Honda Research Institute EU

Honda Research Institute Europe GmbH
https://www.honda-ri.de/

From neurons to cortex: a multi-level approach to understanding the brain

Rüdiger Kupper, Marc-Oliver Gewaltig, Andreas Knoblauch, Ursula Körner, Edgar Körner

2007

Preprint:

This is an accepted article published in Neurocomputing Research Trends. The final authenticated version is available online at: https://doi.org/[DOI not available]

From neurons to cortex: a multi-level approach to understanding the brain

Rüdiger Kupper, Marc-Oliver Gewaltig, Andreas Knoblauch, Ursula Körner, and Edgar Körner

Affiliation for all authors:
Honda Research Institute Europe GmbH,
Carl-Legien-Str. 30, D-63073 Offenbach/Main, Germany

Contents

Abstract

At the Honda Research Institute, we aim to understand the operating principles of the brain. We think that the cortex is composed of elementary building blocks, the columns, that apply one generic algorithm to varying sensory data. The brain is thus not a collection of highly specialized neural circuits, providing tailored solutions to individual problems, but it uses the same set of powerful processing strategies over and over again.

Based on neurobiological knowledge about the primate cortex we substantiate this idea in a model of the visual system, on several levels of detail.

At the single neuron level, the visual system can immediately profit from a spike-latency code to rapidly segment and recognize scenes. A wave of spikes traveling through a cascade of feature detectors rapidly activates a high-level hypothesis about stimulus content, so that an appropriate reaction (e.g. escape) is possible.

At the level of neural circuits, we simulate the signal flow in columns, across the six cortical layers and between several cortical areas. We show how the columnar subsystems interact to predict and recognize stimuli in terms of acquired knowledge. Columnar communication integrates top-down and bottom-up signals to describe the stimulus consistently across all cortical areas. It iterates this process to refine the description, causing oscillations in neural activity. Internal descriptions of entirely new stimuli can be constructed from old ones, and entrained with the help of the hippocampal formation.

At the system level, we implement a large-scale model of main visual cortical areas, parts of the hippocampal formation, and sub-cortical structures. Cortical columns both predict future stimuli and vote for motor actions to confirm them. Here, they control saccades to learn and recognize objects from a sequence of partial views, based on step-by-step prediction and refinement of an object hypothesis in time.

In our models the columns link the neural level to the system level. They help us to understand how groups of nerve cells ultimately create the macroscopic function of the brain.

1 Introduction

Animals at the phylogenetic level of amphibians do not yet have developed a cortex; they select their responses from a limited repertoire of alternatives, by evaluating a small number of trigger features in their stimuli. Feed-forward models of visual processing often use the same strategy, they use *a priori* information to limit the number of possible alternatives. The advent of the cortex in phylogenetic development allowed animals an increasingly deeper analysis of their sensory input. The neocortex has evolved to find structure and regularity in the barrage of afferent signals and to interpreting the sensory events according to the animal's needs.

But the ability of the cortex to interpret sensory information in different ways comes at a price: Sensory stimuli regularly include complex and ambiguous scenes. For a brain with cortex, it is more difficult to interpret a sensory input due to the combinatorial explosion of possible alternative interpretations, compared to the simple detection of a few trigger features. The widely accepted assumption is that the cortex uses previously acquired knowledge that is projected top-down to guide the analysis of the sensory input.

Models of bidirectional neocortical processing focus on the iterative refinement of recognition, using top-down prediction. A plausible mechanism of how prediction could speed up the interpretation of a visual scene is the removal of already recognized parts from the input and to continue with the residual parts. A number of recent models follow this strategy. But even though they can explain some experimentally observed phenomena, they fail to explain a number of phenomena at the psychophysical and single-neuron levels: First, there is evidence from a number of studies that already the first wave of spikes that follow a stimulus evoke a correct and sufficiently detailed initial hypothesis in the inferotemporal cortex. Second, at least in the ventral visual pathway of primates many neurons respond as long as the stimulus is present. This would not be the case if the recognized parts had been removed from the input.

Iterative refinement takes time and should only be necessary to discriminate similar stimuli but not to categorize them. Thus, the initial hypothesis must be precise enough to grasp the gist of a scene and at the same time available fast enough to select the appropriate response and to initiate refinement. Here we describe our theory [25] of processing in the cortex. At the system level, it has two main ingredients: a feed-forward system that quickly categorizes the stimulus, and a slower feed-back controlled refinement system. At the level of cortex areas, we assume that the specific columnar architecture of the neocortex reflects this basic computational principle: Each column hosts a fast feed-forward system in the cortex layer 4 and a refinement system in layers 2/3. At the level of neurons we assume that spikes play an important role in rapidly coding and transmitting information.

1.1 Assumptions

1.1.1 Stimulus hypothesis

Experimental evidence suggests that awake animals have an internal representation of the world that allows them to plan and execute their actions. If the sensory stimulus changes significantly, the internal representation is modified or replaced by a new one. In the cortex an internal representation, which we call *hypothesis*, is generated from the sensory input and guides all subsequent processing by limiting the ways in which the raw input can be interpreted.

1.1.2 Columnar architecture

Many cortical areas, including the ventral visual pathway of primates [36] have a columnar architecture. In our model the cortex is described as a homogeneous network of columns that link the cortical layers into computational units. The structure of columns is roughly the same in all areas, regardless of the information that is processed there. We further assume, that computation within the cortex can be described in terms of columns and their interactions. While a *minicolumn* is the smallest functional module of the cortex, a *hypercolumn* is the set of all minicolumns that have the same receptive field (for the sensory cortices). In non-topographic areas, a hypercolumn may comprise all descriptions related to one object.

1.1.3 Spikes and latency code

There is convincing experimental [19, 28, 34] and theoretical [10, 26, 34, 44, 45] evidence that the spike latency of neurons encodes their stimulation strength. Experimental evidence for spike latency coding in the visual system of primates can already be found in the classical work of Hubel and Wiesel [18]. Figure 1 shows recordings from orientation-selective cells in monkey primary visual cortex. It shows spike trains and mean firing rates for different orientations of a bar stimulus. The panel on the right is based on an original figure from Hubel and Wiesel [18]. The gray bars indicate the latency of the first spike. Obviously, the spike latency is a good estimate of the firing rate: The latency is large when firing rate is small, and vice versa. But unlike the firing rate which must be integrated over longer time intervals, the spike latency can be assessed from the first spike of a neuron.

1.2 Chapter overview

This chapter describes our progress in substantiating this theory. In the following section we demonstrate, how response spike latencies of neurons can quickly generate a stimulus hypothesis. We start with a model of early visual processing in the cortex that shows how the response latency of visual cortical neurons can help to detect homogeneous image regions. With this information the cortex can quickly segment prominent objects in the scene to contribute to the global hypothesis. We then extend the latency concept to other types of

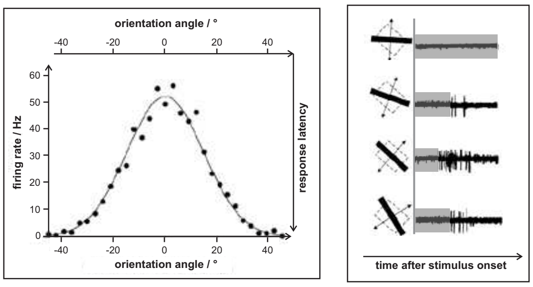


Figure 1: Firing rate and response latency of typical primary visual neurons. *Left,* average firing rate of a cat V1 neuron plotted as a function of the orientation angle of a light bar stimulus; *right,* recordings from a neuron in the primary visual cortex of a monkey, diagrams to the left of each trace show the receptive field as a dashed square and the light source as a black bar; *shaded region,* latency of the first spike in response to the bar stimulus. Response latency and firing rate carry similar information: latency of the first response spike is large when firing rate is small, and vice versa. [Figure adapted from 8].

feature detectors and show that a latency code can indeed be used in a feed-forward network of feature detectors.

In section 3 we demonstrate how the cortex uses its six layers to analyze and interpret sensory signals. The model consists of three simplified cortical areas, each including a feed-forward system to quickly generate a stimulus hypothesis (A-system) and a feed-back controlled system (B-system) to refine the hypothesis. In this model we demonstrate how the two systems (A and B) of our columnar cortex improve the recognition of stimuli by including previously acquired knowledge into the analysis and interpretation of stimuli.

In section 4 we present a model that includes visual fixations and saccades and that can learn new objects and object views. We demonstrate how the cortex learns and predicts parts of the stimulus and uses this information to guide recognition. Given the partial view of an object, the model will generate a hypothesis and saccade to a new fixation point to confirm or modify its hypothesis.

2 From latency code to fast hypothesis generation

In early visual processing, speed matters. If an animal encounters a visual scene, it is important to quickly arrive at an hypothesis about its main constituents so that an appropriate reaction (e.g. escape) is possible. Thorpe et al. [38] have shown, that the visual cortex of human subjects can solve the problem of scene categorization in less than 150 ms. In this section we describe two models that show how neurons in the brain can quickly interpret sensory data, by encoding it in their first action potentials following stimulus onset. These models operate on the level of single neurons and spikes.

2.1 Homogeneity detection in the visual system

To interpret the visual input it is important to segment those parts of a scene that potentially make up relevant objects. But how can the visual system segment relevant regions, when a hypothesis about objects in the scene is not yet available? We propose that the brain uses homogeneous regions to label potentially relevant objects in a scene.

In the following model, neurons act as *feature detectors* in the sense that the better the stimulus matches their preferred feature, the faster they respond with a spike. Thus, the time interval between stimulus and response spike, also called *latency*, encodes how well the stimulus matched the preferred feature of the neuron. This latency code results from the biophysical property of neurons.

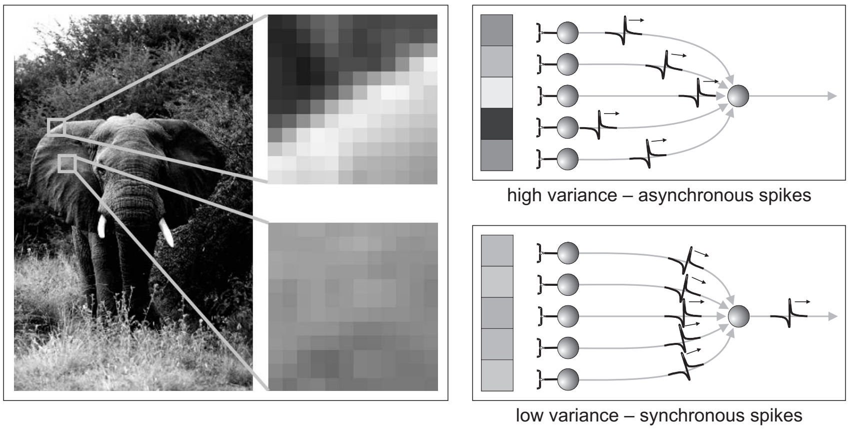

high variance – asynchronous spikes
low variance – synchronous spikes

Figure 2: The principle of spike-latency based homogeneity processing; example from the visual domain. *Left:* Edge regions in the stimulus typically have high variance of luminance, while surface regions typically have low variance of luminance. *Right:* A set of visual neurons respond at latencies depending on the luminance inside their receptive field. The receiving neuron responds when the incoming action potentials coincide. This makes it selective for homogeneous luminance inside its receptive field. In the visual domain, this allows identifying surface regions.

Consider a neuron receiving input from a population of neurons, all responding to the same feature, but at different retinal positions. Because cortical neurons effectively work as coincidence detectors [see e.g. 2, 24], the post-synaptic neuron will then respond best, if all pre-synaptic neurons fire at the same time. For example, if the pre-synaptic neurons encode the luminance in their receptive field, the post-synaptic neuron will respond best to an homogeneous illumination of its receptive field, because in this case all luminance detectors receive the same input luminance. The visual system can use this information on homogeneous image regions, to confine edge detection to borders between mid-sized objects. This principle is illustrated in figure 2. The left panel shows two different patches of an image. The upper patch shows a pronounced gradient, corresponding to an edge of the elephant's ear. The lower patch is very homogeneous, corresponding to the surface of the ear. Suppose that each pixel is the receptive field of a neuron that encodes the luminance in its spike latency and that all luminance detectors in a patch project to a common target neuron (whose receptive field is the patch). Thus, a target neuron whose receptive field corresponds to the upper patch will receive a dispersed set of spikes from its input neurons (fig. 2 *top right*), while a target neuron, looking at the homogeneous patch will receive a synchronous volley of input spikes that will a response of the target neuron (fig. 2 *bottom right*). Note that the response of the target neuron does not depend on how well the individual luminance detectors were activated, but rather on that they are all activated in the same way.

2.1.1 Model

To illustrate this principle, consider a 100×100 sheet N_s of latency-coding neurons, each selective for the luminance in its receptive field. That is, a neuron in N_s responds fastest if the luminance in its receptive field is maximal. The sheet N_s is arranged such that it preserves retinotopy (fig. 3). In addition there is a sheet N_r of neurons, receiving convergent input from a local neighborhood d of neurons in N_s. Since the neurons in N_r are coincidence detectors, they are selective for homogeneous luminance in their receptive fields [26].

For the neuron in the sheets N_s and N_r we used standard leaky integrate and fire neurons [27, 40]. The neurons in the sending population N_s received no synaptic input, but were stimulated by a current I_{stim}, corresponding to the luminance in the neuron's receptive field. The sub-threshold membrane potential of a neuron in N_s was, thus, given by

$$C \cdot \frac{dU}{dt} = -\frac{1}{\tau} \cdot U + I_{\text{stim}}, \tag{1}$$

where U is the membrane potential relative to the resting potential.

The neurons in population N_r only received synaptic input from a local neighborhood d in N_s with post-synaptic currents modeled as α-functions. Their sub-threshold membrane potential was, thus, given by

$$C \cdot \frac{dU}{dt} = -\frac{1}{\tau} \cdot U + \sum_{\text{syn} \in d} I_{\text{syn}}(t). \tag{2}$$

2.1.2 Stimuli

As stimuli we extracted patches of 100×100 pixels from natural gray-scale images (fig. 6, panel *A*), corresponding to ten by ten degrees in the visual field. The pixel luminances of each stimulus patch were transformed into stimulus currents for the neurons in N_s. Since dark regions will not elicit a neural response, we used two separate input sheets, one responding to high luminance values (ON) and one responding to low luminance (OFF). As a result, the neurons in the sheets N_s-ON and N_s-OFF transformed local luminance into spike latencies.

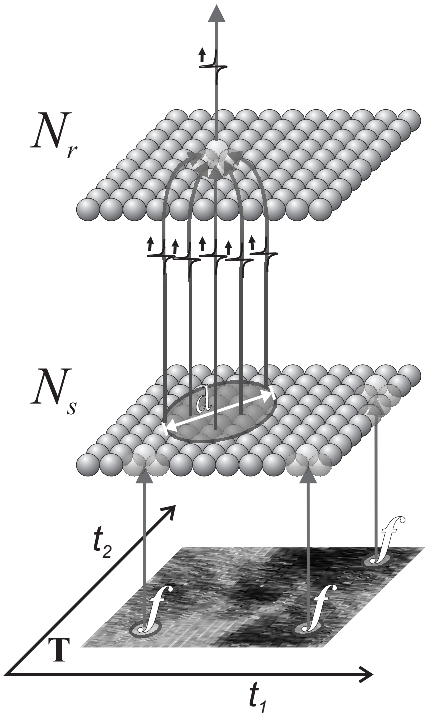


Figure 3: Detecting homogeneous appearance of a feature f (for example, luminance). N_s is a sheet of neurons with topologically arranged receptive fields in the input space $\mathbf{T}$. They encode the local strength of feature f in their spike-latency. Neurons in sheet N_r receive action potentials from a local neighborhood d of neurons in N_s. Their sensitivity for coincident synaptic events makes them selective for homogeneous appearance of feature f inside their receptive fields.

Given a stimulus current I that is strong enough to bring the integrate and fire neuron to threshold, the latency for the first spike is given by [9]:

$$t_{cross} = -\tau_m \, ln \left(\frac{R_m \, I + E_l - V_{th}}{R_m \, I + E_l - V_{start}} \right), \tag{3}$$

with τ_m and R_m the membrane time constant and resistance, and E_l the leakage potential. The latency t_{cross} depends on the membrane potential V_{start} at which the neuron starts integrating the stimulus current. This starting state will naturally vary across a neural population, depending on the individual neurons' stimulation history. Figure 4 shows the current/latency relation for different starting values V_{start}. Small variations in the starting membrane potential cause strong variations in firing latency, especially in the physiologically relevant excitation regime below 600 pA (corresponding to 100 Hz of tonic firing, i.e., a latency of 10 ms, see thick curve). As a consequence, to generate a coherent latency code across the N_s population, the neurons need to be in the same state at stimulus onset. In our model this corresponds to a reset of all membrane potentials to the same initial value.

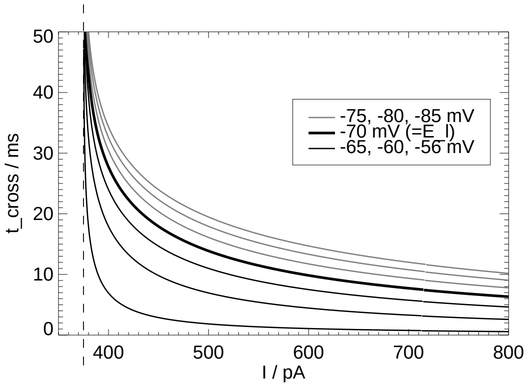


Figure 4: Current/latency relation for injection of a constant current I into a leaky integrate-and-fire model neuron (eq. (3), τ_m=10 ms, E_l=-70 mV, V_{th}=-55 mV, R_m=40 MΩ). Curves show relations for different values of the starting membrane potential V_{start}. For the thick curve, the starting potential is equal to the reset potential of the integrate-and-fire neuron. Latency for this curve corresponds to the spike interval for tonic firing. Thin curves correspond to starting potentials for depolarized states. Grey curves correspond to starting potentials of neurons hyperpolarized by inhibition.

2.1.3 Neural reset

In previous work, we investigated several neural mechanisms that could reset the membrane potentials of a population of neurons to consistent values [26]. The most obvious mechanisms rely on the relaxation of the membrane potential to its resting value, e.g. during the retinal smear created by the moving eye, or blocking the input to the visual cortex at the LGN (saccadic suppression). These can only partly succeed, because a typical saccade is not long enough to allow the membrane potentials to decay sufficiently. Among other conditions, we compared input suppression and common inhibition to an artificial reset of the membrane potentials.

In this model, we reset membrane potentials of all neurons in layer N_s at $t = 100$ ms. We then allow the neurons to fire for 100 ms and record their spike trains. We repeatedly re-initiate latency coding by resetting the neurons after each 100 ms of firing, a period roughly corresponding to a short saccade-interval. Stimulation is constantly applied during the whole procedure. This procedure makes the N_s ON and OFF populations two retinotopically arranged sets of latency-coding neurons selective for local luminance. The N_s neurons re-generate the latency-code after each reset (every 100 ms).

2.1.4 Homogeneity-selective cells

Sub-populations of the neurons in the two N_s layers with a diameter of $d = 1.1^\circ$ of visual angle projected convergently onto two sheets of 100×100 model neurons (N_r, ON and OFF respectively), while preserving topology. Transmission delays and synaptic weights

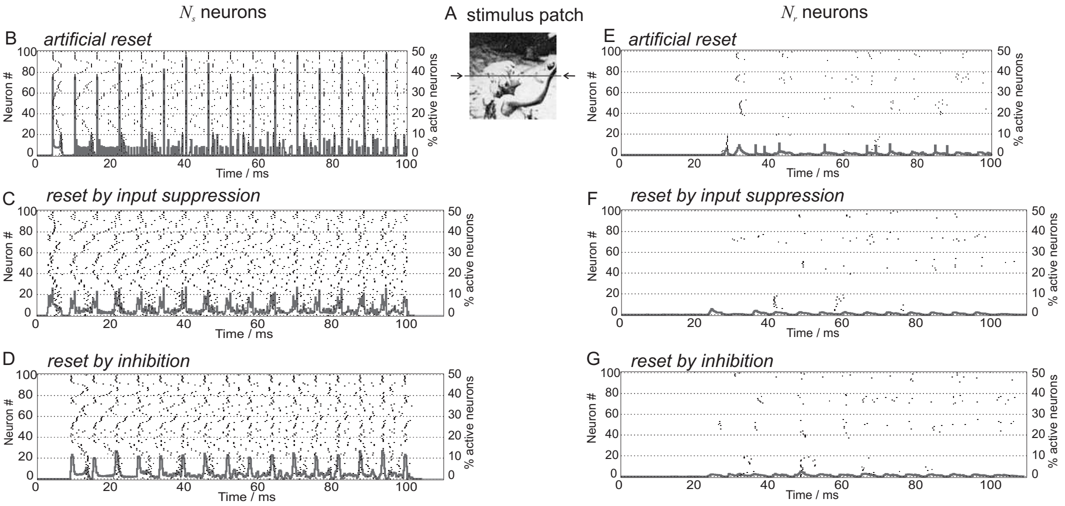


Figure 5: Processing of spatial homogeneity; typical spike responses of 100 N_s and N_r neurons. *A,* stimulus patch, topological RF-positions of the 100 selected neurons correspond to the center row of pixels (marked by arrows); *B and E,* spike responses of N_s and N_r neurons, when N_s neurons were prepared for latency coding using artificial reset; *C and F,* using input suppression; *D and G,* using inhibition; *grey curves,* spike time histograms of all 100 000 model neurons in N_s or N_r.

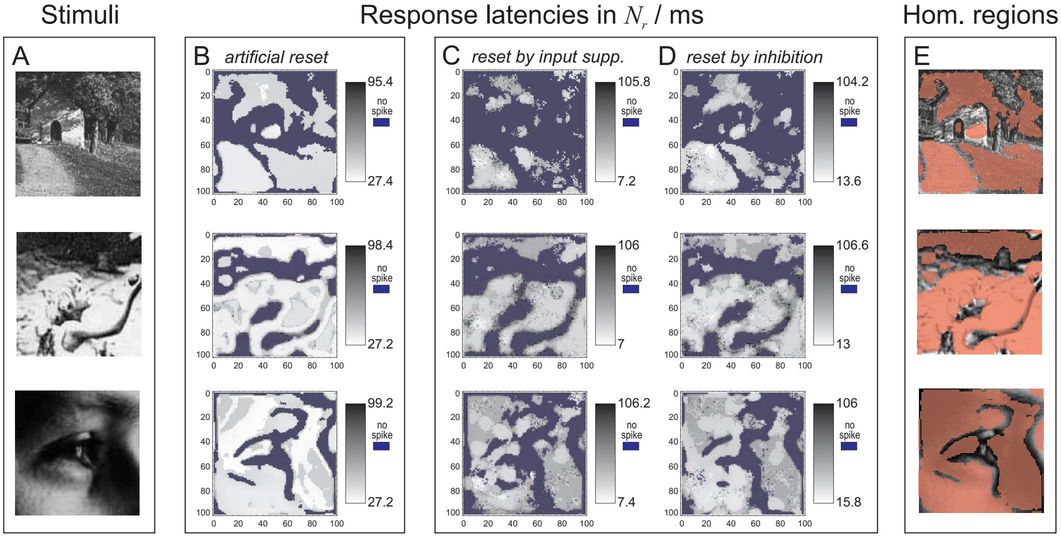
were identical for all connections. This made the N_r ON and OFF populations two retinotopically arranged sets of neurons selective for *spatially homogeneous* luminance across distances of $d = 1.1°$ of angle in the visual field.

2.1.5 Results

We recorded spike responses of all neurons over the complete time-course of simulation. From this data, we retrieved topological latency maps of N_s and N_r neurons. A single simulation run covered multiple resets of the membrane potentials, so that latency codes for the applied stimulus were repeatedly generated. After each reset, the response latency of a neuron is defined as the time span between the reset and the neuron's first spike thereafter.

Figure 5 shows typical spike-trains recorded from N_s and N_r neurons. N_s neurons were excited according to the visual stimulus shown in panel *A*. At $t = 0$, latency coding was initiated by resetting all neurons to consistent membrane potentials by either input suppression, common inhibition, or artificial reset of model neurons. The first two are neurophysiologically plausible methods, while the latter is a process that we use for evaluation, but which is not likely to occur in the real brain [26]. Regions of homogeneous luminance in the stimulus patch cause groups of neighboring N_s neurons to respond with similar latencies. Their action potentials show up as “spike-fronts” in the plots. These coincident spikes, in turn, cause spike responses in the N_r neurons in corresponding locations.

Note to the editor: this figure is available in alternative versions, colored and gray. The colored version is easier to understand.



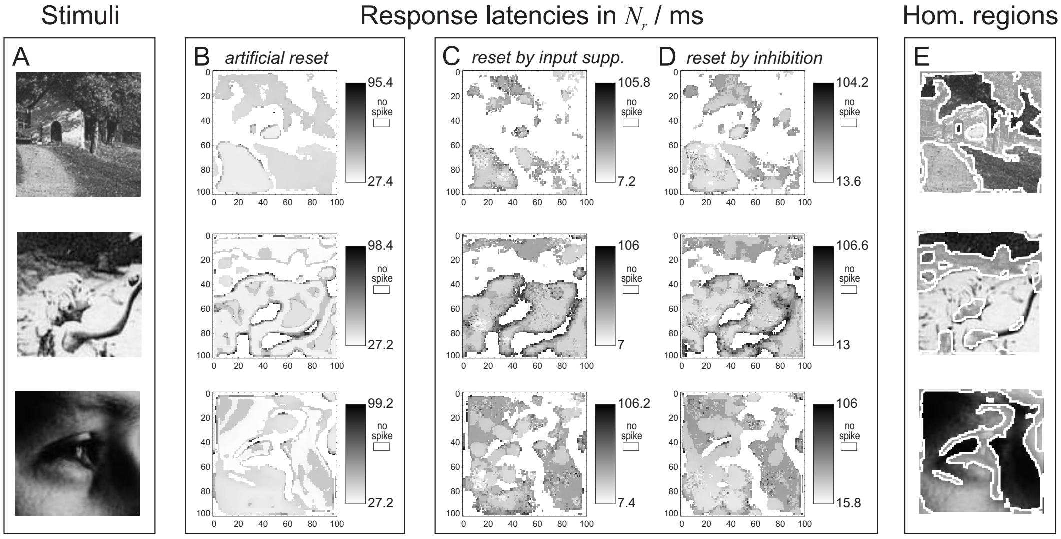


Figure 6: Detecting homogeneity of luminance: results from network simulation. *A*, stimulus patches; *B–D*, typical response latencies in the neural population N_r (white: no action potential was produced). *B*, N_r latencies obtained when the sending neural population N_s was optimally prepared for latency coding by an artificial reset; *C and D*, by neurophysiologically plausible reset mechanisms [input suppression and inhibition, 26]. Panel *E* combines data from panels *A* and *B*: regions of the stimuli where homogeneity-selective neurons responded appear surrounded by white lines, other regions appear reduced in contrast. The plots show ON- and OFF-responses (bright as well as dark homogeneous regions were processed).

Figure 6 shows typical response latencies of N_r neurons on a gray-level scale. (White color indicates that the respective neuron did not produce an action potential.) Panel *A* shows three different stimulus patches. Panel *B* depicts response latencies obtained when the N_s neurons were prepared for latency coding by artificial reset of the model neurons' membrane potentials. Artificial reset produced the most reliable latency code, but is not biologically plausible. Panels *C* and *D* depict response latencies obtained when the N_s neurons were reset by the two biologically plausible mechanisms, input suppression and inhibition. They produced a less reliable latency code, since they can achieve only a partial reset of the neurons' membrane potentials. Consequently, the response latencies scatter more. Still, the locations of active N_r neurons correspond to regions of homogeneous luminance in the stimulus patch — the N_r neurons act as *homogeneity-selective neurons*. Note that by relying on only the first action potentials of latency-coding neurons, spike-latency based homogeneity processing is very fast, with first components signaled already after 5–20 ms.

2.1.6 Discussion

In the initial phase of visual scene interpretation, information provided by surface-selective neurons can help generate a good starting hypothesis on scene contents "at a first glance". Since edges and surfaces of physical objects exclude each other, the output of surface-selective neurons (potentially located in the konio-cellular pathway of the visual system) can be used to suppress responses in orientation-selective cells in regions that are unlikely to be relevant for the rapid, initial understanding of the scene [14].

After a first hypothesis about the content of the visual scene has been established at higher processing levels, a top-down signal can then enable a refined analysis of object detail [see 25, 41]. This would simply involve inhibiting surface-selective neurons of the konio-cellular pathway, thereby dis-inhibiting the previously suppressed orientation-selective neurons in V1. Information on oriented contrast at fine detail will then be channeled into an already pre-adjusted system. This process seems to define a natural time-course of processing from coarse to fine, without unconditionally blurring the stimulus by low-pass-filtering in the way conventional models of resolution pyramids do. By contrast, orientation responses for presumed object boundaries are relayed in full fidelity from the start, and can take part in the formation of a reliable hypothesis.

2.2 Feature extraction from a single spike wave

Homogeneity detection of luminance and other features may indeed be an important step in processing of all sensory cortices. However, it is unlikely that the scene categorization performed by the subjects of Thorpe et al. [38] can be based on the coarse segmentation of image regions this process provides. Categorization of e.g. an animal in a complex scene requires the extraction of highly specific identifying features, possibly including their correct spatial relations. Given that the visual signal needs to traverse something like fourteen or more synapses from the retina to the highest brain areas [13], Thorpe's categorization task is achieved with a processing time of as few as 10 ms per neural stage. This restricts processing to one spike per neuron, and suggests a processing mode that must indeed be similar to the coincidence detection we applied in our model of homogeneity detection.

Thorpe et al. [39] suggest that visual stimuli are categorized by a hierarchy of feature detectors in a feed-forward network, where each feature-selective cell evaluates *the order of spikes* in a single incoming spike-wave. They successfully demonstrated that this coding scheme (*rank coding*) can be used for fast and reliable feature detection [43]. However, to evaluate a rank code, every feature-selective neuron needs a local readout circuit (fig. 7, *left*). This is both biologically implausible and expensive to implement.

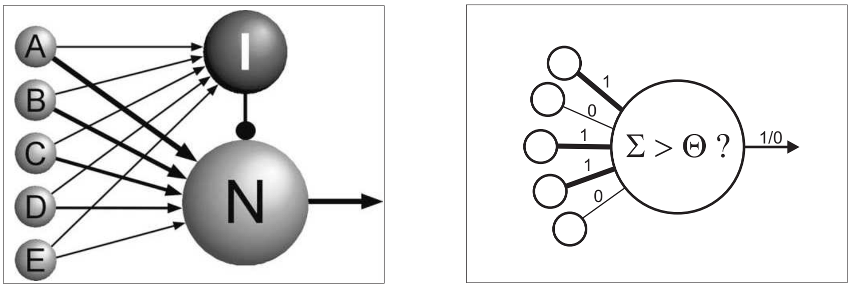


Figure 7: Rank coding versus spike counting. *Left*, neural circuit for evaluating a rank code, as proposed by Thorpe et al. [figure taken from 43]. Each readout neuron requires a dedicated inhibitory pool that increases its firing rate with every incoming spike. *Right*, proposed scheme for spike-counting that can be performed by a single neuron. It is valid when spike intervals are below the synaptic integration time.

We propose an alternative scheme of rapid feature extraction. Post-synaptic neurons could simply count the *number* of incoming spikes in the wave, and fire if the number of incoming spikes exceeds a threshold (fig. 7, *right*). This simple scheme is a valid approximation of neural integration, when spike intervals are below the synaptic integration time (synaptic potentials superimpose constructively). It uses spike-times only implicitly and does not require additional circuitry. We interpret neuronal spikes as a labeled line code, where a spiking neuron indicates the simple presence of a feature. Post-synaptic neurons integrate incoming spikes in an unweighted, “spike-counting” fashion, and fire a spike themselves, when the number of integrated spikes passes a threshold – that is, if enough partial features of “their” compound feature are present. This is an extremely simple neural coding scheme, but biologically easy to implement, cheap, and very fast.

We conducted a simulation study to show if this coding scheme yields neural responses which can serve as the basis for rapid scene or object classification. Besides fulfilling the temporal constraints, two conditions must be met if the individual shall base a behavioral decision on the neural response: The neural response must be specific to a certain stimulus, so that a stimulus can be differentiated from others, and it must be produced reliably when the respective stimulus appears, so that it is not missed.

2.2.1 Model

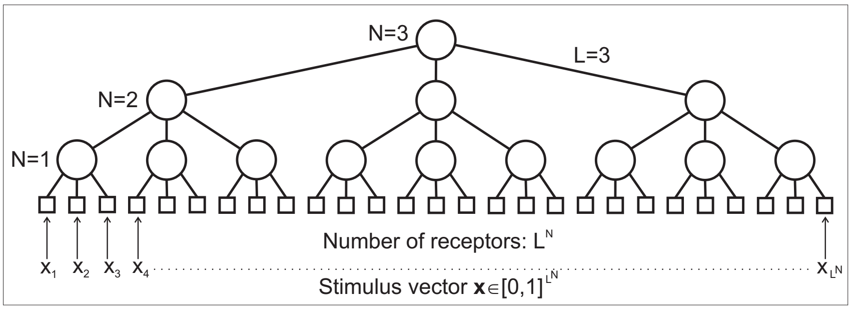


Figure 8: A cascade of spike-counting neurons, with N steps and L incoming lines per neuron.

We evaluate the responses of a cascade of spike-counting feature detectors (fig. 8), with respect to two measures that describe *specificity* and *reliability* of the neural response. The cascade consists of a set of primary receptors (*squares* in fig. 8), and several levels of feature detectors (*circles*). We assume that each of the primary receptors is tuned to some arbitrary stimulus feature and that the tuning curve is smooth and has a peak around a single preferred feature value with a given tuning width. Furthermore, we assume that all neurons are probabilistic with a response probability p. For the primary receptors, the response probability is defined by their tuning curves. For all other detectors, p depends on the number of spikes the detector receives from its input neurons.

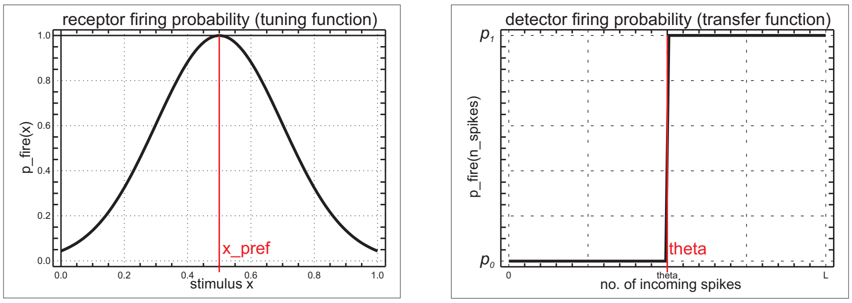


Figure 9: Tuning curves of the primary receptors, and the feature detectors. Neurons produce an output spike according to the shown probability functions. *Left*, primary receptors transform an analog stimulus value into a binary output state (0 or 1). *Right*, feature detectors signal their binary output state according to the sum of incoming spikes.

The primary receptors transform an analog stimulus vector into a vector of spike values. They signal an output spike with a probability that is given by a Gaussian tuning function (fig. 9, *left*). Spikes generated by the primary receptors converge on a set of feature detectors in the next level of the cascade. The feature detectors count spikes and respond with probability p_1, if the number of input spikes exceeds a threshold Θ, and with probability p_0

otherwise (fig. 9, *right*). The spikes converge on the next level of feature detectors, which apply the same scheme, and so on, resulting in a spike wave traveling through the cascade from the receptors to the single top detector. For different sets of stimuli and receptor tunings, we evaluate the firing probability of the top detector in cascades with N steps and L incoming lines per neuron.

In a detector cascade with N steps and L incoming lines per neuron, the number of primary receptors is L^N. We assume that the L^N primary receptors code for L^N arbitrary features of the input, where each feature takes a value in a normalized range of $[0, 1]$. (Think of it as local luminance, edge orientation in the range $[0°, 360°]$, local frequency content, etc.) We choose a set of receptor-preferences

$$P = \{\mathbf{x}_{pref,i}\}, \quad \mathbf{x}_{pref,i} \in [0, 1]^{L^N}, \tag{4}$$

where each vector $\mathbf{x}_{pref,i}$ specifies the preferred feature values for all of the L^N primary receptors, defining the stimulus that maximally excites the detector cascade. We can look upon P as defining a set of physically different neural cascades in the sensory cortex, each starting from a specifically tuned set of primary receptors and ending in a single *grandmother cell* at the top, whose response indicates the presence of the respective compound feature. We choose a set of test-stimuli

$$S = \{\mathbf{x}_i\}, \quad \mathbf{x}_i \in [0, 1]^{L^N}, \tag{5}$$

which we apply as inputs to the primary receptors. For all stimuli $\mathbf{x} \in S$ and all cascades $\mathbf{x}_{pref} \in P$, we experimentally determine the firing probability p_{top} of the top detector, by repeatedly simulating single-spike waves traveling upwards through the cascade.

2.2.2 Results

Figure 10 shows typical firing probabilities of the top detector in cascades with N=2 levels and L=9 incoming lines per neuron. The three panels show simulation runs for three different settings of the firing threshold Θ=4,5,6. Stimuli along the x-axis have been sorted by the response they elicit, with the least effective stimulus on the left, and most effective stimulus on the right. Depending on the value of the firing threshold Θ of the feature detectors in the cascade, the top detector can fire with a high probability for most of the stimuli (*left*), for some of the stimuli (*middle*), or its response probability may be low for all stimuli (*right*). Very little variation can be seen for the different cascades along the y-axis, so response probabilities are characteristic for the coding process and largely independent from the stimulus preferences.

In the following we determine the parameter ranges that yield top detectors that have both a high firing probability and at the same time only a small number of stimuli they respond to (that is, they show *reliable* and *specific* responses). For a fixed cascade $\mathbf{x}_{pref} \in P$, we characterize the response of the top detector to the complete test-set S by two measures, *reliability* and *specificity*.

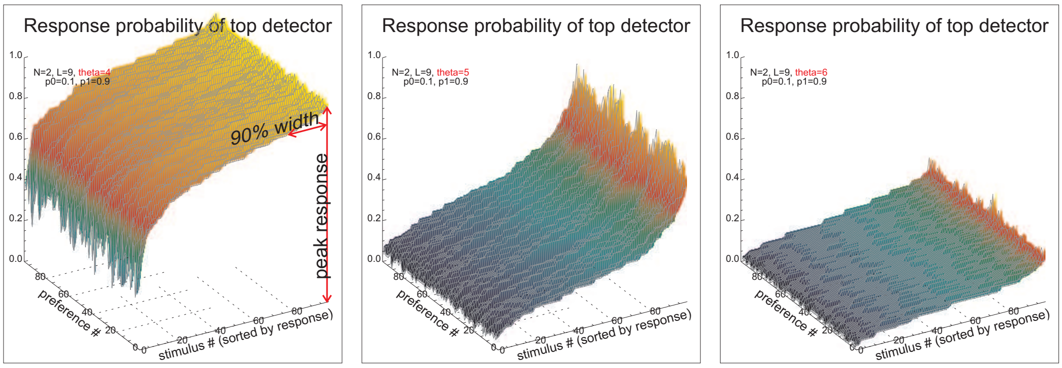


Figure 10: Firing probablilty of the top detectors for all combinations of stimuli and receptor preferences, in cascades with N=2 levels and L=9 incoming lines per neuron. Detector response probabilities (p0,p1)=(0.1,0.9). Panels show firing probability for three different settings of the firing threshold Θ=4,5,6. Stimuli sorted by the response they elicit.

Reliability We define response reliability as the peak response probability of the top detector to all test-stimuli $\mathbf{x} \in S$:

$$rel(\mathbf{x}_{pref}) := \max_{\mathbf{x} \in S} \{p_{top}(\mathbf{x})\}. \tag{6}$$

Specificity We define response specificity as the number of stimuli $\mathbf{x} \in S$ the top detector significantly responds to (90% of peak probability, red arrow in fig. 10, *left*):

$$spec(\mathbf{x}_{pref}) := |\{\mathbf{x} \in S \mid p_{top}(\mathbf{x}) > 0.9 \cdot rel(\mathbf{x}_{pref})\}|. \tag{7}$$

Figure 11 shows the average reliability and specificity for the cascades in P. Values are shown as a function of the threshold Θ, for different heights of the detector cascade N=1,2,3 and a number of L=9 incoming lines per neuron. Plots from left to right show results for increasing noise in the spike generation process in the single detector neurons (p0,p1)=(0,1), (0.1,0.9), (0.2,0.8), (0.3,0.7). The results show that responses are *reliable* for $\Theta < \frac{L}{2}$. They are *specific* to a small number of stimuli only for $\Theta \approx \frac{L}{2}$, and specific responses vanish rapidly with increasing noise (panels from left to right, $p0 \uparrow, p1 \downarrow$).

2.2.3 Discussion

Our study shows that stimulus-specific responses can be gained from a single spike wave traversing a cascade of spike-counting feature detectors. However, in this simplified setup of uniform cascades with N steps and L incoming lines per neuron that we investigated here, stimulus responses that are specific to a small number of stimuli are limited to the parameter regime where the threshold for the spike count is half the number of input synapses ($\Theta \approx \frac{L}{2}$). In this regime, response probability is reduced. We can conclude that the visual system can use this coding scheme for rapid feature detection when responses need not be 100%

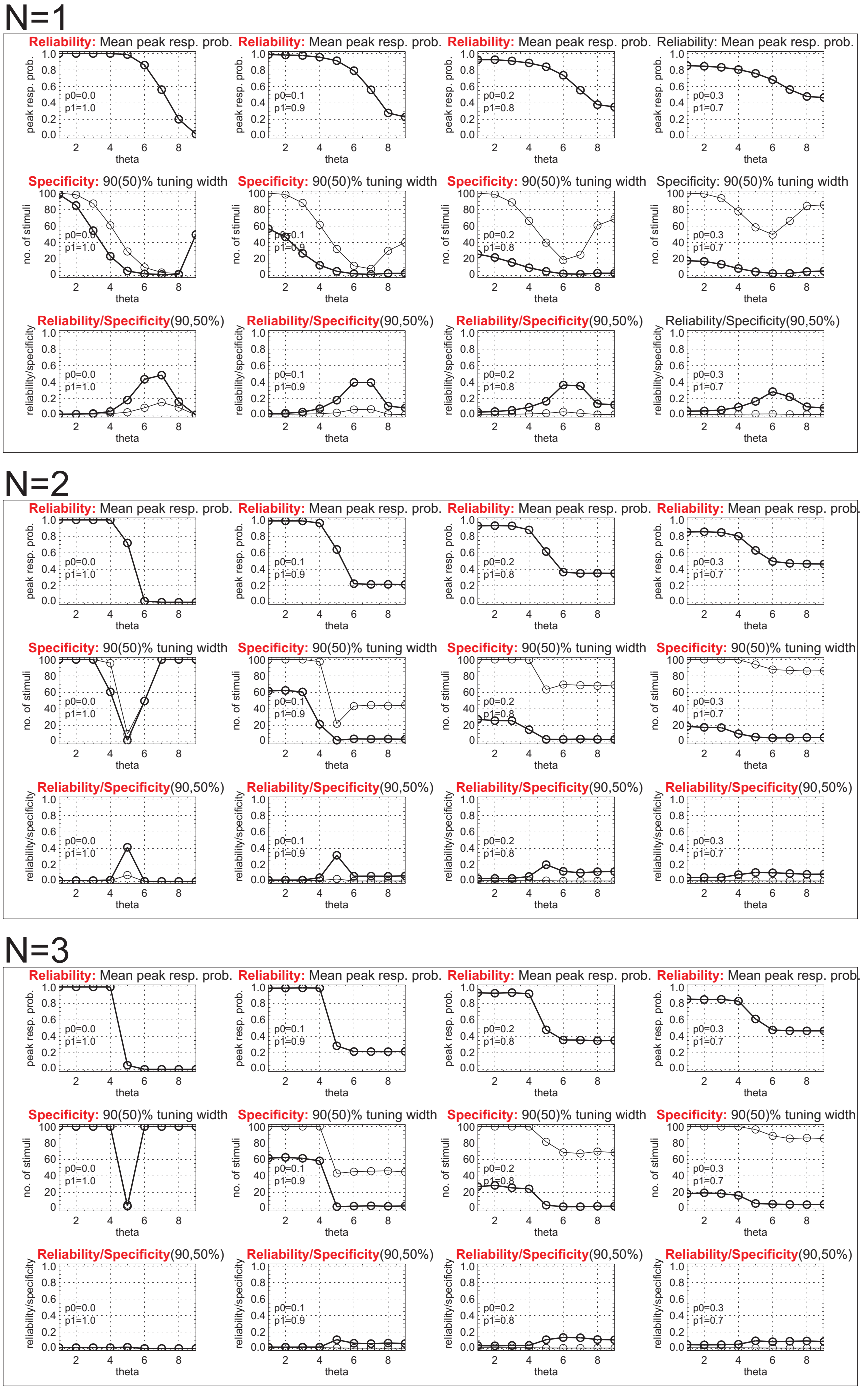


Figure 11: Reliability and specificity of the top detector responses as a function of the threshold Θ, in cascades with N=1,2,3 levels and L=9 incoming lines per neuron. *Plots from left to right:* Increasing noise in spike generation (p0,p1)=(0,1), (0.1,0.9), (0.2,0.8), (0.3,0.7). *Top-rows*, reliability; *middle rows*, specificity (90% and 50% peak response width); *bottom rows*, reliability/specificity.

reliable, for example when they can be repeatedly generated, or when the same feature is extracted by several cascades in parallel. It is well conceivable that several spike waves travel the cascade in close succession, so that several samples of the top response can be taken in a short time.

The narrow range of usable thresholds suggests homeostatic processes that keep the neurons near the working point. In addition, the rapid decay of response specificity with increasing noise suggests coding strategies that counteract the influence of noise. Both are not present in the simple setup we used here. However, we have shown that specific and reliable responses for arbitrary features can be gained using an extremely fast and simple coding scheme, which does not require an additional inhibitory pool of neurons for each detector as does the rank coding scheme proposed by Thorpe and colleagues. Moreover, feature detection by simple spike-counting does not depend on the exact spike latencies, as long as the synaptic potentials caused in the receiving neurons overlap sufficiently to add up to a supra-threshold potential. This leaves the "latency channel" as a separate coding channel that is multiplexed with the pure detection of a feature. It is straightforward that a very strong feature present in the stimulus will cause faster responses in the primary receptors, meaning that also the top detector of the respective cascade will respond more early than the one for another feature that is present, too, but weaker. This creates a sort of *endogenous attention* to the strongest stimulus components, because rapid decisions of the individual will be based on the first responses available.

This is an ideal condition for generating a rapid hypothesis on the stimulus content that can be used to perform scene categorization tasks like shown in Thorpe et al. [38], which are the basis for behavioral decisions like for example, fleeing. In addition, a rapid hypothesis can be used to improve and guide further cognitive processing, when the individual has the time. Spike-counting can be reasonably correct to base rapid actions on it and serve as a hypothesis. For exact and detailed object recognition with low error rates it is certainly not sufficient. Here, slower mechanisms must be at work, that integrate evidence over time, leading to increasingly complex hypotheses, expectations and knowledge. Modeling this process of dynamic scene interpretation is a second field of study at the Honda Research Institute.

3 From the hypothesis to predictive stimulus recognition

Recognizing and categorizing previously experienced objects and scenes as well as learning new representations is still one of the hardest problems in artificial intelligence and subdisciplines such as computer vision or robotics. Because sensory stimuli regularly include complex and ambiguous scenes, it is necessary to use top-down prediction in order to reduce the tremendous number of possible interpretations at each level of representation. Thus, any solution has to detail (i) how prior knowledge is to be integrated (top-down) with the actual (bottom-up) stream of sensory data in a meaningful way, and (ii) when and how new representations are to be created and stably integrated into the previously learned knowledge hierarchy. Because biological organisms are currently the only systems capable of solving these tasks to a satisfactory degree, we want to use the growing knowledge about the anatomy and physiology of the brain to incorporate this knowledge in biologically inspired models.

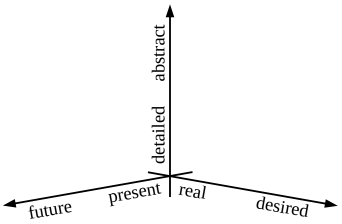


Figure 12: Cortical processes must be able to distinguish between different representational modi. E.g., representational states may refer to present, future, reality, wish, signal, or symbol. We have the idea that, at least for higher mammals, a single cortical column can represent several modi at the same time, and that different modi relate to different cortical layers.

3.1 Stimulus representations in the six layers of cortical columns

Although it is well known for a long time that neocortical anatomy exhibits a 6-layered structure, modelers have often neglected this fact when modeling a cortical patch by a single "monolithical" neuron population (e.g., [22, 32, 35]). This may be attributable to the wish to focus on a single layer or the lack of adequate computational resources to simulate more detailed models, but also to doubting or underestimating the functional significance of discrete within- or between-layer synaptic connections which appear to have a rather "fuzzy" character [1, 5].

In accordance with ideas developed earlier in Körner et al. [25] [cf. 15, 33], we assume that the *basic function of a cortical column* is to adequately represent and predict (or generate) its sensory inputs. To achieve this in a self-organizing, autonomous way it is necessary to have access to different representational modi such as actual vs. predicted sensory input, where we believe that different representational modi of the same entity (e.g., orientation at a particular position in the visual field) are located in different layers within the same column rather than monolithically in different columns or areas (see fig. 12).

How can such a *generative model* look like? We can assume that the model represents external states that produce the observed sensory inputs. Thus at each time the model must represent a state v from the state space V (or more generally, a probability distribution on the state space describing in which state the columnar system "believes" to be in). Then the system should be able to use sensory input s to update the state v according to a function f,

$$v(t + \Delta t) \quad = \quad f(v(t), s(t)) \tag{8}$$

It makes sense to divide the state variable $v = (w, a)$ into two rather independent entities, one variable w describing "external" entities from the outside world, and another variable

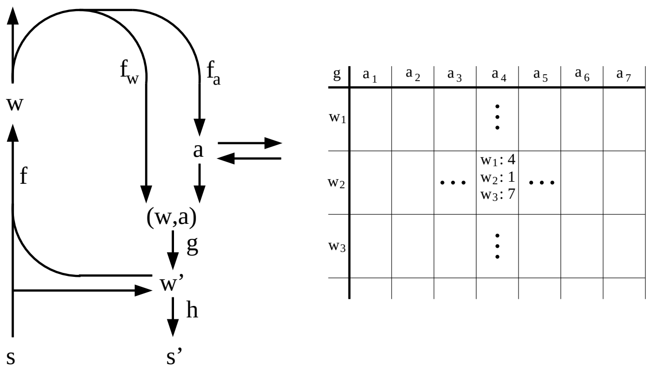


Figure 13: *Left:* Basic functional circuit of a cortical column. Sensory input s is used to update the current world state w. This is used to choose an appropriate action a. World state and action can be used to predict the next world state w' and next sensory input s'. *Right:* A simple histogram representation of the conditional probability density $g(w, a)$ on (discrete) states and actions: The example shows the density of the predicted "world" states w' when being in state w_2 and performing action a_4. Learning is accomplished by simply incrementing bin k of the histogram in row i, column j whenever evidence $(w_i, a_j) \to w_k$ is experienced. Our model essentially implements such a histogram representation where the states or state combinations (w_i, a_j) are "coded" with distributed cell assemblies in order to relieve combinatorial problems and reduce the number of required neuronal units [4, 16, 22, 31, 46].

a describing a local “internal actor”. In addition to updating a state, the system should also be able to predict a future state w' and expected sensory inputs s', so that (see fig. 13, *left* for the functional circuit):

$$\begin{aligned} w(t+\Delta t) &= f(w'(t), s(t)) && (9) \\ a(t) &= f_a(w(t), \ldots) && (10) \\ w'(t) &= g(w(t), a(t)) && (11) \\ s'(t) &= h(w'(t)) && (12) \end{aligned}$$

We will refer to f as the “forward recognition function”, to g as the “predictive function”, and h as the “backward function”.

How can the recognition, prediction, and backward functions be learned? In general, updating the model will involve two phases: (i) finding the most probable w, (ii) given that the world and actor states (w, a) are fixed, the functions can simply be learned by counting up experienced evidence. For example, the prediction function g is essentially a conditional probability (“the probability to get to state w' given state w and action a”) which, for discrete states and actions, can be represented by evidence histograms for each combination (w, a) (fig. 13, *right*).

By comparison with known anatomical facts we can match our functional model (fig. 13) with the layered organization of neocortex [5, 11, 25]. We believe that the forward recognition function f is located in the middle and upper layers, while the remaining functionality related to behavior and predictions is located in the lower layers. Furthermore, we believe that f is split up into two subsystems, one for fast bottom-up recognition in the way we have modeled in section 2.2, and another for refined recognition employing feedback.

In the following we present simulation results substantiating a previously proposed model of computation in neocortical architecture [25]. This model gives a detailed functional interpretation of the well-known six-layered columnar cortical architecture and related sub-cortical (thalamic) structures. It hypothesizes three different but interfering processing systems at each stage of the cortical hierarchy (fig. 14): The *A-system* (comprising the middle cortical layers 4 and lower 3) accomplishes fast bottom-up processing where the first spike wave traveling up the cortical hierarchy can activate a coarse initial hypothesis at each level. We have modeled this process on the single-neuron level in section 2.2; here, we link it to an explicit neural resource, the layers 4 and lower 3 of the cortex. In the *B-system* (superficial layers 2 and upper 3) the initial hypothesis is refined by slower iterative processes involving horizontal and vertical exchange of information. Finally, the *C-system* (deep layers 5 and 6) represents the local hypothesis of a hypercolumn which is used for inducing expectations and predictions for the present and future input signals, and also for inducing the behavior that can confirm these expectations. Recognized or predicted input signals are suppressed at an early cortical stage, and only differences between predicted and actual signals can reach the next higher level. Learning of new representations is induced if the difference signal is too large and if the difference signal reaches the highest level of cortical integration, the hippocampus.

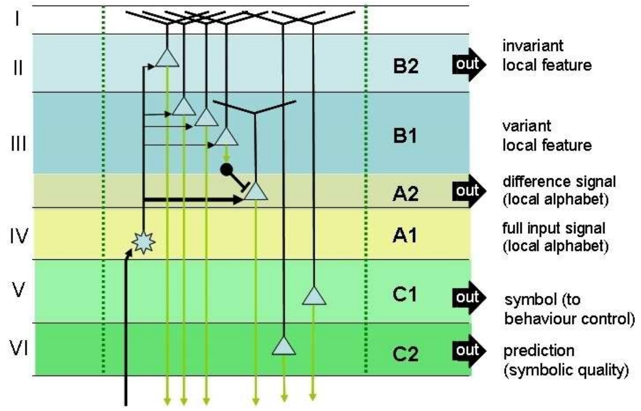

I
II
III
IV
V
VI
B2
B1
A2
A1
C1
C2
out
invariant local feature
variant local feature
out
difference signal (local alphabet)
full input signal (local alphabet)
out
symbol (to behaviour control)
out
prediction (symbolic quality)

Figure 14: Layered model of a cortical column as proposed previously in [25]. Three different processing subsystems corresponding to different vertical locations in the cortical tissue interact within each cortical column. The A-system (middle layers) accomplishes fast bottom-up processing of sensory signals as shown in sec. 2.2, the B-system (superficial layers) refines the input from the A-system by exchanging information with neighboring columns and activating more sparse and abstract representations. The C-system (deep layers) develops representations related to behavior that are then used to confirm predictions in an action/perception cycle.

In this section, we present a simple (but instructive) simulation of stimulus recognition consisting of three cortical levels representing letters, syllables, and words. Focusing on the information flow between columns and areas, we show how the different processing systems interact in order to quench out expected signals and accomplish symbolic recognition of words, and how representations for new words can be constructed based on old representations. This simulation operates on the level of neural circuits, the main modeling units are the layers of a cortical area. The implementation applies a rate-code, although the processes in the A-system build upon the rapid spike-based hypothesis generation we have described before. The stimuli processed by the model are static, although their interpretation is dynamic.

3.2 Object recognition by bidirectional signal flow between cortical columns

Here we describe the COREtext model, a layered cortical model that demonstrates the formation of a fast initial stimulus hypothesis in the columns, and its subsequent refinement by inter-columnar communication. We simulate the signal flow in the A- and B-systems of our cortical model across three hierarchical cortical areas.

3.2.1 Model

The COREtext model consists of three areas, “IT”, “V2”, and “V1”. Each area is composed of three layers, which are linear arrays of h neural subsystems (fig. 15). A neuronal subsystem is an aggregation of m neurons. The number m is referred to as *the number of minicolumns* in the area. The number h is referred to as *the number of hypercolumns* in the area. The COREtext model uses a graded-response neuron model. In each timestep, a neuron accumulates its stimulation s and its modulation m. The resulting activation a is given by $a = \Theta(s \cdot (1 + m))$. Here, Θ is a piecewise linear sigmoidal transfer function. Θ is zero below a firing threshold of 0.3, and rises linearly up to a maximal activation value of $a = 3$, at which it saturates. $\Theta(1) = 1$. The neurons in each subsystem participate in a “winners-take-all” competition, and all but the maximal activations in a subsystem are set to zero.

Synaptic connections between the neurons in the COREtext model are specified separately between pairs of layers. Weights between the neurons of given source and target layers take binary values, multiplied by a constant factor w: $(w_{ijkl}) = w \cdot (b_{ijkl}),\ b_{ijkl} \in \{0, 1\}$, with i: source subsystem, j: source neuron in subsystem, k: target subsystem, and l: target neuron in subsystem. When subsystem and neuron indexes are not of explicit interest, we can shorten this notation to $(w_{ij}) = w \cdot (b_{ij})$. A set of synaptic connections can either be driving or modulating. Inhibitory connections are realized by negative driving weights.

The COREtext model uses a synchronous update of neuronal states. In each simulation time-step, the activity of all neurons is propagated to the postsynaptic neurons, according to the connection matrix. From the resulting activation values, the “winners-take-all” competition is computed in the neural subsystems. The result is stored as the new activation of the neurons. This update cycle repeats in every simulation time-step.

For conceptual simplicity, we chose an abstracted stimulus environment that shows a clear hierarchical structure of how complex objects are composed from smaller parts. We implement word recognition from a string of characters. Stimuli have the form of strings of lower-case characters, arranged on a one-dimensional grid. In our setup of the COREtext model, all layers have the same number h of subsystems (hypercolumns), and are aligned with the input array (fig. 15). We choose a number of three COREtext areas that form a hierarchy, similar to sensory processing pathways in the neocortex. The areas are labeled “V1”,”V2” and “IT” for convenience, where these labels are metaphors taken from the domain of visual processing. Figure 16 (top left) relates features processed in the visual domain to the features processed in the COREtext model. We derived the set of features represented in the three areas from the analysis of three pages of classical German literature [17]. We decomposed the text into syllables according to usual German spelling rules. After removal of duplicates the text analysis yielded a set of 386 different words, a set of 418 syllables, and a set of 30 characters. We call these sets of features the *local alphabets* $L_{\mathrm{IT}}, L_{\mathrm{V2}}$, and L_{V1}. The sizes of the local alphabets determine the number m of minicolumns in the three areas. Each hypercolumn in “IT”, ”V2”, and “V1” represents the full local alphabet. I.e., each subsystem in “V2” is an aggregation of $m_{\mathrm{V2}} = 418$ neurons, with each neuron $n_{l,s,i}$ representing the i^{th} element of L_{V2}.

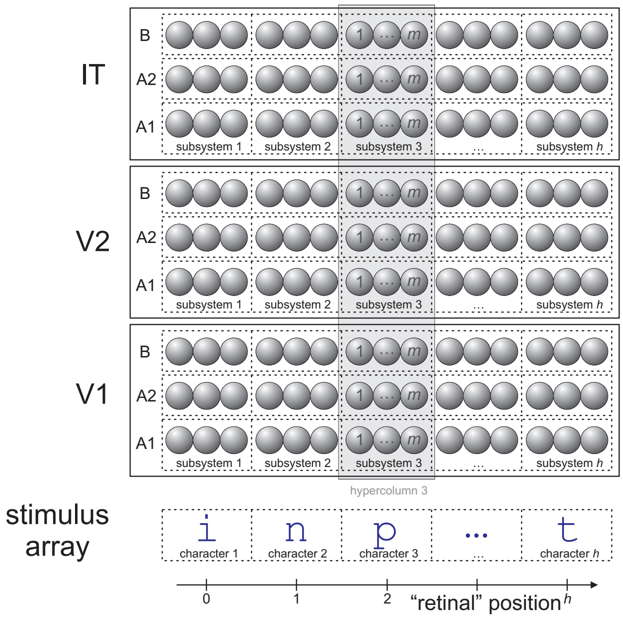


Figure 15: Hierarchy of areas, layers and subsystems. The subsystems are ”retinotopically” aligned with the stimulus array.

Knowledge about the composition of cognitive objects (words) is generated from German spelling rules. From the local alphabets we derive two binary knowledge matrices, $\mathbf{B_1}$ and $\mathbf{B_2}$. We initialize the knowledge matrices in a process that imitates supervised one-shot-learning. It is similar to training all words in L_{IT} by presenting them at position 0 in the input array, while giving their correct decompositions into syllables and characters. For each word in L_{IT} we derive its constituting syllables and characters, as well as their starting position in the word. We store these decompositions in the binary knowledge matrices $\mathbf{B_1}$ and $\mathbf{B_2}$. For example, the word “mutter” decomposes in the way shown in fig. 16 (right).

We designed the connectivity scheme between the nine COREtext layers to reflect the main functional projections that are known to exist between the layers of cortical areas [3, 6, 11, 37]. We aimed to create a reduced model of the functional connectivity that is at the core of the cortical columnar processing. The connectivity scheme is shown in fig. 16, left panel. Layers of the same COREtext area are linked by 1:1-connections (single arrows). These are connections between neurons in the same minicolumn. (E.g., a neuron representing the syllable “mut” in V2-A1 projects to the neuron representing the *same* syllable and at the *same* position in V2-A2.) Stimulation enters the system at layer V1-A1. This layer is linked to the input array (fig. 15) by 1:1 connections.

Layers of different COREtext areas are linked by connection matrices proportional to the two knowledge matrices $\mathbf{B_1}$ and $\mathbf{B_2}$ (double arrows). These are connections that span mini- and hypercolumns, and that implement the knowledge about the hierarchical composition of cognitive objects (words) in the network. Backward arrows ending in circles

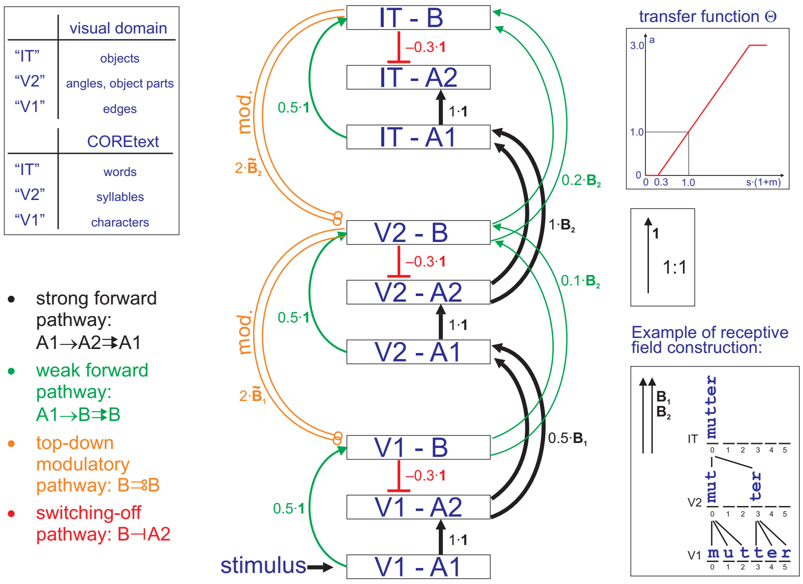


Figure 16: Left: Connectivity scheme between the COREtext layers. *Bottom right:* Example for the construction of a cognitive object (word) from its constituents (syllables and letters) in the local alphabets.

indicate modulatory connections. The modulatory backward connections between the B-systems are reverse to the forward connections. I.e., a neuron in IT-B projects back to exactly the same neurons in V2-B that it receives from. The matrices $\tilde{\mathbf{B}}_1$ and $\tilde{\mathbf{B}}_2$ denote these reverse connections. The exact connection values including scaling factors are shown beside the arrows in fig. 16.

The connection scheme can be categorized into several pathways. The strong A1→A2⇒A1 forward pathway reflects the strong driving synapses that are found between layers 4 and 3 of cortical areas. The weaker A1→B⇒B pathway and the top-down modulatory B⊸B pathway reflect the reciprocal connections between the neurons in upper layers 2 and 3 of cortical areas. The inhibitory B⊣A2 pathway reflects local inhibition via interneurons in upper and middle layers of a column.

3.2.2 Results

Response modes Simulation runs consisted of a constant stimulation by a string of lowercase characters in the input array, and subsequent iteration of the update cycle. The signal flow in the COREtext system, based on the distinct connectivity and the knowledge imprinted in the synaptic connections, takes different response modes.

Figure 17 shows a stimulation example using a string of characters that is identical to a known word. The word "mutter" was part of the analyzed text, and knowledge about its decomposition into syllables (*mut*, *ter*) and characters is imprinted into the connections between the COREtext areas. The stimulus activates the corresponding single-character-detectors at each hypercolumn in “V1”. Form here, a fast wave of activation spreads via the A1→A2⇒A1 path and the A1→B⇒B path (cf. fig. 16). Syllable-detectors in “V2” and word-detectors in “IT” get active according to the forward connections matrices. Starting from these first activations in the B-systems, activity propagates backwards via two pathways. Along the modulatory top-down B⊸B pathway, an active word detector in “IT” supports activations in all detectors in “V2” that are compatible with the word’s decomposition into syllables. Similarly, each active syllable detector supports activations in all compatible detectors in “V1”. At the same time, active neurons in the B-systems inhibit neurons in the A2-systems in the same minicolumn of the same area. (E.g., an active detector for syllable *mut* in V2-B inhibits the detector for the same syllable *mut* and at the same position in V2-A2.)

```
t=10
-----
IT-A1:  0:                      IT-A2:  0:                      IT-B :  0: mut-ter( 1.3) göt-t
IT-A1:  1:                      IT-A2:  1:                      IT-B :  1:
IT-A1:  2:                      IT-A2:  2:                      IT-B :  2:
IT-A1:  3:                      IT-A2:  3:                      IT-B :  3:
IT-A1:  4:                      IT-A2:  4:                      IT-B :  4:
IT-A1:  5:                      IT-A2:  5:                      IT-B :  5:

V2-A1:  0:                      V2-A2:  0:                      V2-B :  0: mut( 3.0) göt( 2.2)
V2-A1:  1:                      V2-A2:  1:                      V2-B :  1:
V2-A1:  2:                      V2-A2:  2:                      V2-B :  2: cher( 0.4) sten( 0.
V2-A1:  3:                      V2-A2:  3:                      V2-B :  3: te( 3.0) tern( 3.0)
V2-A1:  4:                      V2-A2:  4:                      V2-B :  4: er( 0.4)
V2-A1:  5:                      V2-A2:  5:                      V2-B :  5:

V1-A1:  0: m( 1.0)              V1-A2:  0:                      V1-B :  0: m( 3.0)
V1-A1:  1: u( 1.0)              V1-A2:  1:                      V1-B :  1: u( 3.0)
V1-A1:  2: t( 1.0)              V1-A2:  2:                      V1-B :  2: t( 3.0)
V1-A1:  3: t( 1.0)              V1-A2:  3:                      V1-B :  3: t( 3.0)
V1-A1:  4: e( 1.0)              V1-A2:  4:                      V1-B :  4: e( 3.0)
V1-A1:  5: r( 1.0)              V1-A2:  5:                      V1-B :  5: r( 3.0)
```

Figure 17: Stimulation example using a known word “mutter”. Each area consists of 6 hypercolumns (0–5). Labels left to the colon indicate area, layer, and hypercolumn. After the colon, active symbols in the respective subsystem are listed. Neural activations are given in parentheses behind the corresponding character, syllable or word from the local alphabet. Activations of 0 are not shown.

The state shown in fig. 17 is the static pattern of activations in response to the stimulus “mutter”, that is reached after 10 update cycles. In IT-B, the correct symbol *mutter* is active. In V2-B and V1-B compatible constituting syllables and characters are active. The activations in the B-systems support each other via the forward B⇒B and backward B⊸B pathways. At the same time, via the inhibitory B⊣A2 pathway, they “switch off” the according signals in the A2-systems of the same minicolumn. After 10 update cycles, activations in the A2-systems have completely vanished, and activations in the B-systems are mutually supportive.

Figure 18 shows a stimulation example using a string of characters that is similar, but not identical to a known word. The stimulus “vatxrx” is a distorted version of the words “vater” and “vaters” that were part of the analyzed text. The state shown in fig. 18 is the static pattern of activations reached after 8 update cycles. The bottom-up/top-down dynamics activates the symbols *vater* and *vaters* in IT-B, showing that the system was able to compensate for the distortions. In V2-B and V1-B compatible constituting syllables and characters are active. The activations in the B-systems support each other via the forward B⇒B and backward B—◦B pathways. Note that the two distorted characters *x* in V1-B are only weakly active, since they *do not* receive modulatory support from any syllable in V2-B. This means, they can exert only weak inhibition on their V1-A2 counterparts via the inhibitory B⊣A2 pathway. After 8 update cycles, all activations in the A2-systems have vanished, except for the two distorted characters in V1-A2.

```
t=8
-----
IT-A1:  0:                       IT-A2:  0:                       IT-B :  0: va-ter(1.5) va-ters
IT-A1:  1:                       IT-A2:  1:                       IT-B :  1:
IT-A1:  2:                       IT-A2:  2:                       IT-B :  2:
IT-A1:  3:                       IT-A2:  3:                       IT-B :  3:
IT-A1:  4:                       IT-A2:  4:                       IT-B :  4:
IT-A1:  5:                       IT-A2:  5:                       IT-B :  5:

V2-A1:  0:                       V2-A2:  0:                       V2-B :  0: va(3.0) hat(3.0) ta
V2-A1:  1:                       V2-A2:  1:                       V2-B :  1:
V2-A1:  2:                       V2-A2:  2:                       V2-B :  2: ter(3.0) ters(3.0)
V2-A1:  3:                       V2-A2:  3:                       V2-B :  3:
V2-A1:  4:                       V2-A2:  4:                       V2-B :  4:
V2-A1:  5:                       V2-A2:  5:                       V2-B :  5:

V1-A1:  0: v(1.0)                V1-A2:  0:                       V1-B :  0: v(3.0)
V1-A1:  1: a(1.0)                V1-A2:  1:                       V1-B :  1: a(3.0)
V1-A1:  2: t(1.0)                V1-A2:  2:                       V1-B :  2: t(3.0)
V1-A1:  3: x(1.0)                V1-A2:  3: x(0.9)                V1-B :  3: x(0.3)
V1-A1:  4: r(1.0)                V1-A2:  4:                       V1-B :  4: r(3.0)
V1-A1:  5: x(1.0)                V1-A2:  5: x(0.9)                V1-B :  5: x(0.3)
```

Figure 18: Stimulation example using a distorted word “vatxrx”. The best matching symbols have been found in IT-B, and the two distorted characters remain active in V1-A2, indicating parts of the stimulus that could not be explained from knowledge.

Figure 19 shows a stimulation example using the stimulus “rüdiger” that is largely dissimilar to all words that were part of the analyzed text. A variety of symbols in “IT” and “V2” get weakly active, and via the top-down modulatory B—◦B pathway support all compatible syllables and characters in V2-B and V1-B. At the same time, they inhibit the according signals in the A2-systems of the same minicolumn. This changed distribution of activations in the A2 and B systems changes the activations in the B-systems via the forward pathways. The mutual dependence of the two systems leads to a constantly changing pattern of weak activations. The state shown in fig. 19 is the pattern of activations reached after 15 update cycles. This state is not stable, but activations in the A- and B-systems of all areas keep changing between reappearing patterns. As a consequence, the response to the unknown stimulus “rüdiger” is not a stable active symbol in “IT”, but a whole set of alternating symbols.

```
t=15
-----
IT-A1:  0: red-ner(0.5) rühr-t  IT-A2:  0: red-ner(0.4) rühr-t  IT-B :  0: red-ner(0.1) rühr-t
IT-A1:  1:                      IT-A2:  1:                      IT-B :  1:
IT-A1:  2:                      IT-A2:  2:                      IT-B :  2:
IT-A1:  3:                      IT-A2:  3:                      IT-B :  3:
IT-A1:  4:                      IT-A2:  4:                      IT-B :  4:
IT-A1:  5:                      IT-A2:  5:                      IT-B :  5:
IT-A1:  6:                      IT-A2:  6:                      IT-B :  6:

V2-A1:  0: red(0.8) rühr(0.8)   V2-A2:  0: red(0.5) rühr(0.5)   V2-B :  0: red(0.4) rühr(0.4)
V2-A1:  1:                      V2-A2:  1:                      V2-B :  1:
V2-A1:  2: hig(0.4) nigs(0.4)   V2-A2:  2: hig(0.8) nigs(0.8)   V2-B :  2: hig(0.3) nigs(0.3)
V2-A1:  3: cher(0.6)            V2-A2:  3: cher(0.8)            V2-B :  3: cher(0.4)
V2-A1:  4: der(0.6) ler(0.6)    V2-A2:  4: der(0.8) ler(0.8) g  V2-B :  4: der(0.4) ler(0.4)
V2-A1:  5: brin(0.2) trä(0.2)   V2-A2:  5:                      V2-B :  5:
V2-A1:  6: ri(0.2)              V2-A2:  6:                      V2-B :  6:

V1-A1:  0: r(1.0)               V1-A2:  0: r(0.5)               V1-B :  0: r(1.1)
V1-A1:  1: ü(1.0)               V1-A2:  1: ü(0.7)               V1-B :  1: ü(0.7)
V1-A1:  2: d(1.0)               V1-A2:  2: d(0.7)               V1-B :  2: d(0.7)
V1-A1:  3: i(1.0)               V1-A2:  3: i(0.8)               V1-B :  3: i(0.3)
V1-A1:  4: g(1.0)               V1-A2:  4: g(0.8)               V1-B :  4: g(0.3)
V1-A1:  5: e(1.0)               V1-A2:  5: e(0.7)               V1-B :  5: e(0.3)
V1-A1:  6: r(1.0)               V1-A2:  6: r(0.7)               V1-B :  6: r(0.3)
```

Figure 19: Stimulation example using an unknown word “rüdiger”. Activations in the A- and B-systems of all areas keep oscillating through a set of alternatives.

Randomized update In order to rule out that the third response mode (iteration of a set of symbols) is an epiphenomenon caused by the synchronous update of all model neurons, we implemented a randomized update scheme. In this scheme, only a fixed fraction of randomly chosen neurons is updated in a time step, while all other neurons keep their activations. We could confirm, that the phenomenon reproduces independently of the fraction of updated neurons. Area “IT” iterates the same set of symbols in all cases. Typical set sizes are 2–10, depending on the input string. The size of the iterated set, and thus the frequency of occurrence of the individual symbols, depends on the number of known symbols that overlap with the input string. (For example, the iterated words tend to start with the same letter as the input string, making the iterated sets larger for frequent German starting letters.)

The large invariance with respect to the fraction of updated neurons shows that symbol iteration in response to unknown stimuli is a robust propagation phenomenon in our network. It is rooted in the mutual dependence of feed-forward excitation and inhibitory feedback, which cannot be congruent for an unknown stimulus: The winning symbols in the B-systems support the inhibition of their constituting parts via the top-down B⊸B⊣A2 pathway (fig. 16). However, an unknown stimulus activates a set of parts that is different from the constituting parts of any one known symbol at the next hierarchic level (otherwise, it would be known). The remaining, non-inhibited parts will consequently cause another symbol to win the competition at the next hierarchic level, causing an iteration of symbols that overlap with the stimulus.

3.2.3 Discussion

After a stimulus is applied, a fast wave of bottom-up activation spreads via the forward pathways. Symbols in the upper cortical areas get active after only a few monosynaptic propagations of activity. In this part of the activity spread, the whole system acts as a multilayered perceptron. This fast forward activation of a first hypothesis about the stimulus content is compatible with findings on the speed of processing of categoric information in the human visual system [38]. After the formation of a fast initial hypothesis, it is consolidated with the evident stimulation via the inter-areal top-down modulatory pathway. All stimulus parts that could be confirmed get “switched-off” in the A2 systems (middle cortical layers), indicating that the active symbols in upper cortical areas correctly predict these parts of the stimulus [33]. Finally, the B-systems maintain a self-consistent explanation of the stimulus from “pure knowledge”.

In the case of a stimulus with variations or distortions (second example), a self-consistent explanation can also be established. The “switching-off” of activity in the A2-systems must, however, leave residuals, since parts of the stimulus establish a bottom-up evidence that cannot be confirmed by top-down consolidation. The B-systems represent an abstracted or corrected version of the stimulus, as would be expected from “pure knowledge”. Still, information on the unexpected details is not lost: Residuals in the A2-system clearly identify the parts of the stimulus that cannot be explained from knowledge. This residual activity in the A2-system can be used in several ways. First of all, its pure existence is an indication, that the recognized symbols do not entirely represent the stimulus. Second, the residual activity is specific in the position (the hypercolumn) it appears in. It indicates the exact position of the unexplained parts in the stimulus, and can thus guide a motor action, e.g., a saccade, to gather additional information on the yet unexplained parts. We show simulations of this function in section 4. Third, the residual activity is specific in the exact symbol from the local alphabet (the minicolumn) it appears in. The residual activity in the A2-system thus fulfills the necessary prerequisites to enable incremental learning in the cortical hierarchy: It indicates *when* to learn, *what* to learn, and also *where* to learn it.

Our third example showed that an “unknown” stimulus cannot be represented across the hierarchy of areas in a self-consistent way. Instead, at all hierarchical levels, the system keeps “associating” possible symbols that are locally compatible both with the momentary bottom-up-stream of signals, and the momentary top-down-stream of hypothesized symbols. This state of activation is clearly different from the case of stable activation with residuals. It indicates that the knowledge *does not suffice* to explain the stimulus. It is important that the neural system has a means to indicate this conflict, instead of converging into some stable, but necessarily inappropriate state of activation that would ultimately deceive the individual into taking wrong actions and drawing wrong conclusions. Still, this type of activation is more than a pure “error-state”: Activations tend to converge towards sets of repeating symbols. The exact sets of alternating symbols are determined by the interaction between the evident signal, and the system’s knowledge imprinted in the synapses. Thus, they are specific to the stimulus: The system starts to paraphrase the stimulus in its own terms. — The human drive to make sense from everything could actually be rooted in the core circuitry of our brains. It is also an ideal basis for the formation of new stimulus concepts. The repeating set of cognitive symbols itself can be learned as a description of the new stimulus. We think, that the hippocampus is the instance on top of the cortical hierarchy that performs this transformation [25].

3.3 Stimulus recollection from a top-down "idea"

Different cortical areas specialize on processing of different stimulus aspects (like movement, color, simple shape, object identity, location, for the visual modality). A stimulus that is coherent on the retina and sensors of other modalities casts more or less independent "shadows" across the cortex: The interplay of cortical areas decomposes it according to its respective aspects, which are then represented in the various specialized areas. This decomposition poses a problem, when a stimulus shall be recalled, be it to support the processing of a physically existing stimulus that is hard to recognize, or be it to create a mental image of a non-existing stimulus. We believe that during recollection, the cortex reconstructs the same activity patterns as when analyzing the corresponding physical stimulus. Pre-activations in highly specialized areas are then the starting points for creating a consistent activation on all levels of the cortex and thus the percept of the stimulus. In the process of recollection, high brain areas such as the prefrontal cortex or the hippocampus activate coarse descriptions of independent stimulus aspects in the various specialized areas, which are usually invariant with respect to each other (like position and shape). To create consistent neural activity across all cortical areas, the independent top-down activations need to be integrated with each other, and with the physical stimulus.

The COREtext model of the last section demonstrated how the cortex can use a rapidly formed hypothesis for guiding the detailed stimulus recognition process. We showed how the columnar subsystems interact to predict and recognize stimuli in terms of locally stored knowledge. In this model, columnar communication integrated bottom-up signals with internally generated top-down signals to describe the stimulus consistently across all cortical areas. This leads to the question what happens, if a hypothesis is not generated bottom-up through a fast recognition process, but given as a top-down signal, which is not necessarily coherent with the bottom-up stimulus. Will it guide processing of a stimulus towards the expectation? Here we extend our previous model to demonstrate that the same setup of intercommunicating columns can use the stored knowledge to integrate a pre-activation on the highest level with the bottom-up recognition process. Given only coarse or invariant top-down activation, the model can (i) guide and support the recognition of noisy or ambiguous stimuli, and (ii) recall known objects, at the highest level of detail, by creating specific neural activations across all cortical areas. The second process corresponds to recollection or mental imagery, the process in which the brain internally creates a percept from an "idea", without a physical stimulus.

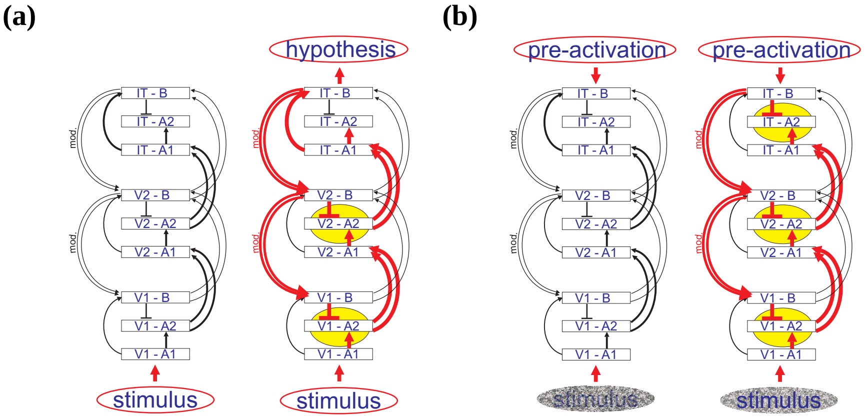


Figure 20: Recognition and recall in the COREtext model. *(a)*, bottom-up mode (recognition); *(b)*, top-down mode (recall).

3.3.1 Model

The COREtext model is based on the cortical columns as the fundamental building block of the cortex. Their reappearing architecture of neural subsystems and signal flow between them forms the basis for the associative capabilities of the cortex. The different processing subsystems interact to switch off expected signals and accomplish symbolic recognition of words. In this process, the model integrates bottom-up signals with internally generated top-down predictions at all levels of processing. In this purely bottom-up-driven mode of the COREtext model (fig. 20 *(b)*), a retinal stimulus rapidly activates internal hypotheses about the stimulus in all areas, via the A1→A2→A1 forward path. At the same time, these hypotheses are routed backwards via the modulatory B→B path, and integrated with the bottom-up stream by switching off the stimulus parts that meet the hypothesis (yellow circles in fig. 20). By iterating this process, the system establishes a consistent description of the stimulus across all cortical areas.

Here, we extend this model by replicating the previously location-specific connectivity at all retinal locations, making recognition of words invariant to their retinal location. Furthermore, we introduce the possibility to stimulate not only the A1-system of "V1", but also the B-system of "IT". In this top-down mode (fig. 20 *(c)*), the bottom-up stimulus can be unspecific or noisy, but is complemented by an already existing top-down pre-activation. The model then integrates the bottom-up and top-down signals to find a state of activation that is consistent with both at all cortical levels.

The top-down activation supports recognition of a stimulus in several ways. (i) If the stimulus is noisy and could not be recognized in the pure bottom-up-driven mode, the pre-activation of the highest area supports weak bottom-up activations that are consistent with the top-down signal. It stabilizes recognition of the stimulus in the B-systems while inconsistent parts or noise leave residual activity in the A2-systems indicating where the stimulus did not meet the expectation. (ii) If the stimulus is ambiguous and did not lead to a stable pattern of activity, because no consistent description across all levels could be found, pre-activation of one of the alternative objects (words) in the highest area stabilizes the recognition of this object, and marks the other parts of the stimulus as errors, which leave residual activation in the A2-systems. In both cases, the dynamics of the interacting neural subsystems promotes the top-down influence across all model areas. (iii) If the physical stimulus is unspecific or missing, the top-down activation shapes the diffuse bottom-up activation towards recognition of the respective object. Because the dynamics of the interacting neural subsystems strives towards consistent neural activity on all cortical levels, it (re-)creates a detailed and specific mental image of the recalled object.

3.3.2 Results

Here, we show results of (iii) and describe how the COREtext model reconstructs a spatially detailed representation of characters, syllables and words, given only coarse location- and feature-invariant top-down input. We started from the following setup:

1. IT-B receives pre-activation of the word “ach” indicating that it shall recall this word, but no information, where on the retina the word would be located. I.e., “IT” is completely certain about the feature (word), but completely uncertain about the location.
2. V1-A1 receives pre-activation of a certain location, indicating that it shall recall *something* located there, but no information on what it would be. I.e., “V1” is completely certain about the location, but completely uncertain about the features (letters).
3. No stimulation is applied in the intermediate area “V2”.

The initial distribution is shown in figure 21. Note that the activation in V1-A1 is completely unspecific with respect to the feature (all possible characters were activated, not all of them visible in the figure), while activation in IT-B is completely unspecific with respect to the location. We then allowed activity to iterate according to the synaptic connectivity, in the same way it does when the COREtext model performs recognition of a real word stimulus. Given that the diffuse activation in “V1” supplies the system with enough energy, activity spreads from “V1” to “V2”, where it meets with the modulatory top-down input from “IT”. After four iterations, activity is distributed as shown in figure 22. Note that (re-)constructions of the stimulus at the highest possible fidelity have appeared in the B-systems of “V1” and “V2”: “V2” has constructed the syllable ”ach” at the correct position (4), and “V1” has constructed the letters ”a”, ”c”, ”h” at the correct positions (4, 5, 6). In the following, the system associates previously learned knowledge to the recalled stimulus, in the same way it does for real stimuli (like the words “betrachtung” and “nacht”), see figure 23. Note also, that the specific activity is self-stabilizing. After turning off the top-down input to “IT”, consistent activation of ”ach” at position 4 remained across all areas.

```
t= 0     ................................................................................
-----                                       ^---------
IT-A1:  0:                                IT-A2:  0:                                IT-B :  0: ach(1.0)
IT-A1:  1:                                IT-A2:  1:                                IT-B :  1: ach(1.0)
IT-A1:  2:                                IT-A2:  2:                                IT-B :  2: ach(1.0)
IT-A1:  3:                                IT-A2:  3:                                IT-B :  3: ach(1.0)
IT-A1:  4:                                IT-A2:  4:                                IT-B :  4: ach(1.0)
IT-A1:  5:                                IT-A2:  5:                                IT-B :  5: ach(1.0)
IT-A1:  6:                                IT-A2:  6:                                IT-B :  6: ach(1.0)
IT-A1:  7:                                IT-A2:  7:                                IT-B :  7: ach(1.0)
IT-A1:  8:                                IT-A2:  8:                                IT-B :  8: ach(1.0)
IT-A1:  9:                                IT-A2:  9:                                IT-B :  9: ach(1.0)

V2-A1:  0:                                V2-A2:  0:                                V2-B :  0:
V2-A1:  1:                                V2-A2:  1:                                V2-B :  1:
V2-A1:  2:                                V2-A2:  2:                                V2-B :  2:
V2-A1:  3:                                V2-A2:  3:                                V2-B :  3:
V2-A1:  4:                                V2-A2:  4:                                V2-B :  4:
V2-A1:  5:                                V2-A2:  5:                                V2-B :  5:
V2-A1:  6:                                V2-A2:  6:                                V2-B :  6:
V2-A1:  7:                                V2-A2:  7:                                V2-B :  7:
V2-A1:  8:                                V2-A2:  8:                                V2-B :  8:
V2-A1:  9:                                V2-A2:  9:                                V2-B :  9:

V1-A1:  0:                                V1-A2:  0:                                V1-B :  0:
V1-A1:  1:                                V1-A2:  1:                                V1-B :  1:
V1-A1:  2:                                V1-A2:  2:                                V1-B :  2:
V1-A1:  3:                                V1-A2:  3:                                V1-B :  3:
V1-A1:  4: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z  V1-A2:  4:                                V1-B :  4:
V1-A1:  5: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z  V1-A2:  5:                                V1-B :  5:
V1-A1:  6: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z  V1-A2:  6:                                V1-B :  6:
V1-A1:  7:                                V1-A2:  7:                                V1-B :  7:
V1-A1:  8:                                V1-A2:  8:                                V1-B :  8:
V1-A1:  9:                                V1-A2:  9:                                V1-B :  9:
```

Figure 21: Example of stimulus recollection, phase 1. “V1” receives spatially localized but unspecific pre-activation. “IT” receives space-invariant but specific pre-activation of the word “ach”.

```
t= 4     ..............................................................................
-----                                      ˆ---------
IT-A1:  0:                               IT-A2:  0:                               IT-B :  0: ach(1.0)
IT-A1:  1:                               IT-A2:  1:                               IT-B :  1: ach(1.0)
IT-A1:  2:                               IT-A2:  2:                               IT-B :  2: ach(1.0)
IT-A1:  3:                               IT-A2:  3:                               IT-B :  3: ach(1.0)
IT-A1:  4:                               IT-A2:  4:                               IT-B :  4: ach(1.1)
IT-A1:  5:                               IT-A2:  5:                               IT-B :  5: ach(1.0)
IT-A1:  6:                               IT-A2:  6:                               IT-B :  6: ach(1.0)
IT-A1:  7:                               IT-A2:  7:                               IT-B :  7: ach(1.0)
IT-A1:  8:                               IT-A2:  8:                               IT-B :  8: ach(1.0)
IT-A1:  9:                               IT-A2:  9:                               IT-B :  9: ach(1.0)

V2-A1:  0: fleisch(0.4)                  V2-A2:  0: fleisch(0.1)                  V2-B :  0:
V2-A1:  1: stillt(0.4) spräch(0.4) spran V2-A2:  1: stillt(0.1) spräch(0.1) spran V2-B :  1:
V2-A1:  2: wuchs(0.4) wasch(0.4) ufers(0 V2-A2:  2: wuchs(0.1) wasch(0.1) ufers(0 V2-B :  2:
V2-A1:  3: über(0.4) zens(0.4) zend(0.4) V2-A2:  3: über(0.1) zens(0.1) zend(0.1) V2-B :  3:
V2-A1:  4: über(0.4) ßen(0.4) zur(0.4) z V2-A2:  4: über(0.1) ßen(0.1) zur(0.1) z V2-B :  4: ach(0.4)
V2-A1:  5: über(0.1) üb(0.1) ßen(0.1) zu V2-A2:  5:                               V2-B :  5:
V2-A1:  6:                               V2-A2:  6:                               V2-B :  6:
V2-A1:  7:                               V2-A2:  7:                               V2-B :  7:
V2-A1:  8:                               V2-A2:  8:                               V2-B :  8:
V2-A1:  9:                               V2-A2:  9:                               V2-B :  9:

V1-A1:  0:                               V1-A2:  0:                               V1-B :  0:
V1-A1:  1:                               V1-A2:  1:                               V1-B :  1:
V1-A1:  2:                               V1-A2:  2:                               V1-B :  2:
V1-A1:  3:                               V1-A2:  3:                               V1-B :  3:
V1-A1:  4: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  4: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  4: a(0.3)
V1-A1:  5: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  5: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  5: c(0.3)
V1-A1:  6: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  6: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  6: h(0.3)
V1-A1:  7:                               V1-A2:  7:                               V1-B :  7:
V1-A1:  8:                               V1-A2:  8:                               V1-B :  8:
V1-A1:  9:                               V1-A2:  9:                               V1-B :  9:
```

Figure 22: Example of stimulus recollection, phase 2. The model integrates the top-down and bottom-up activations. Although both pre-activations were partially invariant, the integration result in V1-B and V2-B is specific both in features and their locations.

```
t=23     ................................................................................
-----                                       ^---------
IT-A1:  0: be-trach-tung(0.4)            IT-A2:  0: be-trach-tung(0.2)            IT-B :  0: be-trach-tung(0.7)
IT-A1:  1: spre-chen(0.1)                IT-A2:  1:                               IT-B :  1:
IT-A1:  2: wu-schen(0.2) wasch-un-gen(0. IT-A2:  2:                               IT-B :  2:
IT-A1:  3: nacht(0.4)                    IT-A2:  3: nacht(0.2)                    IT-B :  3: nacht(0.7)
IT-A1:  4: ach(0.4)                      IT-A2:  4: ach(0.2)                      IT-B :  4: ach(0.7)
IT-A1:  5:                               IT-A2:  5:                               IT-B :  5:
IT-A1:  6:                               IT-A2:  6:                               IT-B :  6:
IT-A1:  7:                               IT-A2:  7:                               IT-B :  7:
IT-A1:  8:                               IT-A2:  8:                               IT-B :  8:
IT-A1:  9:                               IT-A2:  9:                               IT-B :  9:

V2-A1:  0: fleisch(0.7)                  V2-A2:  0: fleisch(0.4)                  V2-B :  0: fleisch(0.9)
V2-A1:  1: spräch(0.7)                   V2-A2:  1: spräch(0.4)                   V2-B :  1: spräch(0.9)
V2-A1:  2: trach(0.9)                    V2-A2:  2: wasch(0.4) raucht(0.4) leuch( V2-B :  2: trach(3.0)
V2-A1:  3: wach(0.9) nacht(0.9)          V2-A2:  3: wach(2.1)                     V2-B :  3: nacht(3.0)
V2-A1:  4: ach(0.9)                      V2-A2:  4: schö(0.4) schwar(0.4) schul(0 V2-B :  4: ach(3.0)
V2-A1:  5: cher(0.4) chen(0.4)           V2-A2:  5: cher(0.2) chen(0.2)           V2-B :  5: cher(0.7) chen(0.7)
V2-A1:  6:                               V2-A2:  6:                               V2-B :  6:
V2-A1:  7:                               V2-A2:  7:                               V2-B :  7:
V2-A1:  8:                               V2-A2:  8:                               V2-B :  8:
V2-A1:  9:                               V2-A2:  9:                               V2-B :  9:

V1-A1:  0:                               V1-A2:  0:                               V1-B :  0:
V1-A1:  1:                               V1-A2:  1:                               V1-B :  1:
V1-A1:  2:                               V1-A2:  2:                               V1-B :  2:
V1-A1:  3:                               V1-A2:  3:                               V1-B :  3:
V1-A1:  4: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  4: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  4: a(3.0)
V1-A1:  5: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  5: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  5: c(3.0)
V1-A1:  6: ß(0.6) ü(0.6) ö(0.6) ä(0.6) z V1-A2:  6: ß(0.4) ü(0.4) ö(0.4) ä(0.4) z V1-B :  6: h(3.0)
V1-A1:  7:                               V1-A2:  7:                               V1-B :  7:
V1-A1:  8:                               V1-A2:  8:                               V1-B :  8:
V1-A1:  9:                               V1-A2:  9:                               V1-B :  9:
```

Figure 23: Example of stimulus recollection, phase 3. The system associates previously learned knowledge to the recalled stimulus, in the same way it does for real stimuli.

3.3.3 Discussion

We have shown that in the COREtext model, a top-down pre-activation of an object (word) description promotes recognition of a noisy or unspecific stimulus. The dynamics of the interacting neural subsystems promotes the top-down influence across all model areas and creates detailed and specific activity at all cortical levels. The modified COREtext model uses shift-invariant synaptic connectivity. This allows for rich combinatorics of the participating features, as every feature can be recognized at all possible retinal locations. At the same time it introduces van der Malsburg's classical *superposition problem* [42, often referred to as the "binding problem"]: A top-down pre-activation of a word without a specific location pre-activates all its constructive syllables and characters at all possible locations. This leads to a cloud of pre-activated characters lacking any information on their spatial relations.

The simulation shows that the bidirectional signal flow in our model is able to resolve the spatial ambiguities and produce a high-fidelity stimulus reconstruction that is detailed and specific regarding features, their locations and their spatial relations. In particular we have demonstrated that it is sufficient to give (1) a completely location-invariant top-down input to "IT", and (2) a completely feature-invariant bottom-up input to "V1", in order to dynamically reconstruct a full-detail mental image across all areas. The process can be seen as a model for the way in which the cortex recalls a full-detailed, feature- and location-specific description of an object. Although the pre-activation can be invariant to certain stimulus aspects, very specific recollections can nevertheless be created. Top-down activation needs to be given to the specialized areas representing specific stimulus aspects. After this bootstrapping, the associative capabilities of the cortex, implemented in the distinct functional circuitry of the cortical columns, reconstruct a detailed recollection or mental image of the stimulus.

We have seen how the sensory cortex can rapidly classify a newly encountered stimulus, and how it can use this first information to guide a detailed recognition or recollection, in which it incorporates all of its knowledge about stimuli and their composition. Consequently it is interesting to know, how the brain acquires new knowledge by learning from its environment, and how it permanently stores it in a way that the recognition process can use it.

4 From static knowledge to life-long learning

In both examples we have seen in the last section – the purely stimulus-driven bottom-up mode of the COREtext model, and the top-down mode integrating a more or less unrelated idea of a stimulus into the recognition process – the sensory cortex analyzed a static stimulus pattern applying a highly dynamic recognition process. Activity across all cortical subsystems and across all sensory areas iterates to find a description of the stimulus that is consistent with the knowledge stored in the synaptic connectivity of the sensory cortex. Both the stimulus as well as the knowledge applied to recognize it were however assumed to be static in these examples. Neither did the sensory stimulus change during the process of recognition, nor did our model cortex learn anything new about the stimuli it encountered. Moreover, the simulation comprised only the A- and the B-system of our layered cortex model, not the C-system in the lower layers of the sensory cortex, which we link to behavior control and the prediction of a changing stimulus over time (figs. 13 and 14).

4.1 The brain acquires knowledge by actions

For an individual acting in a natural environment (at least for higher species like mammals), it is obvious that the stimuli it encounters are *never* static. This is not only because the environment may change, but mainly because the individual itself is constantly changing the relation of its own body to the outer world by all sorts of behaviors. It will change its place by locomotion and manipulate objects with its limbs. It will scan its surroundings through its senses, moving the ears and the eyes, and concentrate on different aspects in its environment.

Recognition alone is worthless, because its ultimate use is to derive actions that allow the individual to survive. Be it by fleeing from a predator, finding shelter, or by spotting and collecting food.

In the following, we embed our concept of the six-layered cortical column (sec. 3.1) into a model of the whole visual system. Here, we move on from dynamic recognition of static stimuli, to stimulus prediction and learning as a process over time. We discuss a large-scale implementation of the visual system involving several primary and higher visual cortical areas as well as parts of the hippocampal formation and further sub-cortical structures involved in generating eye saccades. With this model we can demonstrate object classification and the learning of new object representations based on the incremental refinement of an object hypothesis. The first view initiates an object hypothesis. The incremental refinement happens by deriving behavioral output from the lower layers 5 and 6 of the sensory cortex (the C-system in our conceptual model), which is used to generate eye saccades. With every saccade, the sensory cortex decides what part of its environment to see next in order to integrate evidence over seconds and minutes and learn the composition of stimuli. It predicts the corresponding new object views and, by comparing the actual object view with the predicted object view, strengthens or weakens the initial hypothesis.

4.1.1 Model

We model several primary (V1, V2) and higher visual cortical areas from the "what" (V4, IT) and "where" path (V6) as well as parts of the hippocampal formation (EC) and further sub-cortical structures involved in generating eye saccades and triggering learning (see fig. 25, *right panel*). At this point we use a reduced setup sufficient to simulate and explore the basic neuronal dynamics and columnar functions without too much computational expenses. Most neuron populations are modeled either as a simple k-WTA population (i.e., at each simulation step, the k most excited neurons are activated), or as simplified spiking associative memories ([20–22]; cf. [12]).

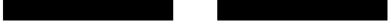


Figure 24: *Top:* We used simple line drawings for stimulating the visual model, e.g., a house or a church. Preprocessing in areas R, V1, V2, V6 essentially extracts key features (such as corners) as the basis for saccade generation and object view recognition in the higher areas. *Bottom:* V6 representations of the key features corresponding to the house and church stimuli.

In retinal area R we represent binary images (size 81×81) of simple line drawings of buildings (see fig. 24). The cells in primary visual area V1 receive input via oriented Gabor-like filter kernels (at each spatial location 8 different orientations). Area V2 represents additionally corner parts (i.e., conjunctions of two local orientations at appropriate spatial positions, 120 per location). Area V6 averages over all the corners at each spatial location thereby representing the positions of "key features" in a visual scene (see fig. 24). Area V4 represents invariant visual object views at a lower spatial resolution. Area IT ("inferior temporal") represents visual objects. For learning new objects we activate a static representation in an "auditory" area AC in order to simulate the representation of a spoken word labeling the new object. Converging multimodal (visual, auditory) input to a further area EC (entorhinal cortex) of the hippocampal complex allows the binding of different object views to a single object representation. Furthermore there are a number of presumably sub-cortical auxiliary areas involved in generating saccades and triggering learning (areas Sac1, Sac2, Sac3, SacX, LX). Most importantly we have implemented a simple model of the superior colliculus (SC/Sac2) representing target locations of saccades determined by the location of *key features* (input from V6) and an spatial attentional window (region of interest represented in area Sac1 and biased by V4).

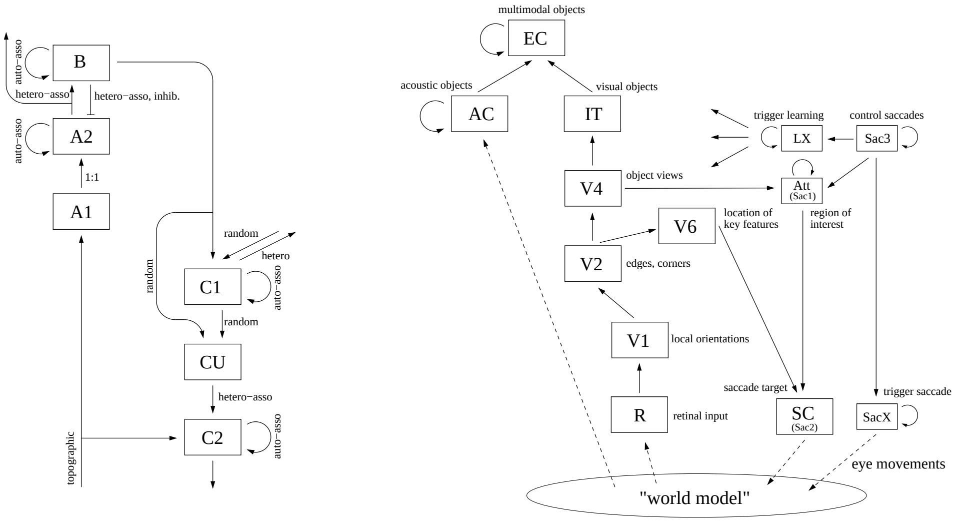


Figure 25: *Left:* Columnar model as implemented for our current simulations. Boxes correspond to cell populations (corresponding to particular cortical layers, cf. fig. 14), arrows to synaptic connections between the cell populations/layers. Populations with recurrent connections (A2, B, C1, C2) are implemented as simplified spiking associative memories [20–22]. The additional population CU is used to combine representations of B and C1 (cf. fig. 13). *Right:* Layout of our visual model of saccadic object recognition. The model consists of various visual areas (R, V1, V2, V4, V6, IT), auditory areas (AC), hippocampal areas (EC), saccade related areas (SC, S1, S3), and some auxiliary areas triggering learning and the execution of saccades (LX, SX). Currently only area V4 implements the full columnar model.

Each area consists of one or several neuron populations in order to implement our concept of the layered architecture of cortical columns (sec.3.1). Figure 25 (*left*) shows the architecture of our columnar model as implemented for area V4. The other areas implement only parts of our columnar model, mostly the fast forward processing A-system. In the following we give a detailed description of our columnar model as implemented in area V4. Currently, the columnar model consists of six neuron populations. Population A1 is the input layer corresponding to layer 4 cells. It consists of 120×9 cells corresponding to the 120 V2 features and nine spatial fields (central, plus eight peripheral), pooling over the corresponding (much larger set of) V2 cells. Our A1 cells have no local recurrent excitatory connections but feed-forward and lateral inhibition with a "soft" winner-takes-all is emulated by activating only the $k = 13$ most excited cells (k-WTA). A1 cells project in a one-to-one manner to population A2 corresponding to lower layer 3. A2 has the same size as A1 but is modeled as a spiking associative memory [20–22]. Thus A2 has recurrent excitatory connections which are used to store auto-associatively feature vector prototypes learned during saccadic object recognition. Similarly, population B (size 100) is also modeled as spiking associative memory but the auto-associatively learned patterns consist of randomly selected cells ($k = 5$) and therefore have rather symbolic character. Population B receives hetero-associative input from A2, where backprojections of B can inhibit A2 thereby emulating the ideas of quenching off expected signals (as described in the CORE-text model in sec. 3.2). The representation of B can be used to bias the selection of behavior relevant symbols in population C1 (corresponding to cortical layer 5) which is also realized as spiking associative memory of 100 cells representing currently nine possible target directions of saccadic eye movements (central plus eight equispaced directions). During learning of new objects this population receives random inputs and thereby biases the production of random saccades to explore new visual scenes or objects. An additional neuron population CU ("combinatorial units") is used to represent conjunctions (w, a) of "world states" and "actions" (see fig. 13). The most simple way to do this is to model CU as a k-WTA population (where we used population size 100 or 2500 and $k = 13$) receiving inputs from B and C1 via random connections. This will lead to nearly "uncorrelated" activity in CU for any combination of w and a, which useful to hetero-associatively link conjunctions (w, a) to predicted states w' represented in population C2. C2 is modeled again as spiking associative memory of size 100 (and assembly size $k = 5$). The short-cut link of the external sensory input (from V2) to C2 is used to learn the prediction $(w, a) \rightarrow w'$ (see below for more details). The prediction represented in population C2 can be used to bias expections in other (lower) areas of the cortical hierarchy and/or to narrowing the search space within the same cortical column (e.g., via a modulating input from C2 to A1, A2, or B).

Figure 26 illustrates the time course of activation in the different parts of the cortical column. Note that different states of the external stimulus and its internal interpretations are present in different subsystems at the same time.

Learning may occur in three different subsystems of the cortical columns.

1. (PCA-like) basis-vector learning system for populations A1 and A2.

2. Clustering-like learning algorithm in populations A2 and B.

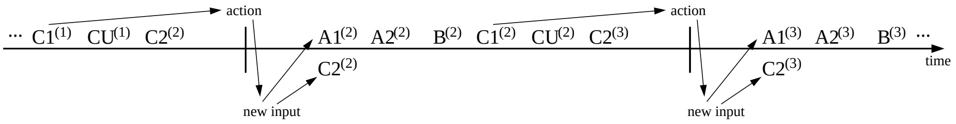


Figure 26: Time course of activation in the different populations or layers in a cortical column within an action/perception cycle. An action (for example a saccade) causes new sensory input which reaches first A1 and via the input shortcut C2. Then the main stream proceeds by activating A1, B, C1, CU, and C2. The shortcut is used to learn the predictive hetero-association $CU^{(i-1)} \rightarrow C2^{(i)}$. Assuming the cell populations being located as illustrated in fig. 14 this is consistent with neurobiological results [e.g. 11]. Indices $^{(i)}$ refer to the i-th object view.

3. Predictive learning on symbolic states in populations B, C1, CU, and C2.

The first learning system is concerned with generating appropriate basis vectors for adequately describing the sensory inputs. So far this learning system has not been implemented in our model, i.e., all sensory inputs to layer A1 are hard-wired. Instead, we have conducted some isolated simulation experiments (unpublished results) which suggest that a simple standard statistical learning procedures (such as essentially additive Hebbian learning plus an adequate synaptic normalization procedure) will do the job similar to PCA or maximation of reconstruction quality and sparseness which can generate plausible receptive field properties [29, 30].

The second learning system involving populations A2 and B performs a kind of clustering on the space spanned by the basis vectors of A1/A2, quite similar to ART networks [7] but relying on cell assemblies instead of simple nodes. Essentially, our current implementation performs the following operation,

$$\text{IF } (|A2 - H \cdot B| > \Theta) \text{ THEN NEW SYMBOL}$$

where A2 and B denote the activity vectors of the corresponding neuron populations, H is the matrix of hetero-associative inhibitory connections from B to A2, and $| \ldots |$ essentially sums over the residual activity after subtracting B from A2. This learning process involves the following steps. Input from A1 initiates a retrieval in the spiking associative memory A2, i.e., A1 will activate a prototype (or a mixture of several prototypes) in A2 that is most similar to A1. Then A2 will activate a corresponding cell assembly in the spiking associative memory B. This activation pattern is fed back via inhibitory hetero-associative connections to A2. Ideally, the two representations in A2 and B match each other and B will quench the activity in A2. However, if the activity vector in A2 is too far from a previously learned protoype then there will be considerable residual activity. This then initiates the learning of new “symbols”. For that, the original activity of A1 and A2 is stored auto-associatively in A2 and a new “symbolic” cell assembly (generated by noise) is stored in B and bidirectionally and hetero-associatively linked with the new A2 assembly. Additionally, a new cell assembly (of the same quality as A2) is stored in C2.

The third learning system essentially learns to make predictions on the symbolic states learned by the second learning system. Functionally, it implements the learning of the conditional probability density histograms $p(w'|w, a)$ illustrated in fig. 13. This involves the following steps. First a new unquenched sensory signal s enters the column. On the main path it will travel via A1, A2, B, C1 finally to C2. However, on the shortcut to cortical layer 6 it will directly enter C2 and transiently activate, by hetero-association, a symbol corresponding to the new sensory input. Since at that time the old state symbols are still active in B, C1, and CU, a simple asymmetric STDP-like Hebbian learning rule will hetero-associatively link the old CU-representation to the new C2 symbol. We acknowledge that it may be challenging to create a more detailed neurodynamic model which implements the algorithmic learning procedures in our model using realistic spiking neuron and plasticity models.

4.1.2 Results

In order to test our model we have conducted simulations applying simple line stimuli as shown in fig. 24. In each simulation run we presented one new stimulus or several new stimuli in a sequence. During the presentation of a stimulus the system will saccade on the key features (i.e., corners) of the stimulus. “Eye movements” are controlled by the saccade control system (areas S1, S2, S3, SX, LX) as follows: In the first phase the system executes a saccade defined by the visual target map in area SC/S2 (superior colliculus). After the saccade it follows up to three correction saccades in order to center the fixation on the location of the most salient feature. After that learning is enabled for ten simulation steps by modulating input from area LX (which could correspond to hippocampal and/or subcortical areas) to several areas: (i) within the A/B system of area V4 in order to learn a new object view in case the current object view differs too much from the views experienced previously; (ii) within the C system of area V4 in order to learn to predict the outcome of the saccade; (iii) between area IT and EC in order to associate the particular object view representation in V4 and IT with a static auditory representation in the auditory area AC via entorhinal area EC. At the same time, population C1 of area V4 preselects a future saccade direction out of nine possible directions ($2\pi\frac{i}{8}$ for $i = 0, 1, \ldots, 7$ or “center”). Via area S1 this biases a particular region of the visual field in area SC. Since SC receives also the locations of the key features from area V6, the most active cells in the target map of SC can again select the location of the following saccade.

Figure 27 shows simulation results when stimulating with the house stimulus illustrated in fig. 24. Data are taken from area V4 where the full columnar model has been implemented (fig. 25). The plot shows the probability of making a correct prediction about a future object view before executing a saccade as a function of the number of saccades when learning a new object (here the house of fig. 24). The different lines correspond to different model parameters. Best results (*blue line 3*) with rapid convergence occur if population CU is large and the corresponding representations are uncorrelated, i.e., have random character, in order to minimize crosstalk. Then the predictive representation generated in the column is essentially equivalent to the prediction histogram illustrated in fig. 13. The results show

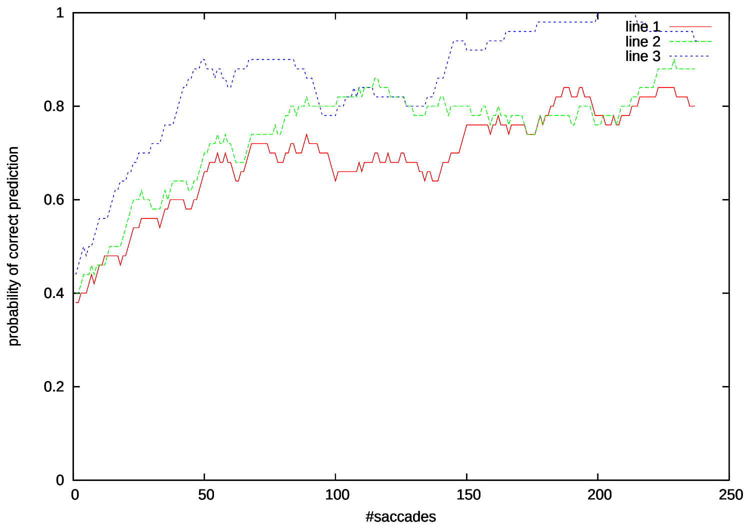


Figure 27: Simulation results from area V4 during learning of a new object (house of fig. 24). Probability of making a correct prediction about a future object view before executing a saccade, as a function of the number of saccades. *Red line 1*, $n_{\mathrm{CU}} = 100$, low position invariance; *green line 2*, $n_{\mathrm{CU}} = 100$, high position invariance; *blue line 3*, $n_{\mathrm{CU}} = 2500$, high position invariance.

that our model is able to rapidly learn new object views and to learn to predict the outcome of saccades under a certain object hypothesis.

4.1.3 Discussion

We have proposed that the basic function of the cortical microcircuitry is to learn to represent, actively predict, and confirm the columnar inputs [23]. One important feature is that several representations of the external stimulus co-exist in different parts of the column at a time. Another important feature is the close linkage of cortical representations to the generation of actions via the layer 5 pyramids. We demonstrated our ideas by a cell-assembly based implementation of a hypercolumn integrated with a large-scale model of the visual system for saccadic object recognition.

The model learns to associate retinal stimuli with the saccades that “cause” them (by relocating the gaze). In contrast to the COREtext model, which used predictive coding on a static stimulus, it learns to successfully predict the next retinal stimulus *over time*. Essentially, in order to learn the prediction the system has to learn something like the prediction histogram of figure 13. Combining a large number of world states with a large number of possible behaviors/actions, the complete histogram will be quite large. Here, we have compressed the representations by using cell assemblies where a state or state combination is not represented by a single cell or a conjunctive unit, but instead by distributed cell groups [4, 16, 32]. Theoretically, this could reduce the number of required neural units from

$M = \#\text{states} \times \#\text{actions}$ (if we use one conjunctive unit for each combination of world states and actions) to approximately $\sqrt{M}$ for large networks of distributed cell assemblies [21, 31].

5 Conclusion

In this chapter, we have collected a number of simulations that were designed to substantiate our concept of the cortex. While animals without a cortex have a very limited repertoire of reacting to a certain stimulus, we believe that the cortex of higher animals enables the individual to flexibly interpret a stimulus depending on the animal's needs. For this, the cortex needs to represent different interpretations of the sensory input at the same time, such as detailed and abstract descriptions of what is present, what is desired, what was expected from the context, and what is to be expected after the individual decides on a certain action. In our conceptual model, this multi-representation of sensory inputs and internal states of the individual is established in the different layers of the cortical tissue. Our main assumption is that the local neural circuitry of the cortical tissue is roughly the same in all areas, regardless of the information they process. In this local circuit (a column) that vertically links the six cortical layers, the cortex represents different interpretations of the sensory stimulus in the different layers. We follow a multi-level approach in simulating this cortical model. For the fast forward pathway of the columnar circuit (*A-system*), we believe that spike-based processing is essential to accommodate the temporal needs of the individual. In simulations on a single-neuron level we have shown that this pathway can rapidly extract key features of the stimulus using a spike-latency code (sec. 2). At the level of coupled neural circuits, we have simulated the bidirectional signal flow in a hierarchy of columns (*A- and B-systems*). We have shown how dynamic integration of bottom-up and top-down signals with locally stored knowledge guides recognition, and enables the cortex to interpret a sensory stimulus differently depending on its internal state, e.g. an expectation of the individual (sec. 3). Finally, we have embedded an implementation of the full columnar circuit on the level of spiking associative memories into a large-scale model of the visual system. We have demonstrated how the columnar *C-system* interacts with the A- and B-systems to derive motor actions that allow the cortex to actively confirm an object hypothesis. The model acquires support for an object hypothesis by saccading to expected key features, and learns to associate new object views with each other (sec. 4).

By instantiating overlapping aspects of our cortical model at different levels of abstraction, we can validate our concept both on the system level as well as on the level of detailed circuitry. We will follow this strategy to maintain and improve a model that enables us to understand, how macroscopic cortical function comes along from the underlying neurobiological structure based on the processes this structure implements.

References

[1] M. Abeles. *Corticonics: Neural circuits of the cerebral cortex.* Cambridge University Press, Cambridge UK, 1991.

[2] M. Abeles. Role of cortical neuron: integrator or coincidence detector? *Israel Journal of Medical Sciences*, 18:83–92, 1982.

[3] A. P. Bannister. Inter- and intra-laminar connections of pyramidal cells in the neocortex. *Neurosci. Res.*, 53(2):95–103, Oct. 2005.

[4] V. Braitenberg. Cell assemblies in the cerebral cortex. In R. Heim and G. Palm, editors, *Lecture notes in biomathematics (21). Theoretical approaches to complex systems.*, pages 171–188. Springer-Verlag, Berlin Heidelberg New York, 1978.

[5] V. Braitenberg and A. Schüz. *Anatomy of the cortex. Statistics and geometry.* Springer-Verlag, Berlin, 1991.

[6] E. M. Callaway. Feedforward, feedback and inhibitory connections in primate visual cortex. *Neur. Netw.*, 17(5–6):625–632, June–July 2004.

[7] G. Carpenter and S. Grossberg. Adaptive resonance theory. In M. Arbib, editor, *The Handbook of Brain Theory and Neural Networks, Second Edition*, pages 87–90. MIT Press, Cambridge, MA, 2003.

[8] P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience. The MIT Press, Cambridge, Massachusetts, London, England, 2001.

[9] P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, chapter 5.4 Integrate-and-Fire Models. In *Computational Neuroscience*, Dayan and Abbott [8], 2001.

[10] A. Delorme. Early cortical orientation selectivity: How fast inhibition decodes the order of spike latencies. *J. Comp. Neurosci.*, 15(3):357–365, Nov.–Dec. 2003.

[11] R. J. Douglas and K. A. Martin. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27:419–451, 2004.

[12] R. Fay, U. Kaufmann, A. Knoblauch, H. Markert, and G. Palm. Combining visual attention, object recognition and associative information processing in a neurobotic system. In S. Wermter, G. Palm, and M. Elshaw, editors, *Biomimetic Neural Learning for Intelligent Robots*, volume 3575 of *Lecture Notes in Articifical Intelligence*, pages 118–143. Springer-Verlag, Berlin Heidelberg, 2005.

[13] D. J. Felleman and D. C. van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*, 1(1):1–47, Jan.–Feb. 1991.

[14] M.-O. Gewaltig, U. Körner, and E. Körner. A model of surface detection and orientation tuning in primate visual cortex. *Neurocomp.*, 52–54:519–524, 2003.

[15] J. Hawkins. *On Intelligence.* Times Books Henry Holt, New York, 2004.

[16] D. Hebb. *The organization of behavior. A neuropsychological theory.* Wiley, New York, 1949.

[17] H. Hesse. *Siddhartha: eine indische Dichtung*, volume 2499 of *Project Gutenberg Etext*, lines 418–514. Project Gutenberg, `http://www.gutenberg.org/etext/2499`, twelth edition, Feb. 2001.

[18] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.*, 195(1):215–243, Mar. 1968.

[19] R. S. Johansson and I. Birznieks. First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nature Neurosci.*, 7(2):170–177, Feb. 2004.

[20] A. Knoblauch. Synchronization and pattern separation in spiking associative memory and visual cortical areas. *PhD thesis, Department of Neural Information Processing, University of Ulm, Germany*, 2003.

[21] A. Knoblauch. Neural associative memory for brain modeling and information retrieval. *Information Processing Letters*, 95:537–544, 2005.

[22] A. Knoblauch and G. Palm. Pattern separation and synchronization in spiking associative memories and visual areas. *Neural Networks*, 14:763–780, 2001.

[23] A. Knoblauch, R. Kupper, M.-O. Gewaltig, U. Körner, and E. Körner. A cell assembly based model for the cortical microcircuitry. *Neurocomputing*, 2007.

[24] P. König, A. K. Engel, and W. Singer. Integrator or coincidence detector - the role of the cortical neuron revisited. *Trends Neurosci.*, 19:130–137, 1996.

[25] E. Körner, M.-O. Gewaltig, U. Körner, A. Richter, and T. Rodemann. A model of computation in neocortical architecture. *Neur. Netw.*, 12(7–8):989–1005, 1999.

[26] R. Kupper, M.-O. Gewaltig, U. Körner, and E. Körner. Spike-latency codes and the effect of saccades. *Neurocomp.*, 65–66C:189–194, 2005. Special issue: Computational Neuroscience: Trends in Research 2005 – Edited by E. de Schutter.

[27] L. Lapicque. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et de Pathologie générale*, 9:620–635, 1907.

[28] I. Nelken, G. Chechik, T. D. Mrsic-Flogel, A. J. King, and J. W. H. Schnupp. Encoding stimulus information by spike numbers and mean response time in primary auditory cortex. *J. Comp. Neurosci.*, 19(2):199–221, Oct. 2005.

[29] E. Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

[30] B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Visison Research*, 37:3311–3325, 1997.

[31] G. Palm. On associative memories. *Biological Cybernetics*, 36:19–31, 1980.

[32] G. Palm. *Neural Assemblies. An Alternative Approach to Artificial Intelligence.* Springer, Berlin, 1982.

[33] R. P. N. Rao and D. H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neurosci.*, 2(1): 79–87, 1999.

[34] P. Reinagel and R. C. Reid. Temporal coding of visual information in the thalamus. *J. Neurosci.*, 20(14):5392–5400, July 2000.

[35] R. Ritz, W. Gerstner, U. Fuentes, and J. van Hemmen. A biologically motivated and analytically soluble model of collective oscillations in the cortex. II. Applications to binding and pattern segmentation. *Biol. Cybern.*, 71:349–358, 1994.

[36] K. Tanaka. Inferotemporal cortex and object vision. *Annu. Rev. Neurosci.*, 19:109–139, 1996.

[37] A. M. Thomson and A. P. Bannister. Interlaminar connections in the neocortex. *Cereb. Cortex*, 13(1):5–14, Jan. 2003.

[38] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381(6):520–522, June 1996.

[39] S. Thorpe, A. Delorme, and R. van Rullen. Spike-based strategies for rapid processing. *Neur. Netw.*, 14(6–7):715–725, July 2001.

[40] H. C. Tuckwell. *Introduction to Theoretical Neurobiology*, volume 1, chapter 3, The Lapique model of the nerve cell, pages 85–123. Cambridge University Press, Cambridge, 1988. ISBN 0-521-35096-4.

[41] S. Ullman. *High-Level Vision: Object Recognition and Visual Cognition*. Psychology. MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142, USA, `http://mitpress.mit.edu`, 2000. ISBN 0-262-71007-2.

[42] C. van der Malsburg. The correlation theory of brain function. Technical Report 81-2, Dept. of Neurobiology, Max-Planck-Institute for Biophysical Chemistry, Göttingen, Germany, July 1981.

[43] R. van Rullen and S. J. Thorpe. Surfing a spike wave down the ventral stream. *Vis. Res.*, 42:2593–2615, 2002.

[44] R. van Rullen and S. J. Thorpe. Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. *Neur. Comp.*, 13(6):1255–1283, June 2001.

[45] R. van Rullen, R. Guyonneau, and S. J. Thorpe. Spike times make sense. *Trends Neurosci.*, 28(1):1–4, Jan. 2005.

[46] D. Willshaw, O. Buneman, and H. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222:960–962, 1969.