Reorienting attention across the horizontal and vertical meridians: evidence in favor of a premotor theory of attention. *Neuropsychologia*, 25(1):31–40, 1987.

[142] Uri Rokni and Haim Sompolinsky. How the brain generates movement. *Neural Computation*, 24(2):289–331, 2012.

[143] Ruth Rosenholtz. Search asymmetries? What search asymmetries? *Perception & Psychophysics*, 63(3):476–489, 2001.

[144] John Ross, M. Concetta Morrone, Michael E. Goldberg, and David C. Burr. Changes in visual perception at the time of saccades. *Trends in Neurosciences*, 24(2):113–121, 2001.

[145] Gennaro Ruggiero, Francesco Ruotolo, and Tina Iachini. The role of vision in egocentric and allocentric spatial frames of reference. *Cognitive Processing*, 10(2 Supplement):283–285, 2009.

[146] Radu B. Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.

[147] Ueli Rutishauser, Dirk Walther, Christof Koch, and Pietro Perona. Is bottom-up attention useful for object recognition? In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–37–II–44, 2004.

[148] Dov Sagi and Bela Julesz. Enhanced detection in the aperture of focal attention during simple discrimination tasks. *Nature*, 321(6071):693–695, 1986.

[149] Yulia Sandamirskaya, John Lipinski, Ioannis Iossifidis, and Gregor Schöner. Natural human-robot interaction through spatial language: a dynamic neural field approach. In *RO-MAN, 2010 IEEE*, pages 600–607. IEEE, 2010.

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

[151] Yulia Sandamirskaya and Tobias Storck. Learning to Look and Looking to Remember: A Neural-Dynamic Embodied Model for Generation of Saccadic Gaze Shifts and Memory Formation. In Petia Koprinkova-Hristova, Valeri Mladenov, and Nikola K. Kasabov, editors, *Artificial Neural Networks*, volume 4 of *Springer Series in Bio-/Neuroinformatics*, pages 175–200. Springer International Publishing, 2015.

[152] 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.

[153] Andries F. Sanders and Mechtilda J. M. Houtmans. Perceptual processing modes in the functional visual field. *Acta Psychologica*, 58(3):251–261, 1985.

[154] Boris Schauerte and Gernot A. Fink. Focusing computational visual attention in multi-modal human-robot interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, pages 6:1–6:8. ACM, 2010.

[155] Werner X. Schneider. Visual-spatial working memory, attention, and scene representation: A neuro-cognitive theory. *Psychological Research*, 62(2):220–236, 1999.

[156] John P. Scholz and Gregor Schöner. The uncontrolled manifold concept: identifying control variables for a functional task. *Experimental Brain Research*, 126(3):289–306, 1999.

[157] Gregor Schöner. Dynamical systems approaches to cognition. In Ron Sun, editor, *Cambridge Handbook of Computational Cognitive Psychology*, chapter 4, pages 101–126. Cambridge University Press, 2008.

[158] Gregor Schöner, Michael Dose, and Christoph Engels. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems*, 16(2–4):213–245, 1995.

[159] Anne R. Schutte, John P. Spencer, and Gregor Schöner. Testing the dynamic field theory: Working memory for locations becomes more spatially precise over development. *Child Development*, 74(5):1393–1417, 2003.

[160] John R. Searle. *Intentionality: An essay in the philosophy of mind*. Cambridge University Press, 1983.

[161] John R. Searle. *Mind: a brief introduction*. Oxford University Press, 2004.

[162] Mariana M. Silva, John A. Groeger, and Mark F. Bradshaw. Attention-memory interactions in scene perception. *Spatial Vision*, 19(1):9–19, 2006.

[163] Daniel J. Simons and Daniel T. Levin. Change blindness. *Trends in Cognitive Sciences*, 1(7):261–267, 1997.

[164] Daniel T. Smith and Thomas Schenk. The premotor theory of attention: Time to move on? *Neuropsychologia*, 50(6):1104–1114, 2012.

[165] Richard B. Stein. What muscle variable(s) does the nervous system control in limb movements? *Behavioral and Brain Sciences*, 5(04):535–541, 1982.

[166] Axel Steinhage and Thomas Bergener. Dynamical systems for the behavioral organization of an anthropomorphic mobile robot. In *From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior*, pages 147–152. MIT Press, 1998.

[167] Soeren Strauss and Dietmar Heinke. A robotics-based approach to modeling of choice reaching experiments on visual attention. *Frontiers in Psychology*, 3:105, 2012.

[168] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[169] Duje Tadin, Joseph S. Lappin, Randolph Blake, and Emily D. Grossman. What constitutes an efficient reference frame for vision? *Nature Neuroscience*, 5(10):1010–1015, 2002.

[170] John G. Taylor. Neural 'bubble' dynamics in two dimensions: foundations. *Biological Cybernetics*, 80(6):393–409, 1999.

[171] Jan Tekülve. Eine neuronal dynamische Architektur zur Erzeugung zielorientierter robotischer Armbewegungen. Master's thesis, Ruhr-Universität Bochum, 2014.

[172] Kasim Terzić, David Lobato, Mário Saleiro, and J. M. H. (Hans) du Buf. A fast neural-dynamical approach to scale-invariant object detection. In Chu K. Loo, Keem S. Yap, Kok W. Wong, Andrew Teoh, and Kaizhu Huang, editors, *Neural Information Processing*, volume 8834 of *Lecture Notes in Computer Science*, pages 511–518. Springer International Publishing, 2014.

[173] Esther Thelen, Daniela Corbetta, and John P. Spencer. Development of reaching during the first year: The role of movement speed. *Journal of Experimental Psychology: Human Perception and Performance*, 22(5):1059–1076, 1996.

[174] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 321–328. IEEE, 2000.

[175] Emanuel Todorov. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nature Neuroscience*, 3(4):391–398, 2000.

[176] Emanuel Todorov and Michael I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002.

[177] Godfried T. Toussaint. Solving geometric problems with the rotating calipers. In *Proceedings of IEEE MELECON*, 1983.

[178] Anne M. Treisman. The perception of features and objects. In Alan D. Baddeley and Lawrence Weiskrantz, editors, *Attention: Selection, awareness, and control: A tribute to Donald Broadbent*, pages 5–35. Clarendon Press/Oxford University Press, 1993.

[179] Anne M. Treisman. The binding problem. *Current Opinion in Neurobiology*, 6(2):171–178, 1996.

[180] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[181] Leslie G. Ungerleider and James V. Haxby. 'What' and 'where' in the human brain. *Current Opinion in Neurobiology*, 4(2):157–165, 1994.

[182] J. F. Van Sonderen, J. J. Denier Van der Gon, and C. C. A. M. (Stan) Gielen. Conditions determining early modification of motor programmes in response to changes in target location. *Experimental Brain Research*, 71(2):320–328, 1988.

[183] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518. IEEE, 2001.

[184] Edward K. Vogel, Geoffrey F. Woodman, and Steven J. Luck. Storage of features, conjunctions, and objects in visual working memory. *Journal of Experimental Psychology: Human Perception and Performance*, 27(1):92–114, 2001.

[185] Claes von Hofsten. Developmental changes in the organization of prereaching movements. *Developmental Psychology*, 20(3):378–388, 1984.

[186] Derrick G. Watson and Glyn W. Humphreys. Visual marking: Evidence for inhibition using a probe-dot detection paradigm. *Perception & Psychophysics*, 62(3):471–481, 2000.

[187] Mary E. Wheeler and Anne M. Treisman. Binding in short-term visual memory. *Journal of Experimental Psychology: General*, 131(1):48–64, 2002.

[188] Jeremy M. Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238, 1994.

[189] Jeremy M. Wolfe. Visual search. In Harold Pashler, editor, *Attention*, chapter 1, pages 13–73. Psychology Press, 1998.

[190] Jeremy M. Wolfe. Inattentional amnesia. In Veronika Coltheart, editor, *Fleeting memories: Cognition of brief visual stimuli*, pages 71–94. MIT Press, 1999.

[191] Jeremy M. Wolfe. Guided search 4.0: Current progress with a model of visual search. In Wayne D Gray, editor, *Integrated models of cognitive systems*, chapter 8, pages 99–119. Oxford University Press New York, 2007.

[192] Geoffrey F. Woodman and Steven J. Luck. Do the contents of visual working memory automatically influence attentional selection during visual search? *Journal of Experimental Psychology: Human Perception and Performance*, 33(2):363–377, 2007.

[193] Shu-Chieh Wu and Roger W. Remington. Characteristics of covert and overt visual orienting: Evidence from attentional and oculomotor capture. *Journal of Experimental Psychology: Human Perception and Performance*, 29(5):1050–1067, 2003.

[194] Stephan K. U. Zibner. Three dimensional space-feature fields and their application in scene representation. Master's thesis, Ruhr-Universität Bochum, 2009.

[195] Stephan K. U. Zibner and Christian Faubel. Dynamic scene representations and autonomous robotics. In *Dynamic Thinking. A Primer on Dynamic Field Theory*, chapter 9, pages 227–245. Oxford University Press, 2015 (in press).

[196] Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, and Gregor Schöner. Dynamic neural fields as building blocks of a cortex-inspired architecture for robotic scene representation. *Autonomous Mental Development, IEEE Transactions on*, 3(1):74–91, 2011.

[197] Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, Gregor Schöner, and John P. Spencer. Scenes and tracking with dynamic neural fields: How to update a robotic scene representation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pages 244–250. IEEE, 2010.

[198] Stephan K. U. Zibner, Christian Faubel, and Gregor Schöner. Making a robotic scene representation accessible to feature and label queries. In *Development and Learning and on Epigenetic Robotics (ICDL-Epirob), 2011 First Joint IEEE International Conferences on*, 2011.

[199] Stephan K. U. Zibner, Jan Tekülve, and Gregor Schöner. The Neural Dynamics of Goal-Directed Arm Movements: A Developmental Perspective. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, pages 154–161, 2015.

[200] Stephan K. U. Zibner, Jan Tekülve, and Gregor Schöner. The Sequential Organization of Movement is Critical to the Development of Reaching: A Neural Dynamics Account. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, pages 39–46, 2015.

Appendix A

Additional Material, Data Set, and Software

In this appendix, I present additional material for the reader as well as the data set and software that were used to create the results presented in this thesis.

Videos

The time course of the behaviors of my scene representation architecture is captured in the following video demonstrations.

Visual Exploration

A video demonstrating visual exploration is available at `http://bit.ly/1PSp84f` (exploration.mpeg, 10.9 MB). A screenshot is shown in Figure A.1. The top row of the video shows, from left to right, the camera input, the output of the saliency processing and the activation of the *attention field*. The bottom row shows, from left to right, a visual reconstruction displaying the content of space-color working memory, activation of the *space-color working memory*, and the sigmoided activation of several nodes (intention nodes relating to the scene and object levels, condition of satisfaction node of *exploration*, and *match* and *no match nodes* of the three feature channels). Active intention and CoS nodes are blue, match nodes are red, and no match nodes are green.

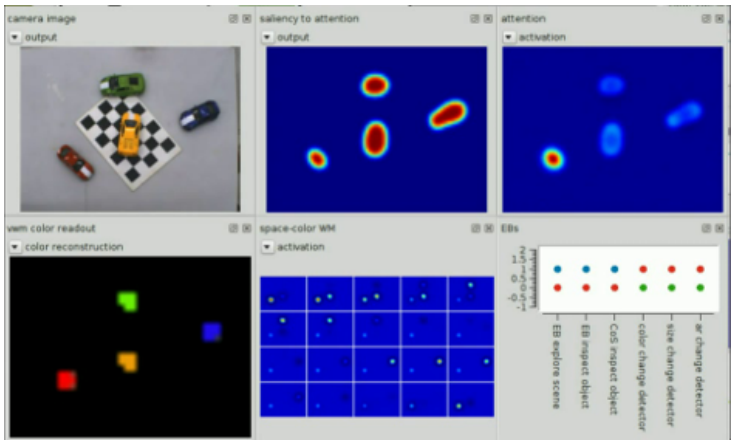

Figure A.1: A sample screenshot of the video demonstrating the exploration behavior.

Maintenace

Maintenance of the working memory representation is demonstrated in a video available at `http://bit.ly/1QPfWxz` (change_detection.mpeg, 26.4 MB). A screenshot of the video is shown in Figure A.2. The video shows the same fields and nodes as described above. In this demonstration, only the color channel is active. Over time, I exchange cups in the scene with cups of a different color. The representation is correctly updated, both when the exchanged cup is currently in the attentional focus, as well as when the exchange happens while attention is focused on another object.

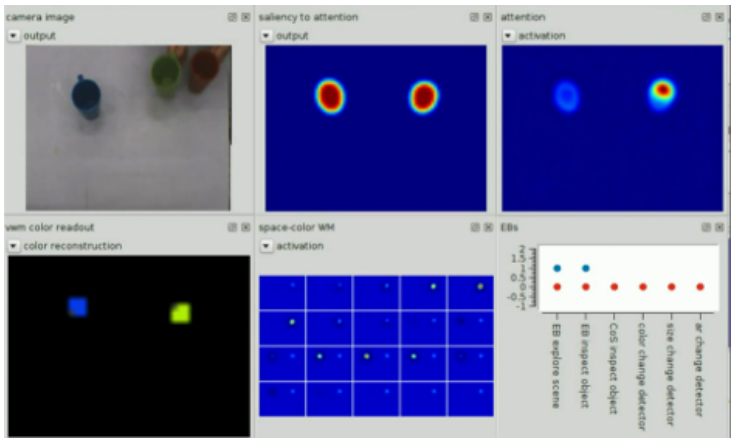

Figure A.2: A sample screenshot of the video demonstrating the maintenance behavior.

Query

The query behavior is demonstrated in a video at `http://bit.ly/1Z6oTst` (query.mpeg, 10.1 MB). A screenshot is shown in Figure A.3. The top row of the videos shows, from left to right, the output of the saliency processing, the activation and the sigmoided activation of the *attention field*. The bottom row shows, from left to right, the sigmoided activation of dynamic nodes involved in the query behavior, the camera input, and a user interface, which allows to activate and deactivate cues.

In this video, I activate several cues in sequence. The first cue is *blue*, which matches none of the objects in the scene. Consequently, the CoD node activates for every attended object and search continues. A cue for *yellow* selects the yellow cookie box and activates the CoS node. Switching to a combined cue of *red* and *round* (with *round* signifying an aspect ratio of one) results in a selection of the red cup.

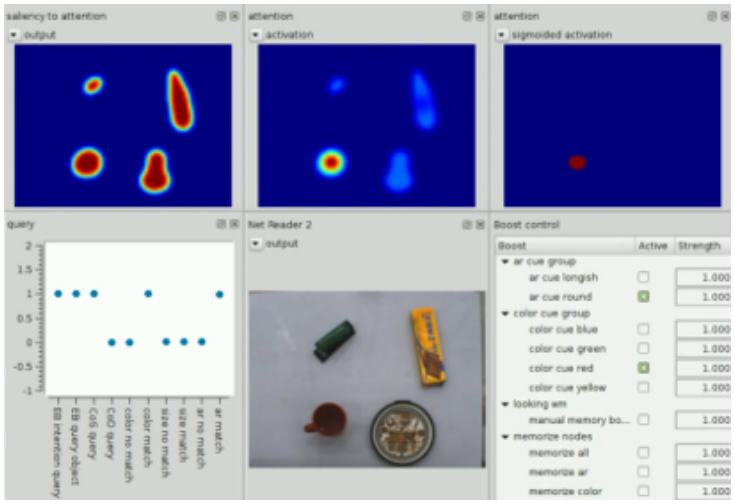

Figure A.3: A sample screenshot of the video demonstrating the query behavior.

Data Set

The data set of scenes used in the experiments can be downloaded using the following links. All scenes are video files encoded in MPEG format.

- static scenes: `http://bit.ly/1iy9os1` (scenes.zip, 255 MB)
- dynamic scenes: `http://bit.ly/1Fw38eQ` (change.zip, 71 MB)

Software

The presented scene representation and movement generations architectures are implemented in *cedar* [103], a software library designed as a toolbox for DFT modeling. With *cedar*, the methods and architectures of this thesis can be analyzed and reproduced by other researchers. The following sections point to sites hosting the essential components. I assume basic knowledge of the software framework to use these components. A starting point for new users is the *cedar* website `http://cedar.ini.rub.de/`, which contains instructions on how to install *cedar* and tutorials to get started.

Methods

I provide several examples of the introduced methods for *cedar* covering projections, nodes, biased selection, and match detection. The examples can be accessed at `http://bit.ly/1Lag1hd` (examples.zip, 0.1 MB).

- projections.json: exemplary expansion and contraction projections
- nodes.json: examples of the four types of nodes
- biased_selection.json: different degrees of bias during selection decisions
- match_detector.json: a basic match detector

Plugins for Scene Representation

Two *cedar* plugins are necessary to load and simulate the scene representation architecture. They can be downloaded at the following sites.

- `https://bitbucket.org/StephanZibner/sceneplugin`
- `https://bitbucket.org/StephanZibner/stereoimagingplugin`

Scene Representation Architecture

The full architecture can be downloaded at the following site: `http://bit.ly/1W9248i` (architecture.zip, 0.04 MB). The zip archive contains a readme file that explains how to set up a camera input, start the simulation, and activate different behaviors.

Movement Generation Architecture

Instructions on how to access the movement generation architecture can be found at the following site: `http://www.neuraldynamics.eu/index.php?page=architectures`.

Appendix B

Notation and Lists of used Field, Node, and Input Variables

The large amount of differential equations and their parameters requires suitable notation guidelines to keep them readable. In Chapter 3 I use generic equations as templates for architecture components. By contrast, Chapter 4 contains concrete, named building blocks. Table B.1 gives an overview of the general usage of variables in this thesis, both for generic equations and architecture descriptions. Tables B.2, B.3, and B.4 contain definitions of DFs, dynamic nodes, and inputs used in the scene representation architecture. I do not include a complete list of parameter values here on purpose, as the sheer amount of parameters would require disproportionate amounts of additional pages. All parameters can be looked up in *cedar* configuration files, as explained in Appendix A.

All equations in this thesis follow these guidelines:

- each field and node has a custom time constant and resting level, which are not marked with additional indices to save space
- resting levels are always negative and the detection threshold of any sigmoid function is at zero
- convolutions of weight matrices w and DF output $\sigma(u)$ are represented in brackets using the symbol $*$; the relevant dimensions extracted from the convolution result are added in parentheses after the brackets (for example $[w * \sigma(u)](x, y, t)$)
- DFs and dynamic nodes have three-letter subscript indices that are

abbreviations of their functional role (for example `atn` as abbreviation of an attention field)

- if DFs or dynamic nodes exists once for every feature channel, a superscript F is added to the activation variable; if other superscript indices are attached to the activation variable, they are concatenated using a comma as separator (e.g., $u_{\text{cue}}^{\text{pd,F}}$)
- dynamic nodes may have an additional superscript index that signifies their role (for example `pd` for a peak detector)
- weight matrices w and scalar weights c have concatenated indices of the form *target*,*source*, with *target* and *source* being the indices of target and source, respectively (e.g., $w_{\text{atn,sal}}$ for weights from a saliency field to an attention field)
- weight matrices and scalar weights implementing lateral interactions have the same index as the activation variable (e.g., w_{atn} for u_{atn})
- projections that implement inhibitory influences are made explicit by using a negative sign for inhibitory summands, with weight matrices and scalar weights being positive

Table B.1: Variables used throughout the thesis.

Abbrev.	**Explanation**
u	field or node activation
p	memory trace
τ	time constant
h	resting level of field or node
s	input to DFs or dynamic nodes
σ, θ	transfer functions
β	steepness parameter of sigmoid function
x, y	spatial dimensions
f	feature dimension
r	feature dimension *color*
k	feature dimension *size*
a	feature dimension *aspect-ratio*
t	time
i, j, k, m, n	indices
w	kernels, weight matrices
c	scalar weights
rb	superscript index for nodes boosting the resting level of DFs
pd	superscript index for peak detector nodes
ci	superscript index for nodes inducing categorical peaks in DFs
cd	superscript index for nodes detecting categorical peaks in DFs

Table B.2: Input definitions.

Abbrev.	**Explanation**
$s^{\text{col}}_{\text{bup}}$	bottom up input into early space-color field
$s^{\text{siz}}_{\text{bup}}$	bottom up input into early space-size field
$s^{\text{rat}}_{\text{bup}}$	bottom up input into early space-aspect-ratio field
$s^{\text{F}}_{\text{cue}}$	external cue input for F $\in$ {color, size, aspect ratio}

Table B.3: Field definitions.

Abbrev.	**Explanation**
$u^{\text{F}}_{\text{esf}}$	early space-feature fields for F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{cue}}$	feature cue fields for F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{con}}$	conspicuity fields for F $\in$ {color, size, aspect ratio}
u_{sal}	saliency field
$u^{\text{F}}_{\text{fex}}$	fields representing extracted feature for F $\in$ {color, size, aspect ratio}
u_{atn}	attention field
$u^{\text{F}}_{\text{sfq}}$	space-feature query fields for F $\in$ {color, size, aspect ratio}
u_{lwm}	looking working memory field
p_{atn}	memory trace of attention field
$u^{\text{F}}_{\text{sfm}}$	space-feature working memory fields for F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{fme}}$	feature memorization fields for F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{fqu}}$	feature query for F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{exp}}$	feature expectation field for feature F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{nom}}$	no match field for feature F $\in$ {color, size, aspect ratio}
$u^{\text{F}}_{\text{cin}}$	conditional spatial inhibition for feature F $\in$ {color, size, aspect ratio}
u_{meb}	memory bias field

Table B.4: Node definitions.

Abbrev.	**Explanation**
$u_{\text{int}}^{\text{es}}$	intention node for behavior *explore scene*
$u_{\text{int}}^{\text{io}}$	intention node for behavior *inspect object*
$u_{\text{cos}}^{\text{io}}$	condition of satisfaction node for behavior *inspect object*
$u_{\text{int}}^{\text{qs}}$	intention node for behavior *query scene*
$u_{\text{cos}}^{\text{qs}}$	condition of satisfaction node for behavior *query scene*
$u_{\text{int}}^{\text{qo}}$	intention node for behavior *query object*
$u_{\text{cos}}^{\text{qo}}$	condition of satisfaction node for behavior *query object*
$u_{\text{cod}}^{\text{qo}}$	condition of dissatisfaction node for behavior *query object*
$u_{\text{mat}}^{\text{F}}$	match node for feature F $\in$ {color, size, aspect ratio}
$u_{\text{nom}}^{\text{F}}$	no match node for feature F $\in$ {color, size, aspect ratio}
$u_{\text{hme}}^{\text{F}}$	nodes indicating the presence of space-feature memory for F $\in$ {color, size, aspect ratio}
$u_{\text{mef}}^{\text{F}}$	nodes indicating the task to memorize features for F $\in$ {color, size, aspect ratio}
$u_{\text{fex}}^{\text{pd,F}}$	peak detector for feature extraction fields for F $\in$ {color, size, aspect ratio}
$u_{\text{exp}}^{\text{pd,F}}$	peak detector for feature expectation fields for F $\in$ {color, size, aspect ratio}
$u_{\text{nom}}^{\text{pd,F}}$	peak detector for no match fields for F $\in$ {color, size, aspect ratio}
$u_{\text{cue}}^{\text{pd,F}}$	peak detector for feature cue fields for F $\in$ {color, size, aspect ratio}

Appendix C

Statistics on Cyclic Metrics

Given measurements along a cyclic metric, such as color hue or angular heading direction, equations for calculating the mean and the error have to be adapted to take circularity into account (an airthmetic mean, for example, does not yield meaningful results on cyclic metrics). The solution used in this thesis is to project all values of the cyclic metric onto points on a unit circle in Cartesian space, mapping the values to angles $[0, 2\pi)$ and using a fixed radius of 1. Arithmetic means are then calculated in Cartesian space $(\mu_{\mathrm{x}}, \mu_{\mathrm{y}})$,

$$\mu_{\mathrm{x}}(A) = \frac{\sum_{i=1}^{n} \cos \alpha_i}{n} \tag{C.1}$$

$$\mu_{\mathrm{y}}(A) = \frac{\sum_{i=1}^{n} \sin \alpha_i}{n}. \tag{C.2}$$

The result is projected back into polar coordinates and only the angle is used (i.e., dropping the magnitude). Calculating the mean μ_{ang} of n values $\alpha_i \in A$,

$$\begin{aligned} \mu_{\mathrm{ang}}(A) &= \operatorname{atan2}\left(\mu_{\mathrm{y}}(A), \mu_{\mathrm{x}}(A)\right) && \text{(C.3)} \\ &= \operatorname{atan2}\left(\frac{\sum_{i=1}^{n} \sin \alpha_i}{n}, \frac{\sum_{i=1}^{n} \cos \alpha_i}{n}\right), && \text{(C.4)} \end{aligned}$$

transforms all values to Cartesian coordinates, applies the arithmetic mean and transforms the resulting point back to polar space. Calculating the error ε between two values α and β also has to take into account the circularity of the metric,

$$\varepsilon_{\mathrm{ang}}(\alpha, \beta) = \begin{cases} |\alpha - \beta| & \text{if } |\alpha - \beta| < \pi \\ 2\pi - |\alpha - \beta| & \text{else.} \end{cases} \tag{C.5}$$

Appendix D

The DFT Software Framework cedar

Cedar[1] is an open-source software framework written in C++ aimed at assembling and simulating DFT architectures in a graphical user interface. Here, architectures are assembled from a pool of elements on a canvas using drag-and-drop. Elements are divided into two categories: *Looped* elements require a time step to produce output. The time step is used for differential equations representing dynamic fields and nodes, as well as time-dependent processes, such as sensor input and motor output. *Non-looped* elements apply an operation onto their input each time this input changes. They are is used to implement various mathematical operations, such as convolution, summation, applying a scalar weight, resizing, and dimensionality expansion and contraction, among others. Among the elements are sensor inputs and motor output that connect a DFT architecture to a robotic body.

Each element may expose a number of parameters to the graphical user interface, which can be examined and altered during assembly and simulation. Certain parameters, such as the sampling size of dynamic fields, are fixed during simulation.

The output of elements can be the input of other elements. Cycles in the graph of connections (i.e., recurrence) are allowed as long as there is at least one looped element in each cycle. The graphical user interface reports possible conflicts requiring an intervention of the user. Examples of such conflicts are a mismatch in field dimensionality or sampling of the continuous field dimensions. Conflicts are resolved by adding elements in-between conflicting sources and targets.

Numerical approximation of differential equations is triggered by timers

[1] `http://cedar.ini.rub.de`

that measure the elapsed time since the last time step and use this measurement as Δt for the current time step. The timers adhere to a configurable minimal time step, with pauses inserted if computation is faster than the minimal step size. This reduces the jitter and thus keeps the approximation error of the forward Euler in a fixed interval.

Inputs, outputs, and internal stages of processing can be plotted during simulation. Suitable plots are automatically chosen based on the dimensionality and annotation of data. Sets of plots can be saved and restored to allow inspection of a subset of elements. Data can be recorded with a parameterizable frequency during simulation.

With *cedar*'s experiment dialog, a given DFT architecture can be simulated using a set of initial conditions and conditional events that may depend on reaching a given simulation time or a specified matrix value, among others. During an experimental trial, associated data is recorded and stored under the trial number.

Additional functionality missing from the core *cedar* framework can be added in plugins.

Appendix E

Curriculum Vitae

Stephan Klaus Ulrich Zibner

Zur Person

Geboren am 05.08.1983 in Hattingen, Deutschland

Berufserfahrung

Seit Sep 2009	Institut für Neuroinformatik, Ruhr-Universität Bochum, Bochum wissenschaftlicher Mitarbeiter und Doktorand
Mai 2006 - März 2008	Fakultät für Psychologie, Arbeitseinheit Sprach- und Kommunikationspsychologie, Ruhr-Universität Bochum, Bochum studentische Hilfskraft
Juli 2003 - April 2004	HELIOS Klinik Holthausen, Hattingen Zivildienst

Ausbildung

2007-2009	Ruhr-Universität Bochum, Bochum Angewandte Informatik (Master of Science)

2004-2007 Ruhr-Universität Bochum, Bochum
Angewandte Informatik (Bachelor of Science)

1990-2003 Städtisches Gymnasium Holthausen, Hattingen
Abitur

Publikationen

Begutachtete Veröffentlichungen in wissenschaftlichen Journals

2011 Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis und Gregor Schöner. Dynamic neural fields as building blocks of a cortex-inspired architecture for robotic scene representation. *Autonomous Mental Development, IEEE Transactions on*, 3(1):74–91, 2011.

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

Begutachtete Konferenzbeiträge

2010 Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis, Gregor Schöner und John P. Spencer. Scenes and tracking with dynamic neural fields: How to update a robotic scene representation. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, Seiten 244–250. IEEE, 2010.

2010 Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis und Gregor Schöner. Scene representation for anthropomorphic robots: A dynamic neural field approach. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, Seiten 927–933. VDE, 2010.

2010 Christian Faubel und Stephan K. U. Zibner. A neuro-dynamic object recognition architecture enhanced by foveal vision and a gaze control mechanism. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Seiten 1171–1176. IEEE, 2010.

2011 Stephan K. U. Zibner, Christian Faubel und Gregor Schöner. Making a robotic scene representation accessible to feature and label queries. In *Development and Learning and on Epigenetic Robotics (ICDL-Epirob), 2011*

First Joint IEEE International Conferences on, August 2011.

2013 Oliver Lomp, Stephan K. U. Zibner, Mathis Richter, Iñaki Rañó und Gregor Schöner. A software framework for cognition, embodiment, dynamics, and autonomy in robotics: cedar. In Valeri Mladenov, Petia Koprinkova-Hristova, Günther Palm, Alessandro E.P. Villa, Bruno Appollini und Nikola Kasabov, Editoren, *Artificial Neural Networks and Machine Learning — ICANN 2013, volume 8131 of Lecture Notes in Computer Science*, Seiten 475–482. Springer Berlin Heidelberg, 2013.

2014 Guido Knips, Stephan K. U. Zibner, Hendrik Reimann, Irina Popova und Gregor Schöner. A neural dynamics architecture for grasping that integrates perception and movement generation and enables on-line updating. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, Seiten 646–653, 2014.

2014 Guido Knips, Stephan K. U. Zibner, Hendrik Reimann, Irina Popova und Gregor Schöner. Reaching and grasping novel objects: Using neural dynamics to integrate and organize scene and object perception with movement generation. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, Seiten 311–318, 2014.

2015 Stephan K. U. Zibner, Jan Tekülve und Gregor Schöner. The neural dynamics of goal-directed arm movements: a developmental perspective. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, Seiten 154–161, 2015.

2015 Stephan K. U. Zibner, Jan Tekülve und Gregor Schöner. The Sequential Organization of Movement is Critical to the Development of Reaching: A Neural Dynamics Account. In *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2015 Joint IEEE International Conferences on*, Seiten 39–46, 2015.

Sonstige Veröffentlichungen

2010 Stephan K. U. Zibner, Christian Faubel, Ioannis Iossifidis und Gregor Schöner. Scene Representation Based on Dynamic Field Theory: From Human to Machine. *Frontiers in Computational Neuroscience*, (19), 2010.

2015 Stephan K. U. Zibner und Christian Faubel. Dynamic Scene Representations and Autonomous Robotics. In *Dynamic Thinking: A Primer on Dynamic*

Field Theory (Seiten 223–246). Oxford University Press, 2015 (in Druck).

Tagungs- und Konferenzteilnahmen

2010 International Symposium on Robotics (ISR), München

2010 IEEE International Conference on Development and Learning (ICDL), Ann Arbor, USA

2010 Bernstein Conference on Computational Neuroscience, Berlin

2011 IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-Epirob), Frankfurt

2011 Interdisziplinäres Kolleg, Günne

2012 5th International Conference on Cognitive Systems, Wien, Österreich

2013 Spatial Memory: Bayes and Beyond, Richmond, USA

2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe

2014 IEEE International Conference on Intelligent Robots and Systems (IROS), Chicago, USA

2014 IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-Epirob), Genua, Italien

2015 IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-Epirob), Providence, USA

Auszeichnungen

2011 "Featured paper", August 2011, IEEE Computational Intelligence Society, Artikel "Dynamic neural fields as building blocks of a cortex-inspired architecture for robotic scene representation"

2015 "Best Student Paper Award", ICDL-Epirob 2015 für Artikel "The Neural Dynamics of Goal-Directed Arm Movements: A Developmental Perspective."

2015 "Runner Up" Platzierung, ICDL-Epirob 2015 Babybot Wettbewerb, Preis-

geld 100 US$ für Artikel “The Sequential Organization of Movement is Critical to the Development of Reaching: A Neural Dynamics Account.”