Perceptual Context in Cognitive Hierarchies

Bernhard Hengst, Maurice Pagnucco, David Rajaratnam, Claude Sammut, and
Michael Thielscher

School of Computer Science and Engineering, UNSW Sydney, Australia
{bernhardh,morri,daver,claude,mit}@cse.unsw.edu.au

Abstract. Cognition does not only depend on bottom-up sensor feature abstraction, but also relies on contextual information being passed top-down. Context is higher level information that helps to predict belief states at lower levels. The main contribution of this paper is to provide a formalisation of perceptual context and its integration into a new process model for cognitive hierarchies. Several simple instantiations of a cognitive hierarchy are used to illustrate the role of context. Notably, we demonstrate the use context in a novel approach to visually track the pose of rigid objects with just a 2D camera.

1 Introduction

There is strong evidence that intelligence necessarily involves hierarchical structures [2, 6, 10, 1, 4, 27, 15, 24, 11, 9, 18, 25, 20, 17, 22, 3]. We recently addressed the formalisation of cognitive hierarchies that allow for the integration of disparate representations, including symbolic and sub-symbolic representations, in a general framework for cognitive robotics [8]. Sensory information processing is upward-feeding, progressively abstracting more complex state features, while behaviours are downward-feeding progressively becoming more concrete, ultimately controlling robot actuators.

However, neuroscience suggests that the brain is subject to top-down cognitive influences for attention, expectation and perception [12]. Higher level signals carry important information to facilitate scene interpretation. For example, the recognition of the Dalmatian, and the disambiguation of the symbol $/\!\!\setminus$ in Figure 1 intuitively show that higher level context is necessary to correctly interpret these images[1]. Furthermore, the human brain is able to make sense of dynamic 3D scenes from light falling on our 2D retina in varying lighting conditions. Replicating this ability is still a challenge in artificial intelligence and computer vision, particularly when objects move relative to each other, can occlude each other, and are without texture. Prior, more abstract contextual knowledge is important to help segment images into objects or to confirm the presence of an object from faint or partial edges in an image.

In this paper we extend the existing cognitive hierarchy formalisation [8] by introducing the notion of perceptual context, which modifies the beliefs of a child

---

[1] Both of these examples appear in [16] but are also well-known in the cognitive psychology literature.

Fig. 1. The image on the left would probably be indiscernible without prior knowledge of Dalmations. The ambiguous symbol $\not\!\!\wedge$ on the right can be interpreted as either an "H" or an "A" depending on the word context.

node given the beliefs of its parent nodes. It is worth emphasising that defining the role of context as a top-down predictive influence on a node's belief state and the corresponding process model that defines how the cognitive hierarchy evolves over time is non-trivial. Our formalisation captures the dual influences of context and behaviour as a predictive update of a node's belief state. Consequently, *the main contribution of this paper is the inclusion and formalisation of contextual influences as a predictive update within a cognitive hierarchy.*

As a meta-framework, the cognitive hierarchy requires instantiation. We provide two simple instantiation examples to help illustrate the formalisation of context. The first is a running example using a small belief network. The second example involves visual servoing to track a moving object. This second example quantifies the benefit of context and demonstrates the role of context in a complete cognitive hierarchy including behaviour generation.

As a third, realistic and challenging example that highlights the importance of context we consider the tracking of the 6 degrees of freedom pose of multiple, possibly occluded, marker-less objects with a 2D camera. We provide a novel instantiation of a cognitive hierarchy for a real robot using the context of a spatial cognitive node modelled using a 3D physics simulator. Note, this formalisation is provided in outline only due to space restrictions.

2   The Architectural Framework

For the sake of brevity the following presentation both summarises and extends the formalisation of cognitive hierarchies as introduced in [8]. We shall, however, highlight how our contribution differs from their work. The essence of this framework is to adopt a meta-theoretic approach, formalising the interaction between abstract cognitive nodes, while making no commitments about the representation and reasoning mechanism within individual nodes.

2.1   Motivating Example

As an explanatory aid to formalising the use of context in a hierarchy we will use the disambiguation of the symbol $\mathrel{/\!\!\backslash}$ in Figure 1 as a simple running example. This system can be modelled as a two layer causal tree updated according to Pearl's Bayesian belief propagation rules [26]. The lower-level layer disambiguates individual letters while the higher-level layer disambiguates complete words (Figure 2). We assume that there are only two words that are expected to be seen, with equal probability: "THE" and "CAT".
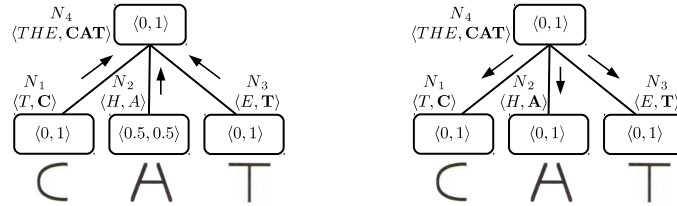

Fig. 2. Disambiguating the symbol $\mathrel{/\!\!\backslash}$ requires context from the word recognition layer.

There are three independent letter sensors with the middle sensor being unable to disambiguate the observed symbol $\mathrel{/\!\!\backslash}$ represented by the conditional probabilities $p(\mathrel{/\!\!\backslash}|H) = 0.5$ and $p(\mathrel{/\!\!\backslash}|A) = 0.5$. These sensors feed into the lower-level nodes (or *processors* in Pearl's terminology), which we label as N_1, N_2, N_3. The results of the lower level nodes are combined at N_4 to disambiguate the observed word.

Each node maintains two state variables; the *diagnostic* and *causal* supports (displayed as the pairs of values in Figure 2). Intuitively, the diagnostic support represents the knowledge gathered through sensing while the causal support represents the contextual bias. A node's overall belief is calculated by the combination of these two state variables.

While sensing data propagates up the causal tree, the example highlights how node N_2 is only able to resolve the symbol $\mathrel{/\!\!\backslash}$ in the presence of contextual feedback from N_4.

2.2   Nodes

A cognitive hierarchy consists of a set of nodes. Nodes are tasked to achieve a goal or maximise future value. They have two primary functions: world-modelling and behaviour-generation. World-modelling involves maintaining a *belief state*, while behaviour-generation is achieved through *policies*, where a policy maps states to sets of actions. A node's belief state is modified by sensing or by the combination of actions and higher-level context. We refer to this latter as *prediction update* to highlight how it sets an expectation about what the node is expecting to observe in the future.

Definition 1. *A cognitive language is a tuple $\mathcal{L} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, \mathcal{C})$, where $\mathcal{S}$ is a set of belief states, $\mathcal{A}$ is a set of actions, $\mathcal{T}$ is a set of task parameters, $\mathcal{O}$ is a set of observations, and $\mathcal{C}$ is a set of contextual elements. A cognitive node is a tuple $N = (\mathcal{L}, \Pi, \lambda, \tau, \gamma, s^0, \pi^0)$ s.t:*

- *$\mathcal{L}$ is the cognitive language for N, with initial belief state $s^0 \in \mathcal{S}$.*
- *Π a set of policies such that for all $\pi \in \Pi$, $\pi : \mathcal{S} \rightarrow 2^{\mathcal{A}}$, with initial policy $\pi^0 \in \Pi$.*
- *A policy selection function $\lambda : 2^{\mathcal{T}} \rightarrow \Pi$, s.t. $\lambda(\{\}) = \pi^0$.*
- *An observation update operator $\tau : 2^{\mathcal{O}} \times \mathcal{S} \rightarrow \mathcal{S}$.*
- *A prediction update operator $\gamma : 2^{\mathcal{C}} \times 2^{\mathcal{A}} \times \mathcal{S} \rightarrow \mathcal{S}$.*

Definition 1 differs from [8] in two ways: the introduction of a set of context elements in the cognitive language, and the modification of the *prediction* update operator, previously called the *action* update operator, to include context elements when updating the belief state.

This definition can now be applied to the motivating example to instantiate the nodes in the Bayesian causal tree. We highlight only the salient features for this instantiation.

Example. *Let $E = \{\langle x, y \rangle \mid 0 \leq x, y \leq 1.0\}$ be the set of probability pairs, representing the recognition between two distinct features. For node N_2, say (cf. Figure 2), these features are the letters "H" and "A" and for N_4 these are the words "THE" and "CAT". The set of belief states for N_2 is $\mathcal{S}_2 = \{\langle\langle d \rangle, c\rangle \mid d, c \in E\}$, where d is the* diagnostic *support and c is the* causal *support. Note, the vector-in-vector format allows for structural uniformity across nodes. Assuming equal probability over letters, the initial belief state is $\langle\langle\langle 0.5, 0.5\rangle\rangle, \langle 0.5, 0.5\rangle\rangle$. For N_4 the set of belief states is $\mathcal{S}_4 = \langle\langle d_1, d_2, d_3\rangle, c\rangle \mid d_1, d_2, d_3, c \in E\}$, where d_i is the contribution of node N_i to the diagnostic support of N_4.*

For N_2 the context *is the causal supports from above, $\mathcal{C}_2 = E$, while the observations* capture the influence of the "H"-"A" sensor, *$\mathcal{O}_2 = \{\langle d \rangle \mid d \in E\}$. In contrast the observations for N_4 need to capture the influence of the different child diagnostic supports, so $\mathcal{O}_4 = \{\langle d_1, d_2, d_3\rangle \mid d_1, d_2, d_3 \in E\}$.*

The observation update operators need to replace the diagnostic supports of the current belief with the observation, which is more complicated for N_4 due to its multiple children, $\tau_2(\{\boldsymbol{d_1}, \boldsymbol{d_2}, \boldsymbol{d_3}\}, \langle \boldsymbol{d}, c\rangle) = \langle \Sigma_{i=1}^3 \boldsymbol{d_i}, c\rangle$. Ignoring the influence of actions, the prediction update operator simply replaces the causal support of the current belief with the context from above, so $\gamma_2(\{c'\}, \emptyset, \langle\langle \boldsymbol{d}\rangle, c\rangle) = \langle\langle \boldsymbol{d}\rangle, c'\rangle$.

2.3 Cognitive Hierarchy

Nodes are interlinked in a hierarchy, where sensing data is passed up the *abstraction hierarchy*, while actions and context are sent down the hierarchy (Figure 3).

Definition 2. *A cognitive hierarchy is a tuple $H = (\mathcal{N}, N_0, F)$ s.t:*

- *$\mathcal{N}$ is a set of cognitive nodes and $N_0 \in \mathcal{N}$ is a distinguished node corresponding to the external world.*
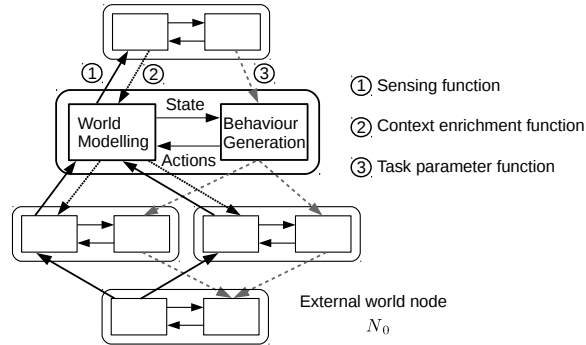
Fig. 3. A cognitive hierarchy, highlighting internal interactions as well as the sensing, action, and context graphs.

- F *is a set of function triples* $\langle \phi_{i,j}, \psi_{j,i}, \varrho_{j,i} \rangle \in F$ *that connect nodes* $N_i, N_j \in \mathcal{N}$ *where:*
  - $\phi_{i,j} : \mathcal{S}_i \to 2^{\mathcal{O}_j}$ *is a sensing function, and*
  - $\psi_{j,i} : 2^{\mathcal{A}_j} \to 2^{\mathcal{T}_i}$ *is a task parameter function.*
  - $\varrho_{j,i} : \mathcal{S}_j \to 2^{\mathcal{C}_i}$ *is a context enrichment function.*
- *Sensing graph: each* $\phi_{i,j}$ *represents an edge from node* N_i *to* N_j *and forms a directed acyclic graph (DAG) with* N_0 *as the unique source node of the graph.*
- *Prediction graph: the set of task parameter functions (equivalently, the context enrichment functions) forms a converse to the sensing graph such that* N_0 *is the unique sink node of the graph.*

Definition 2 differs from the original with the introduction of the *context enrichment* functions and the naming of the *prediction graph* (originally the *action graph*). The connection between nodes consists of triples of sensing, task parameter and context functions. The sensing function extracts observations from a lower-level node in order to update a higher level node, while the context enrichment function performs the converse. The task parameter function translates a higher-level node's actions into a set of task parameters, which is then used to select the active policy for a node.

Finally, the external world is modelled as a distinguished node, N_0. Sensing functions allow other nodes to observe properties of the external world, and task parameter functions allow actuator values to be modified, but N_0 doesn't "sense" properties of other nodes, nor does it generate task parameters for those nodes. Similarly, context enrichment functions connected to N_0 would simply return the empty set, unless one wanted to model unusual properties akin to the quantum effects of observations on the external world. Beyond this, the internal behaviour of N_0 is considered to be opaque.

The running example can now be encoded formally as a cognitive hierarchy, again with the following showing only the salient features of the encoding.

Example. *We construct a hierarchy $H = (\mathcal{N}, N_0, F)$, consisting of five nodes $\mathcal{N} = \{N_0, N_1, \ldots, N_4\}$. The function triples in F will include $\phi_{0,2}$ for the visual sensing of the middle letter, and $\phi_{2,4}$ and $\varrho_{4,2}$ for the sensing and context between N_2 and N_4.*

The function $\phi_{0,2}$ returns the probability of the input being the characters "H" and "A". Here $\phi_{0,2}(\diagup\diagdown) = \{\langle 0.5, 0.5 \rangle\}$.

Defining $\phi_{2,4}$ and $\varrho_{4,2}$ requires a conditional probability matrix $M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ to capture how the letters "H" and "A" contribute to the recognition of "THE" and "CAT".

For sensing from N_2 we use zeroed vectors to prevent influence from the diagnostic support components from N_1 and N_2. Hence $\phi_{2,4}(\langle\langle d\rangle, c\rangle) = \{\langle\langle 0, 0\rangle, \eta \cdot M \cdot d^T, \langle 0, 0\rangle\rangle\}$, where d^T is the transpose of vector d, and η is a normalisation constant.

For context we capture how N_4's causal support and its diagnostic support components from N_1 and N_2 influences the causal support of N_2. Note that this also prevents any feedback from N_2's own diagnostic support to its causal support. So, $\varrho_{4,2}(\langle\langle d_1, d_2, d_3\rangle, c\rangle) = \{\eta \cdot (d_1 \cdot d_3 \cdot c) \cdot M\}$.

2.4   Active Cognitive Hierarchy

The above definitions capture the static aspects of a system but require additional details to model its operational behaviour. Note, the following definitions are unmodified from the original formalism and are presented here because they are necessary to the developments of later sections.

Definition 3. *An active cognitive node is a tuple $Q = (N, s, \pi, a)$ where: 1) N is a cognitive node with $\mathcal{S}$, Π, and $\mathcal{A}$ being its set of belief states, set of policies, and set of actions respectively, 2) $s \in \mathcal{S}$ is the current belief state, $\pi \in \Pi$ is the current policy, and $a \in 2^{\mathcal{A}}$ is the current set of actions.*

Essentially an active cognitive node couples a (static) cognitive node with some dynamic information; in particular the current belief state, policy and set of actions.

Definition 4. *An active cognitive hierarchy is a tuple $\mathcal{X} = (H, \mathcal{Q})$ where H is a cognitive hierarchy with set of cognitive nodes $\mathcal{N}$ such that for each $N \in \mathcal{N}$ there is a corresponding active cognitive node $Q = (N, s, \pi, a) \in \mathcal{Q}$ and vice-versa.*

The active cognitive hierarchy captures the dynamic state of the system at a particular instance in time. Finally, an *initial active cognitive hierarchy* is an active hierarchy where each node is initialised with the initial belief state and policy of the corresponding cognitive node, as well as an empty set of actions.

2.5   Cognitive Process Model

The *process model* defines how an active cognitive hierarchy evolves over time and consists of two steps. Firstly, sensing observations are passed up the hierarchy, progressively updating the belief state of each node. Next, task parameters and context are passed down the hierarchy updating the active policy, the actions, and the belief state of the nodes.

We do not present all definitions here, in particular we omit the definition of the *sensing update* operator as this remains unchanged in our extension. Instead we define a *prediction update* operator, replacing the original *action update*, with the new operator incorporating both context and task parameters in its update. First, we characterise the updating of the beliefs and actions for a single active cognitive node.

Definition 5. *Consider an active cognitive hierarchy $\mathcal{X} = (H, \mathcal{Q})$ where $H = (\mathcal{N}, N_0, F)$. The prediction update of $\mathcal{X}$ with respect to an active cognitive node $Q_i = (N_i, s_i, \pi_i, a_i) \in \mathcal{Q}$, written $\mathbf{PredUpdate}'(\mathcal{X}, Q_i)$, is an active cognitive hierarchy $\mathcal{X}' = (H, \mathcal{Q}')$ where $\mathcal{Q}' = \mathcal{Q} \backslash \{Q_i\} \cup \{Q_i'\}$ and $Q_i' = (N_i, \gamma_i(C, a_i', s_i), \pi_i', a_i')$ s.t:*

- *if there is no node N_x where $\langle \phi_{i,x}, \psi_{x,i}, \varrho_{x,i} \rangle \in F$ then: $\pi_i' = \pi_i$, $a_i' = \pi_i(s_i)$ and $C = \emptyset$,*
- *else:*
  $\pi_i' = \lambda_i(T)$ *and* $a_i' = \pi_i'(s_i)$
  $T = \bigcup \{\psi_{x,i}(a_x) \mid \langle \phi_{i,x}, \psi_{x,i}, \varrho_{x,i} \rangle \in F$ *where* $Q_x = (N_x, s_x, \pi_x, a_x) \in \mathcal{Q}\}$
  $C = \bigcup \{\varrho_{x,i}(s_x) \mid \langle \phi_{i,x}, \psi_{x,i}, \varrho_{x,i} \rangle \in F$ *where* $Q_x = (N_x, s_x, \pi_x, a_x) \in \mathcal{Q}\}$

The intuition for Definition 5 is straightforward. Given a cognitive hierarchy and a node to be updated, the update process returns an identical hierarchy except for the updated node. This node is updated by first selecting a new active policy based on the task parameters of all the connected higher-level nodes. The new active policy is applied to the existing belief state to generate a new set of actions. Both these actions and the context from the connected higher-level nodes are then used to update the node's belief state.

Using the single node update, updating the entire hierarchy simply involves successively updating all its nodes.

Definition 6. *Consider an active cognitive hierarchy $\mathcal{X} = (H, \mathcal{Q})$ where $H = (\mathcal{N}, N_0, F)$, and let Ψ be the prediction graph induced by the task parameter functions in F. The action process update of $\mathcal{X}$, written $\mathbf{PredUpdate}(\mathcal{X})$, is an active cognitive model:*

$$\mathcal{X}' = \mathbf{PredUpdate}'(\dots \mathbf{PredUpdate}'(\mathcal{X}, Q_n), \dots Q_0)$$

where the sequence $[Q_n, \dots, Q_0]$ consists of all active cognitive nodes of the set $\mathcal{Q}$ such that the sequence satisfies the partial ordering induced by the prediction graph Ψ.

Importantly, the update ordering in Definition 6 satisfies the partial ordering induced by the prediction graph, thus guaranteeing that the prediction update is well-defined.

Lemma 1. *For any active cognitive hierarchy $\mathcal{X}$ the prediction process update of $\mathcal{X}$ is well-defined.*

Proof. Follows from the DAG structure.

The final part of the process model, which we omit here, is the combined operator, **Update**, that first performs a sensing update followed by a prediction update. This operation follows exactly the original, and similarly the theorem that the process model is well-defined also follows.

We can now apply the update process (sensing then prediction) to show how it operates on the running example.

Example. *When N_2 senses the symbol $\vdash\!\!\backslash$, $\phi_{0,2}$ returns that "A" and "H" are equally likely, so τ_2 updates the diagnostic support of N_2 to $\langle\langle 0.5, 0.5\rangle\rangle$. On the other hand N_1 and N_2 unambiguously sense "C" and "T" respectively, so N_4's observation update operator, τ_4, will update its diagnostic support components to $\langle\langle 0, 1\rangle, \langle 0.5, 0.5\rangle, \langle 0, 1\rangle\rangle$. The nodes overall belief, $\langle 0, 1\rangle$, is the normalised product of the diagnostic support components and the causal support, indicating here the unambiguous recognition of "CAT".*

Next, during prediction update, context from N_4 is passed back down to N_2, through $\phi_{4,2}$ and γ_2, updating the causal support of N_2 to $\langle 0, 1\rangle$. Hence, N_2 is left with the belief state $\langle\langle 0.5, 0.5\rangle\rangle, \langle 0, 1\rangle\rangle$, which when combined, indicates that the symbol $\vdash\!\!\backslash$ should be interpreted as an "A".

We next appeal to another simple example to illustrate the use of context to improve world modelling and in turn behaviour generation in a cognitive hierarchy.

3   A Simple Visual Servoing Example

Consider a mobile camera tasked to track an object sliding down a frictionless inclined plane. The controller is constructed as a three-node cognitive hierarchy. Figure 4 depicts the cognitive hierarchy and the scene.

The performance of the controller will be determined by how well the camera keeps the object in the centre of its field-of-view, specifically the average error in the tracking distance over a time period of 3 seconds.

The details of the instantiation of the cognitive hierarchy controller follow. The cognitive hierarchy is $H = (\mathcal{N}, N_0, F)$ with $\mathcal{N} = \{N_0, N_1, N_2\}$. N_0 is the unique opaque node representing the environment. The cognitive language for N_1 is a tuple $\mathcal{L}_1 = (\mathcal{S}_1, \mathcal{A}_1, \mathcal{T}_1, \mathcal{O}_1, \mathcal{C}_1)$, and for N_2 it is $\mathcal{L}_2 = (\mathcal{S}_2, \mathcal{A}_2, \mathcal{T}_2, \mathcal{O}_2, \mathcal{C}_2)$. The cognitive nodes are $N_1 = (\mathcal{L}_1, \Pi_1, \lambda_1, \tau_1, \gamma_1, s_1^0, \pi_1^0)$ and $N_2 = (\mathcal{L}_2, \Pi_2, \lambda_2, \tau_2, \gamma_2, s_2^0, \pi_2^0)$. For brevity we only describe the material functions.
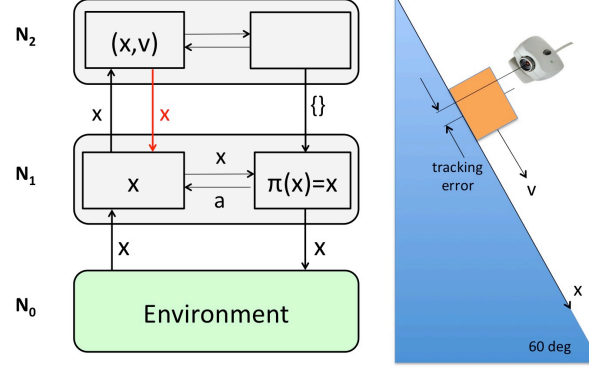
Fig. 4. A three-node cognitive hierarchy controller tasked to visually follow an object. Context flow is shown in red.

The belief state of N_1 is the position of the object: $\mathcal{S}_1 = \{x \mid x \in \mathbb{R}\}$. The belief state of N_2 is both the position and velocity of the object: $\mathcal{S}_2 = \{\langle x, v \rangle \mid x, v \in \mathbb{R}\}$. The object starts at rest on the inclined plane at the origin: $s_1^0 = 0.0$ and $s_2^0 = \langle 0.0, 0.0 \rangle$.

N_1 receives object position observations from the environment: $\mathcal{O}_1 = \{x \mid x \in \mathbb{R}\}$. These measurements are simulated from the physical properties of the scene and include a noise component to represent errors in the sensor measurements: $\phi_{0,1}(\cdot) = \{0.5kt^2 + \nu\}$, with constant acceleration $k = 8.49 \ m/s^2$, t the elapsed time and ν zero mean Gaussian random noise with a standard deviation of 0.1. The acceleration assumes an inclined plane of 60 degrees in a $9.8 \ m/s^2$ gravitational field. The N_1 observation update operator implements a Kalman filter with a fixed gain of 0.25: $\tau_1(\langle \{x\}, y \rangle) = (1.0 - 0.25)y + 0.25x$.

N_2 receives observations $\mathcal{O}_2 = \{x \mid x \in \mathbb{R}\}$ from N_1: $\phi_{1,2}(x) = \{x\}$. In turn it updates its position estimate accepting the value from N_1: $\tau_2(\langle \{x\}, \langle y, v \rangle \rangle) = \langle x, v \rangle$. The prediction update operator uses a physics model to estimate the new position and velocity of the object after time-step $\delta t = 0.05$ seconds: $\gamma_2(\langle \{\}, \{\}, \langle x, v \rangle \rangle) = \langle x + v\delta t + 0.5k\delta t^2, v + k\delta t \rangle$ with known acceleration $k = 8.49$.

Both N_1 and N_2 have one policy function each. The N_2 policy selects the N_1 policy. The effect of the N_1 policy: $\pi_1(x) = \{x\}$, is to move the camera to the estimated position of the object via the task parameter function connecting the environment: $\psi_{1,0}(\{x\}) = \{x\}$.

We consider two versions of the N_1 prediction update operator. Without context the next state is the commanded policy action: $\gamma_1(\langle \{x\}, \{y\}, z \rangle) = y$. With context the context enrichment function passes the N_2 estimate of the position of the object to N_1: $\varrho_{2,1}(\langle x, v \rangle) = \{x\}$, where $\mathcal{C}_1 = \{x \mid x \in \mathbb{R}\}$. The update operator becomes: $\gamma_1(\langle \{x\}, \{y\}, z \rangle) = x$.

When we simulate the dynamics and the repeated update of the cognitive hierarchy at $1/\delta t$ Hertz for 3 seconds, we find that without context the average

tracking error is 2.004 ± 0.009. Using context the average tracking error reduces to 0.125 ± 0.015—a 94% error reduction.[2]

4   Using Context to Track Objects Visually

Object tracking has applications in augmented reality, visual servoing, and man-machine interfaces. We consider the problem of on-line monocular model-based tracking of multiple objects without markers or texture, using the 2D RGB camera built into the hand of a Baxter robot. The use of natural object features makes this a challenging problem.

Current practice for tackling this problem is to use 3D knowledge in the form of a CAD model, from which to generate a set of edge points (control points) for the object [21] . The idea is to track the corresponding 2D camera image points of the visible 3D control points as the object moves relatively to the camera. The new pose of the object relative to the camera is found by minimising the perspective re-projection error between the control points and their corresponding 2D image.

However, when multiple objects are tracked, independent CAD models fail to handle object occlusion. In place of the CAD models we use the machinery provided by a 3D physics simulator. The object-scene and virtual cameras from a simulator are ideal to model the higher level context for vision. We now describe how this approach is instantiated as a cognitive hierarchy with contextual feedback. It is important to note that the use of the physics simulator is not to replace the real-world, but is used as mental imagery efficiently representing the spatial belief state of the robot.

4.1   Cognitive Hierarchy for Visual Tracking

We focus on world-modelling in a two-node cognitive hierarchy (Figure 5). The external world node that includes the Baxter robot, streams the camera pose and RGB images as sensory input to the arm node. The arm node belief state is $s = \{p^a\} \cup \{\langle p_a^i, c^i \rangle | \text{object } i\}$, where p^a is the arm pose, and for all recognised objects i in the field of view of the arm camera, p_a^i is the object pose relative to the arm camera, and c^i is the set of object edge lines and their depth. The objects in this case include scattered cubes on a table. Information from the arm node is sent to the spatial node that employs a Gazebo physics simulator [19] as mental imagery to model the objects.

A novel feature of the spatial node is that it simulates the robot's arm camera as an object aware depth camera. No such camera exists in reality, but the Gazebo spatial belief state of the robot is able to not only provide a depth image, but one that segments the depth image by object. This object aware depth image provides the context to the arm node to generate the required control points.

---

[2] It is of course intuitive in this simple example that as N_2 has the benefit of the knowledge of the transition dynamics of the object it can better estimate its position and provide this context to direct the camera.
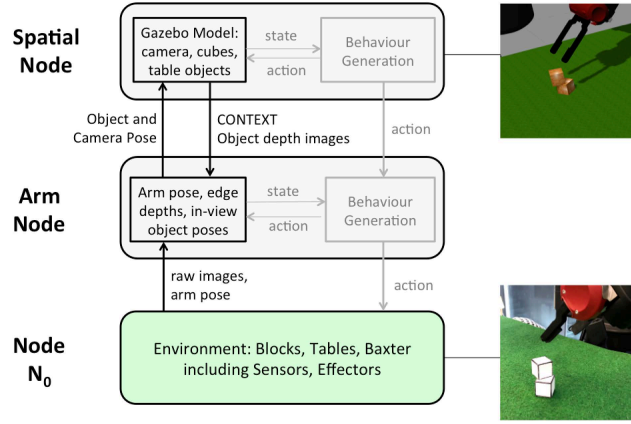
Fig. 5. Cognitive hierarchy comprising an arm node and a spatial node. Context from the spatial node is in the form of an object segmented depth image from a simulated special camera that shadows the real camera.

4.2 Update Functions and Process Update

We now describe the update functions and a single cycle of the process update for this cognitive hierarchy.

The real monocular RGB arm camera is simulated in Gazebo with an object aware depth camera with identical characteristics (i.e. the same intrinsic camera matrix). The simulated camera then produces depth and an object segmentation images from the simulated objects that corresponds to the actual camera image. This vital contextual information is then used for correcting the pose of the visible objects.

The process update starts with the sensing function $\phi_{N_0, Arm}$ that takes the raw camera image and observes all edges in the image, represented as a set of line segments, l.

$$\phi_{N_0, Arm}(\{rawImage\}) = \{l\}$$

The observation update operator τ_{Arm} takes the expected edge lines c^i for each object i and transforms the lines to best match the image edge lines l [21]. The update function uses the OpenCV function *solvePnP* to find a corrected pose p_a^i for each object i relative to the arm-camera a [3].

$$\tau_{Arm}(\{l, c^i | \text{object } i\}) = \{p_a^i | \text{object } i\}$$

The sensing function from the arm to spatial node takes the corrected pose p_a^i for each object i, relative to the camera frame a, and transforms it into the

---

[3] The pose of a rigid object in 3D space has 6 degrees of freedom, three describing its translated position, and three the rotation or orientation, relative to a reference pose.
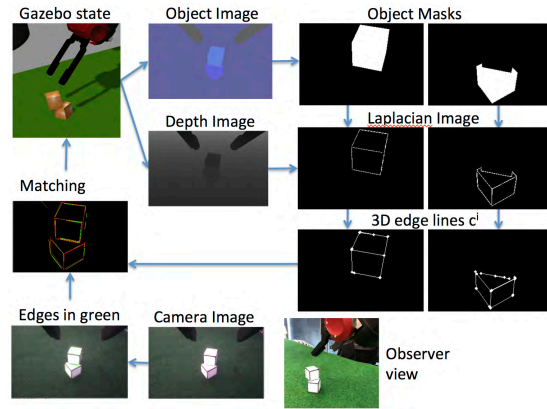
Fig. 6. The Process update showing stages of the context enrichment function and the matching of contextual information to the real camera to correct the arm and spatial node belief state.

Gazebo reference frame via the Baxter's reference frame given the camera pose p^a.

$$\phi_{Arm,Spatial}(\{p^a, \langle p_a^i, c^i \rangle | \text{object } i\}) = \{g_a^i | \text{object } i\}$$

The spatial node observation update $\tau_{Spatial}$, updates the pose of all viewed objects g_a^i in the Gazebo physics simulator. Note $\{g_a^i | \text{object } i\} \subset$ gazebo state.

$$\tau_{Spatial}(\{g_a^i | \text{object } i\}) = \text{gazebo.move}(i, g_a^i) \ \ \forall i$$

The update cycle now proceeds down the hierarchy with prediction updates. The prediction update for the spatial node $\gamma_{Spatial}$ consists of predicting the interaction of objects in the simulator under gravity. Noise introduced during the observation update may result in objects separating due to detected collisions or settling under gravity.

$$\gamma_{Spatial}(\text{gazebo state}) = \text{gazebo.simulate(gazebo state)})$$

We now turn to the context enrichment function $\varrho_{Spatial,Arm}$ that extracts predicted camera image edge lines and depth data for each object in view of the simulator.

$$\varrho_{Spatial,Arm}(\text{gazebo state}) = \{c^i | \text{object } i\}$$

The stages of the context enrichment function $\varrho_{Spatial,Arm}$ are shown in Figure 6. The simulated depth camera extracts an object image that identifies the object seen at every pixel location. It also extracts a depth image that gives the depth from the camera of every pixel. The object image is used to mask out each object in turn. Applying a Laplacian function to the part of the depth image masked out by the object yields all visible edges of the object. A Hough line transform identifies line end points in the Laplacian image and finds the depth of their endpoints from the depth image, producing c^i.
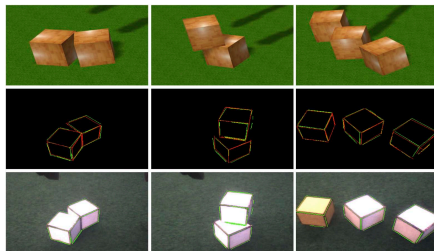
Fig. 7. Tracking several cube configurations. Top row: Gazebo GUI showing spatial node state. 2nd row: matching real image edges in green to simulated image edges in red. Bottom row: camera image overlaid with edges in green.

Figure 7 shows the cognitive hierarchy tracking several different cube configurations. This is only possible given the context from the spatial belief state. Keeping track of the pose of objects allows behaviours to be generated that for example pick up a cube with appropriately oriented grippers.

5   Related Support and Conclusion

There is considerable evidence supporting the existence and usefulness of top-down contextual information. Reliability [5] and speed [7] of scene analysis provide early evidence.

These observations are further supported by neuroscience, suggesting that feedback pathways from higher more abstract processing areas of the brain down to areas closer to the sensors are greater than those transmitting information upwards [13]. The authors summarise the process — "what is actually happening flows up, and what you expect to happen flows down". It has been argued that the traditional idea that the processing of visual information consists of a sequence of feedforward operations needs to be supplemented by top-down contextual influences [12].

In the field of robotics, recent work in online interactive perception shows the benefit of predicted measurements from one level being passed to the next-lower level as state predictions [23].

This paper has included and formalised the essential element of context in the meta framework of cognitive hierarchies. The process model of an active cognitive hierarchy has been revised to include context updates satisfying the partial order induced by the prediction graph. We have illustrated the role of context with two simple examples and a novel way to track the pose of texture-less objects with a single 2D camera. Our motivating example highlighted the use of context in a cognitive hierarchy inspired by belief propagation in causal trees. In fact, as a general result it can be proved [14] that any Bayesian causal tree can be encoded as a cognitive hierarchy, testifying to the representation versatility of our framework.

Acknowledgments

Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the AOARD.

References

1. Albus, J.S., Meystel, A.M.: Engineering of Mind: An Introduction to the Science of Intelligent Systems. Wiley-Interscience (2001), http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471438545.html
2. Ashby, W.R.: Design for a Brain. Chapman and Hall (1952), https://archive.org/details/designforbrainor00ashb
3. Bakker, B., Schmidhuber, J.: Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In: Proceedings of the 8-th Conference on Intelligent Autonomous Systems, IAS-8. pp. 438–445 (2004)
4. Beer, S.: Decision and Control. John Wiley and Sons, London (1966)
5. Biederman, I., Kubovy, M., Pomerantz, J.: On the semantics of a glance at a scene. In: Perceptual Organization, pp. 213–263. Lawrence Erlbaum, New Jersey (1981)
6. Brooks, R.A.: A robust layered control system for a mobile robot. Robotics and Automation, IEEE Journal of **2**(1), 14–23 (Mar 1986). https://doi.org/10.1109/JRA.1986.1087032
7. Cavanagh, P.: What's up in top-down processing? In: Gorea, A. (ed.) Representations of Vision: Trends and tacit assumptions in vision research. pp. 295–304 (1991)
8. Clark, K., Hengst, B., Pagnucco, M., Rajaratnam, D., Robinson, P., Sammut, C., Thielscher, M.: A framework for integrating symbolic and sub-symbolic representations. In: 25th Joint Conference on Artificial Intelligence (IJCAI -16) (2016)
9. Dayan, P., Hinton, G.E.: Feudal reinforcement learning. Advances in Neural Information Processing Systems 5 (NIPS) (1992)

10. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Journal of Artificial Intelligence Research (JAIR) **13**, 227–303 (2000)
11. Drescher, G.L.: Made-up Minds: A constructionist Approach to Artificial Intelligence. MIT Press, Cambridge, Massachusetts (1991)
12. Gilbert, C.D., Li, W.: Top-down influences on visual processing. Nature reviews. Neuroscience **14**(5), 10.1038/nrn3476 (05 2013). https://doi.org/10.1038/nrn3476
13. Hawkins, J., Blakeslee, S.: On Intelligence. Times Books, Henry Holt and Company (2004)
14. Hengst, B., Pagnucco, M., Rajaratnam, D., Sammut, C., Thielscher, M.: Perceptual context in cognitive hierarchies. CoRR **abs/1801.02270** (2018), http://arxiv.org/abs/1801.02270
15. Hubel, D.H., Wiesel, T.N.: Brain mechanisms of vision. A Scientific American Book: the Brain pp. 84–96 (1979)
16. Johnson, J.: Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2010)
17. Jong, N.K.: Structured Exploration for Reinforcement Learning. Ph.D. thesis, University of Texas at Austin (2010)
18. Kaelbling, L.P.: Hierarchical learning in stochastic domains: Preliminary results. In: Machine Learning Proceedings of the Tenth International Conference. pp. 167–173. Morgan Kaufmann, San Mateo, CA (1993)
19. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: International Conference on Intelligent Robots and Systems (IROS). pp. 2149–2154. IEEE/RSJ (2004)
20. Konidaris, G., Kuindersma, S., Grupen, R., Barto, A.: Robot learning from demonstration by constructing skill trees. The International Journal of Robotics Research (2011). https://doi.org/10.1177/0278364911428653
21. Lepetit, V., Fua, P.: Monocular model-based 3d tracking of rigid objects. Found. Trends. Comput. Graph. Vis. **1**(1), 1–89 (Jan 2005). https://doi.org/10.1561/0600000001
22. Marthi, B., Russell, S., Andre, D.: A compact, hierarchical q-function decomposition. In: Proceedings of the Proceedings of the Twenty-Second Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06). pp. 332–340. AUAI Press, Arlington, Virginia (2006)
23. Martin, R.M., Brock, O.: Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In: IROS. pp. 2494–2501. IEEE (2014)
24. Minsky, M.: The Society of Mind. Simon & Schuster, Inc., New York, NY, USA (1986)
25. Nilsson, N.J.: Teleo-Reactive programs and the triple-tower architecture. Electronic Transactions on Artificial Intelligence **5**, 99–110 (2001)
26. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francesco, revised second printing edn. (1988)
27. Turchin, V.F.: The Phenomenon of Science. Columbia University Press (1977)