

**IMPLEMENTING  
POSITION-INVARIANT  
DETECTION OF  
FEATURE-CONJUNCTIONS IN A  
NETWORK OF SPIKING NEURONS**

---

This re-print corresponds to the paper “*Implementing Position-invariant Detection of Feature-conjunctions in a Network of Spiking Neurons*”, by Sander M. Bohte, Joost N. Kok, and Han La Poutré, to appear in the Proceedings of IJCNN’2002, Honolulu, Hawaii, 2002.

# Implementing Position-invariant Detection of Feature-conjunctions in a Network of Spiking Neurons

Sander M. Bohte<sup>1</sup>, Joost N. Kok<sup>1,2</sup> and Han La Poutré<sup>1,3</sup>

<sup>1</sup>CWI, Kruislaan 413, 1090 GB Amsterdam, P.O. Box 94079, The Netherlands.

<sup>2</sup>LIACS, Leiden University, 2300 RA Leiden, P.O. Box 9512, The Netherlands.

<sup>3</sup>School of Technology Management, TU Eindhoven, De Lismortel 2, 5600 MB Eindhoven, The Netherlands.

**Abstract**— The design of neural networks that are able to efficiently detect conjunctions of features is an important open challenge. We develop a feed-forward spiking neural network that requires a constant number of neurons for detecting a conjunction irrespective of the size of the retinal input field, and for up to four simultaneously present feature-conjunctions.

## I. INTRODUCTION

The representation of structured information in neural networks has so far not been satisfactorily solved, though it is thought to be required for efficiently solving a number of notoriously hard problems [1]. In a linguistic sentence like *The red apple and the green pear*, grammar implies the structuring of elements “red”, “green”, “apple”, and “pear” into semantic composites, e.g. structure denoted with brackets:  $\{\{red, apple\}, \{green, pear\}\}$ . The *binding-problem* refers to the problem of how to encode and detect such structured representations in neural networks. We can easily identify elements like *red*, *green*, *apple*, and *pear* each with a neuron that is activated when the element is used. However, the embodiment of the structural brackets has been much debated, as far back as Hebb [2]. Some have even argued that such structural representation is impossible in neural networks [3].

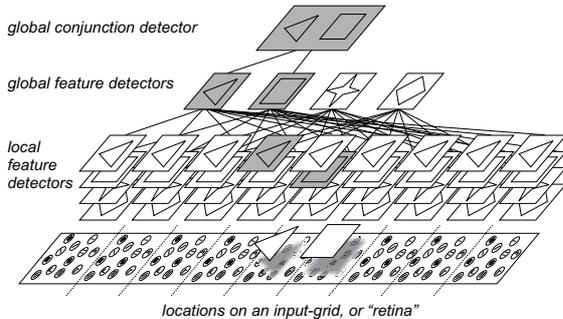


Fig. 1. Global conjunction detection via aggregation of activity in local feature detectors.

In the context of visual perception, the main concern is how to represent and/or detect conjunctions such as *red* and *apple* on a retina, where this conjunction of features can essentially appear anywhere. Creating a *red apple* detector for every location on the retina seems too expensive, at least for every sensible conjunction [1]. The straightforward solution, as also depicted in figure 1, would seem to first create global *apple* and *red* detectors by combining the responses of the respective local detectors, and then detect the *red apple* conjunction from these global

detectors. However, this architecture is prone to errors in the presence of multiple conjunctions, since there are no structuring “brackets” present in the encoding by neural activation (e.g. [1], [4]): the implicit links between *red* and *apple*, and *green* and *pear* are not represented.

For example, the presence of a “triangle-star” and a “hexagon-square” conjunction on the grid of figure 1 would activate the global “triangle” and “square” neurons, and hence *wrongly* the “triangle-next-to-square” neuron. (“ghosting”). The loss of local structure information, the “brackets”, in global detectors is also referred to as the “superposition catastrophe”.

In this paper, we present a feed-forward architecture based on spiking neurons that detects feature-conjunctions using global feature-detectors. It can detect these conjunctions also in the presence of multiple other conjunctions. Importantly, we observe that spiking neurons are more suitable for the superposition of inputs from multiple locations than traditional sigmoidal neurons: a spiking neuron that receives single timed spikes from  $n$  input locations can superimpose these  $n$  inputs by emitting  $n$  timed spikes. Thus, all  $n$  values are preserved, whereas a sigmoidal neuron would squash the  $n$  values into a single output value. We use this property in combination with a local procedure for encoding the (local) presence of a feature-conjunction. In [5], Rachkovskij and Kussul describe a procedure for encoding feature-binding via *Context Dependent Thinning* (CDT) operating on vectors of activity. We design a feed-forward CDT procedure for vectors of timed-spikes via *conditional shunting*. This procedure is implemented in local universal conjunction-detectors that locally perform feature-binding. In the architecture, the local presence of a feature is signaled via an activity vector of timed-spikes. The detectors process these vectors as the neural data-structure. Our local universal conjunction detector receives two such vectors as input. It outputs a vector generated by the CDT-procedure, if the inputs indicate the presence of *any* local feature-conjunction (without identifying the actual features). We aggregate these output-vectors in global universal conjunction-detectors. With vector-based local and global feature-detectors, the presence of *specific* feature-conjunctions can then be detected in specialized global feature conjunction detectors.

We demonstrate our architecture in an example that binds features based on relative proximity, as on the grid of fig. 1. In this architecture, a global detector for the conjunction of say  $\{\text{triangle, square}\}$  consists of some  $N$

neurons, a value independent of the number of input locations. With such global detectors, we can detect up to about 4 or 5 *similar* conjunctions simultaneously, but visual processing seems to be limited in the same way [6].

In this paper, we outline the architecture in section II, and the implementation in section III. The detection of conjunctions is demonstrated in section IV. We discuss and conclude the architecture in sections V and VI.

## II. ENCODING WITH ACTIVITY VECTORS

In this section, we outline a feedforward architecture for the global detection of feature-conjunctions. It is then implemented in spiking neural networks in section III.

### A. Architecture.

We propose an architecture as shown in figure 2. We introduce two local universal conjunction-detectors, denoted  $(X|Y)_R$  and  $(Y|X)_L$ , in addition to the local feature-detectors, denoted  $A, B, C$ , etc. . . The local conjunction-detectors detect and signal the presence of a conjunction of *any* two features. In our example, we consider the binding of shape-*right-next-to-shape*; the same framework can be applied to binding say color-to-shape. The signals of the local detectors are aggregated in respective global detectors, denoted  $\Sigma A, \Sigma B, \Sigma(X|Y)_R$  etc... The presence of particular feature-conjunctions is then detected from the combined information of the global universal conjunction and feature-detectors in dedicated, global detectors ( $\Sigma AB, \Sigma CA, \Sigma BA$  etc..). As we will show, the vector nature of the neural activity processed in these detectors enables the detection of the correct feature-conjunctions, also in the presence of multiple other conjunctions.

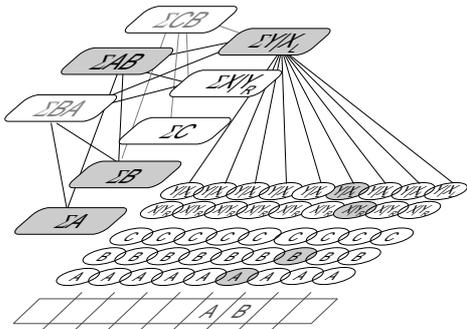


Fig. 2. Architecture for global detection of conjunctions.

The local detection of features can easily be considered in terms of activity-vectors. We assume that all (discrete) locations on an input-grid are populated with identical sets of diversely tuned basic neurons (e.g. grid in fig. 1). The presence of a feature like  $A$  is then characterized by the activity (spikes) it elicits in such a set of basic neurons. The timings of the spikes of the neurons for each set are collected in a vector, where each vector-element contains the activity of one neuron. The detectors in the proposed architecture process such *spike-time* vectors.

At the level of local detectors, we have local feature-detectors that look for a specific feature, say  $A, B, C$  etc. . . Each such detector looks at one set of basic neurons. If it detects that the local input vector sufficiently matches the preferred vector, it propagates the input-vector, with some delay due to computation: only if presented with input  $\vec{A}$ , a local  $A$ -detector outputs  $\vec{A}$ .

We also have local detectors that detect and signal the conjunction of *any* two features. These detectors consider two sets of basic neurons. The idea is that they detect the presence of features in both locations by only considering the actual *amount* of activity. In our example, these nodes look at two locations next to each other in the grid. We have complementary right- and left-facing detectors  $(X|Y)_R$  and  $(Y|X)_L$ . In the presence of say  $A$ -next-to- $B$ , these detectors respectively output vectors  $\vec{A}\setminus b$  and  $\vec{B}\setminus a$ , which each look like  $\vec{A}$  resp.  $\vec{B}$  (def. in section II-C).

The next level in the architecture combines the results of the local detectors. Here we exploit a specific property of spiking neurons: suppose we have two spike-trains each containing  $k$  spikes, then the combined spike-train can contain  $2k$  spikes (if the spikes all have different times). Thus, we can obtain global aggregate vectors by combining the local vectors, where the use of spiking neurons alleviate the “superposition catastrophe” encountered with sigmoidal neurons [1].

The vectors from the respective local detectors are combined to the output-vectors of global feature detectors (“there is a triangle”) and global conjunction-presence detectors (“there are two active consecutive locations”). In the output-vector of a global detector, an element contains to the collected vector of (active) spikes in the corresponding elements from the local detector-vectors. Finally, the vectors from the global detectors are used to detect the presence of *specific* consecutive features in a global feature-conjunction detector.

The detection of the specific-features next-to-each-other from the global detectors is possible, because we make use of a special local procedure: the output vector of a local universal conjunction-detector resembles the vector associated with one of the two features, but this vector is “watermarked” with the vector associated with the other feature. This “watermarking” entails the removal of some spikes in one feature-vector due to the presence of the other feature-vector. The detector and “watermarking”-details are given below, the idea of global conjunction detection via (conditional) vector-propagation is depicted in figure 3, with detector outputs denoted as vectors.

### B. Neural data-structure.

The presence of feature like *triangle* is characterized by the distributed activity vector that its presence elicits in the local set of basic neurons. We let these basic neurons each emit at most one, precisely timed spike. The collected spikes of  $n$  neurons then yield a *spike-time vector*:  $\vec{S} = \langle t_1, t_2, \dots, t_n \rangle$ , with  $t_i$  the time of the spike emitted

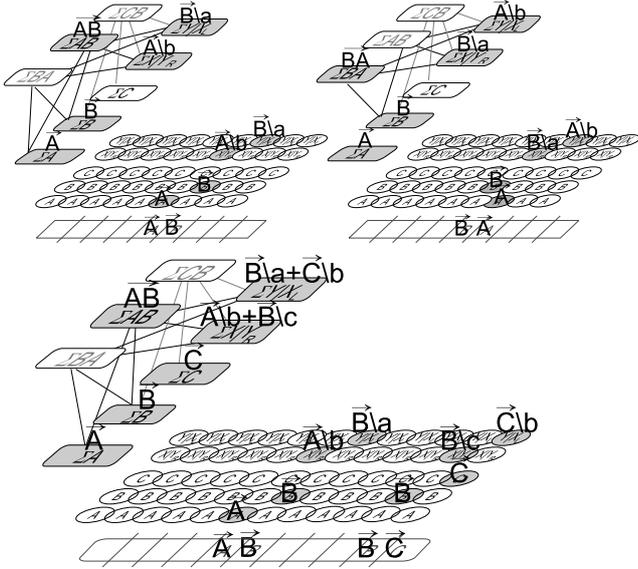


Fig. 3. Vector propagation in a vector-based architecture: correct global conjunction detection.

by neuron  $i$ . Should a neuron emit multiple spikes, then the spike-time vector generalizes to a *spike-train vector*:  $S(\vec{t}) = \langle t_1^i, t_2^i, \dots, t_n^i \rangle$ , where  $t_i^i$  is a vector of spike-times. Detectors in the architecture operate on these spike-train vectors, i.e., this is the neural data-structure.

### C. Local Feature Binding.

The local universal conjunction detectors  $(X|Y)_R$  and  $(Y|X)_L$  in figure 2 perform local universal feature binding, and are the first step in enabling *correct* global detection of feature-conjunctions. These detectors detect and signal “there are two active locations next to each other”. To signal the local conjunction, we adapt the idea of *Context Dependent Thinning* (CDT) as in [5]. In [5], it is observed that the binding of one vector, say  $\vec{A}$ , and another vector, say  $\vec{B}$ , can be signaled by setting part of the active elements (“1’s”) in the vector  $\vec{A}$  to inactive (“0’s”), as a function of  $\vec{B}$ . This *contextually thinned* vector, denoted by  $\vec{A} \setminus b$  is then indicative for the  $AB$  conjunction.

We design a feed-forward CDT procedure using spiking neurons based on *shunting inhibition*, e.g. [7]. A local universal conjunction-detector  $(X|Y)_R^i$  receives as input two spike-time vectors, in our example the spike-time vectors from two consecutive locations,  $i$  and  $i + 1$ . We denote these spike-time vectors with  $\vec{X}$  and  $\vec{Y}$  respectively. The detector determines whether there are sufficient spikes present in  $\vec{X}$  and  $\vec{Y}$  to assume the presence of two features (a conjunction). In that case, it propagates  $\vec{X}$ , with part of  $\vec{X}$  *shunted* by  $\vec{Y}$ . Shunting is defined as follows: a spike in an element  $j$  of  $\vec{Y}$  inhibits the propagation of *later* spikes in a set  $\Gamma_j^i$  of elements in  $\vec{X}$ , where  $\Gamma_j^i$  is fixed in terms of inhibitory connections. With inputs  $\vec{X} = \langle t_1^x, \dots, t_n^x \rangle$  and  $\vec{Y} = \langle t_1^y, \dots, t_n^y \rangle$ , the spike in  $t_i^x$  is propagated if not shunted, i.e. if  $\forall k \in \Gamma_j^i : t_i^x < t_k^y$ . The complementary detector  $(Y|X)_L^i$  shunts  $\vec{Y}$  with  $\vec{X}$ . Importantly, different

thinned spike-time vectors can be superimposed without losing the different vector-patterns, thus alleviating the superposition-catastrophe (up to some point).

### D. Conjunction detection.

A global conjunction detector  $\Sigma AB$  for  $A$ -left-next-to- $B$  ( $AB$ ) consists of an input-layer for detecting correspondence of the input to the conjunction  $AB$ , and an output-layer that propagates the activity in the input-layer if this activity is larger than some threshold (fig 4, dark detector). The input-layer consists of  $N$  ordered elements. Input elements are exclusively connected to the corresponding elements in either the  $\Sigma A$  and  $\Sigma X|Y_R$  detector, or to  $\Sigma B$  and  $\Sigma(Y|X)_L$ . A connection to a pair is made based on the following. When the architecture is presented with  $AB$ , elements are activated in  $\Sigma A$ ,  $\Sigma B$ ,  $\Sigma(X|Y)_L$  and  $\Sigma(Y|X)_R$ . The output vectors of  $\Sigma A$  and  $\Sigma B$  then correspond to respectively  $\vec{A}$  and  $\vec{B}$  (when  $\vec{A}$  signals feature  $A$ , same for  $\vec{B}$ ); due to shunting by  $B$ ,  $\Sigma(X|Y)_R$  contains a particular fraction of the vector  $\vec{A}$ , and vice versa  $\Sigma(Y|X)_L$  a particular fraction of  $\vec{B}$ . Thus, *for these fractions*, corresponding elements  $i$  fire coincidentally in the output vector of both  $\Sigma A$  and  $\Sigma(X|Y)_R$ , or  $\Sigma B$  and  $\Sigma(Y|X)_L$ . An element  $i$  in the input layer of the  $\Sigma AB$  detector is connected to elements  $i$  in a  $\Sigma$ -detector pair, if the pulses from these elements  $i$  would be coincidental when presented with  $AB$ . The threshold for the  $\Sigma AB$  input-elements is set to two coincident pulses, the threshold of the  $\Sigma AB$  output-elements is set to require the activation of all elements in this input-layer that are active if the conjunction  $AB$  is presented. An example of such connectivity is depicted in figure 4A,B. When an active vector contains  $n$  active elements/spikes, the presence of an  $AB$ -conjunction activates  $n/2$  elements in  $\Sigma X|Y_R$  and in  $\Sigma Y|X_L$ , which then activate  $n$  elements in the input-layer of  $AB$  (e.g. 4A). This  $n$ -element pattern is then propagated by the output-layer. For a  $BA$  conjunction however, the active elements in the global universal conjunction-detectors are interchanged, and no input-elements in  $\Sigma AB$  receive synchronous spikes (fig 4B).

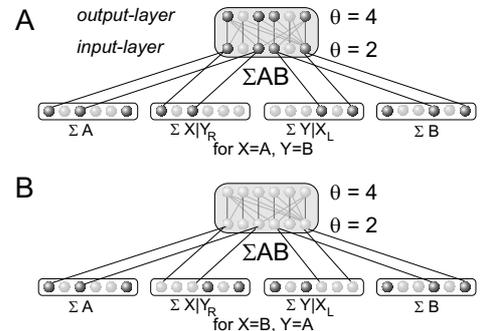


Fig. 4. A) Connectivity for detecting global feature conjunctions. Dark elements are active. B) Active elements in same global conjunction-detector for interchanged feature positions.

With the conjunction  $AB$  present, the presence of additional conjunctions on the input-grid yields additional

spikes in the global conjunction-detectors, but these added spikes do not disturb the  $AB$  spikes that remain present. With the correct features present in non-related conjunctions (“ghosting”), the spike-patterns in the global conjunction-detectors only partially match the  $AB$  pattern, and less input elements are activated in the input layer of  $AB$ : hence the  $AB$  output-elements do not reach threshold and the conjunction-detector is not turned “on”.

These conjunction-detectors can correctly detect the conjunction in the presence of up to some  $M$  other, similar conjunctions on the grid: the presence of conjunctions  $AC$ ,  $AD$ ,  $AE$ , etc. . . , gives rise to the additional superposition of local vectors  $\vec{A}\setminus c$ ,  $\vec{A}\setminus d$ , and  $\vec{A}\setminus e$  in  $\Sigma(X|Y)_L$ , which will increasingly “fill in” the locally shunted spikes in  $\vec{A}\setminus b$  (and similarly for  $\vec{B}\setminus a$  in  $\Sigma(Y|X)_R$ ). The value of  $M$  depends on the amount of shunting, i.e. section IV.

Spiking neurons can learn this connectivity via temporal Hebbian learning rules as in [8]: presented with single  $AB$  conjunction, such a learning rule would translate the synchronous activation of neurons in either of the detector pairs into selectively enhanced weights.

### III. IMPLEMENTATION

In this section, we detail the implementation of the neural detectors outlined in Section II in networks of spiking neurons. Previous research has demonstrated that these neurons operating on timed spikes can perform useful pattern detection tasks [9], [8], [10], [11], [7]. The spiking neurons we use are leaky-integrate-and-fire neurons modeled as Spike Response Neurons [12]. These neurons sum incoming spikes as post-synaptic potentials (PSPs) to calculate an internal variable called “membrane potential”. When this potential reaches a threshold  $\theta$ , a spike is generated and a refractory (negative) response is added to the potential. The time-constant  $\tau$  for the decay of the PSPs is set to 7ms, unless stated otherwise. Connections between neurons each have a single weight and delay.

In our setup, each location on the grid is populated with  $N$  diversely tuned basic neurons. We denote a basic neuron  $j$  by  $S_j$ . When presented with a local feature, some  $n \leq N$  basic neurons emit a spike, generating a spike-time vector  $\vec{S}$  with elements  $s_j$ .

A local feature-detector  $A$  aims to detect the presence of a spike-time vector  $\vec{A}$  in the input  $\vec{S}$ , and then transmit  $\vec{S}$ . The feature detector consists of  $N$  spiking neurons. Each detector neuron,  $A_i$ , receives equally weighted input from all  $n$  active elements in  $\vec{S} = \vec{A}$ , but with a delay such that for  $\vec{S} = \vec{A}$ , all input spikes arrive at the detector-neuron simultaneously, e.g. for an input spike-time  $a_j$ , the delay  $d_j$  between  $S_j$  and  $A_i$  is such that  $a_j + d_j = c_i$ , with  $c_i$  constant for all connections to  $A_i$ . This effectively detects the temporal pattern, e.g. [8]. The constant  $c_i$  is set to  $c_i = c + a_i$ , with  $c$  some constant. The result is that an input vector resembling  $\vec{A}$  is effectively propagated (fig. 7A). Additionally, the  $A$  detector responds in a graded manner to  $A$ -ish vectors, as increasingly different vectors

are propagated with increasing delay and decreasing activity, see also [8], [10].

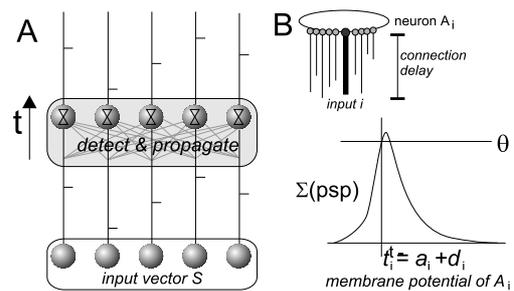


Fig. 5. A) Local feature detector (dark box): each sphere is a spiking neuron, dark neurons are active. Horizontal ticks are timed spikes. B) Top: weights and delays of connections to input. Bottom: time-course of membrane-potential for preferred feature.

A local universal conjunction-detector (fig. 6A) receives input from two consecutive locations. The respective sets of basic neurons are denoted by  $S$  and  $T$ , with activity vectors  $\vec{S}$  and  $\vec{T}$  and neurons  $S_i$  and  $T_i$ . A detector  $(X|Y)_L$  consists of  $N$  spiking neurons  $(X|Y)_i$ , where each neuron is connected to all basic neurons in  $S$  and  $T$ . We set the weights from all inputs  $S_j$  and  $T_j$  to  $(X|Y)_i$  such that the summed activity from  $S$  and  $T$  generates a flat effective potential in  $(X|Y)_i$ . The weight of the connection between  $S_i$  and  $(X|Y)_i$  is set such that the PSP of a spike in  $S_i$  lets  $(X|Y)_i$  reach threshold, but only when both  $S$  and  $T$  contain some  $n$  spikes (indicating the presence of *something*). The flat nature of this potential ensures that the spike-time of  $S_i$  is propagated. The weights from shunting inputs  $T_j \in \Gamma_j^i$  are set such that they strongly inhibit  $(X|Y)_i$  effectively shunting this element (fig. 6B, bottom). The input/output spike-times for a  $(X|Y)_L$  detector for some  $\vec{S}$  and  $\vec{T}$  are shown in figure 7B: most of the non-shunted part of the input spike-time vector is propagated with a fixed delay. However, late spikes are more likely to be shunted, and the relative spike times of early spikes are somewhat delayed due to the still increasing membrane potential early on.

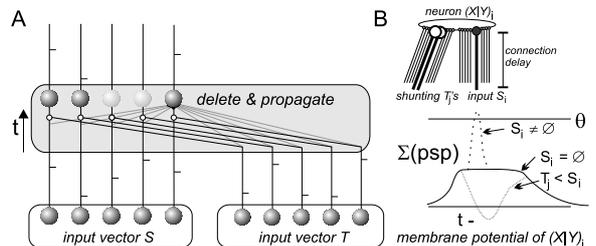


Fig. 6. A) The local universal conjunction detector (dark box) deletes and propagates parts of input  $S$ . B) Top: weights and delays of connections to input. Bottom: time-course of membrane-potential given any two features.

The global feature and universal conjunction-detectors each consist of a single layer of  $N$  spiking neurons. A neuron  $\Sigma_i$  is connected only to corresponding neurons  $i$  in the respective local detectors (fig. 8A), and  $\tau$  is set to 4ms. The threshold is reached by a single spike (fig. 8B).

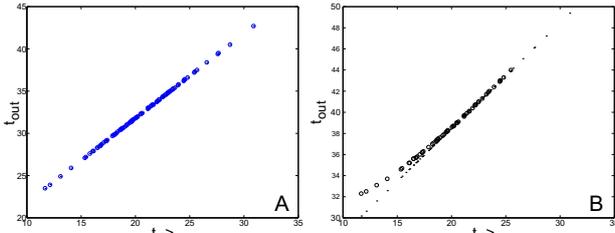


Fig. 7. Vector propagation: input vs. output times. Circles: actual output, dots: delayed input, for: A) local feature detector  $A$ , presented with  $\vec{A}$ , B) local universal conjunction detector, presented with some  $\vec{S}$  and  $\vec{T}$ .

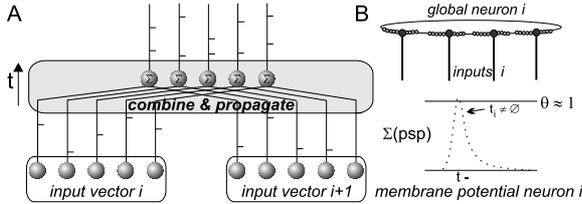


Fig. 8. A) Dark box: global feature or universal conjunction detector. B) Top: weights and delays of connections to input vector. Bottom: time-course of membrane-potential for impinging spike.

A global feature-conjunction detector consists of an input-layer of  $N$  spiking neurons, with  $\tau$  set to 4ms, and an output-layer of  $N$  spiking neurons, with  $\tau$  set to 7ms. The input neurons are connected to global feature and global universal conjunction detectors, as outlined in section II, and fig. 4A. The neurons in the output layer are connected to all neurons in the input-layer, like a feature-detector, and detect the presence of the  $n$  active spikes.

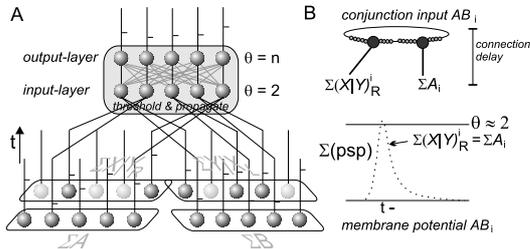


Fig. 9. A) Global conjunction detecting node: dark spheres indicate active neurons, horizontal ticks the timed spikes. B) Top: weights and delays of connections to input vector. Bottom: membrane-potential due to two synchronous impinging spikes.

#### IV. EXPERIMENTS

We implemented the architecture as outlined in section III. We experiment with a conjunction-detector selective for the feature-conjunction *triangle-next-to-square*, i.e. our “ $AB$ ”-conjunction. In the experiments, we place a number of feature-conjunctions on a grid, and we measure the number of activated neurons in the input-layer of  $AB$ . The different feature-conjunctions placed on the grid are shown in figure 10A. Scenes (a) and (b) reflect the uncluttered conjunction-detection problem. Scene  $c$ ) would cause “ghosting” without a special feature-binding operation,  $d$ ),  $e$ ) and  $f$ ) test increasing feature-conjunction clutter without and with the target conjunction present.

In the experiments, the neural input vectors were of length  $N = 500$ , with  $n = 100$  active (spiking) neu-

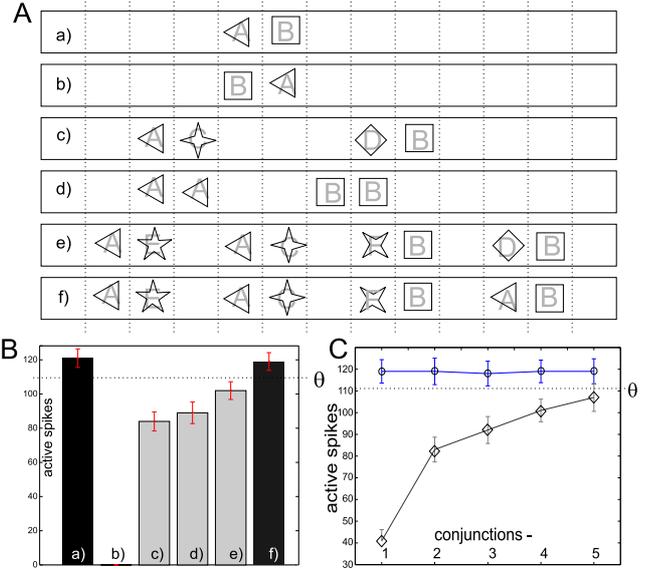


Fig. 10. A) feature-conjunctions on a grid. B) Activation in  $AB$  detector. C) Activation as a function of scene clutter. Error-bars indicate min/max activation in 10 simulations.

rons. Each separate feature was determined by a randomly drawn set of spike-times from a normal distribution, with  $\sigma = 3.5\text{ms}$ . In the local universal conjunction-detectors, approximately half the input spikes were shunted. The experiment was repeated 10 times and the average activations in the input-layer of  $AB$  are shown in figure 10B: there is a clear difference between conjunction-scenes that do not contain the  $AB$ -conjunction, but merely its constituent features, and those that do contain the conjunction. The threshold  $\theta$  for detecting  $AB$  was set to  $\theta \approx n$ , as determined from the uncluttered presentation of  $AB$ .

We performed a systematic comparison between configurations with and without  $AB$  conjunction. Plotted in figure 10C is the  $AB$ -activation for an increasing number of “distracter” conjunctions, with  $AB$  present (circles), and absent (diamonds). Distracters are defined as  $AC$ ,  $DB$ ,  $AE$ ,  $FB$  etc... The difference in activation in the conjunction detector becomes too small to reliably discriminate the two cases for more than 4 conjunctions. A higher “simultaneous representation capacity” can be obtained by increased local shunting, but at the expense of lowering activation in the conjunction detector as compared to the original input vectors. Instead, “attentional” mechanisms could determine a region of interest for a more reliable detection when presented with too many conjunctions.

#### V. DISCUSSION

We chose the example of feature-binding of consecutive shapes on an input grid, to demonstrate how our framework allows a “natural” feed-forward network to perform feature-binding. For this particular example it is especially clear that the extension of the framework to include hierarchical compositional integration enables the efficient recognition of increasingly complex conjunctions in a global, or *position-invariant* manner. The details of

such a framework extend beyond the limited space of this paper, and we refer to the full report [13]. But keeping this goal in mind, we used two complementary CDT-detectors,  $(X|Y)_L$  and  $(Y|X)_R$ , resulting in two “bound” vectors, e.g.  $\vec{A} \setminus b$  and  $\vec{B} \setminus a$ . The combination of these two vectors preserves the initial (average) number of spikes in a feature-vector for processing downstream, that is, if CDT removes about half the active spikes. The intuition is, that if less spikes are removed, the feature-conjunction is signaled weaker; if more are removed, the total number of spikes that can be used downstream decreases. As noted, if CDT shunts about half the spikes, the number of *similar* conjunctions that can be detected simultaneously is about 4 (sections II-D and IV). We note that the human brain seems to perform similarly [6].

The “synchrony hypothesis” [1] has so far been the main theory on dynamic feature-binding, but criticism has been mounting, e.g. [14]. Our architecture has the advantages that it uses an inherently distributed code, enables feed-forward spatial feature-binding, and can be implemented in biologically reasonable spiking neural networks. We also remark that the vector-structure is more a formalization than a spatially localized necessity. The required connectivity only connects similarly tuned neurons from different locations into global neurons. The collective distributed activity of neurons thus connected can be interpreted as a spike-train vector data-structure.

In the experiments, we used biologically reasonable spiking neurons implemented in the Spike-Response Model (SRM). These neurons were used for three reasons: firstly, their ability to emit multiple spikes to implement the global universal conjunction detectors. Secondly, the global feature-conjunction detector uses their ability to detect coincident timing of spike-trains from a pair of global feature and universal conjunction detectors. Thirdly, they can implement the feed-forward CDT procedure via shunting inhibition. Opposed to traditional sigmoidal neurons, all these (different) tasks could be implemented by spiking neurons only differing by threshold and time-constant  $\tau$ . Note though that we had to (implicitly) take into account the limitations of the neurons with regard to temporal precision and limited firing-frequency. To this end we used sparse vectors and presented only a limited number of conjunctions simultaneously. Thus, global neurons only had to superimpose a limited number of spikes, keeping the firing-rate low, and the subsequent coincidence detection was not required to be too precise. Sparse codes are also considered efficient both from an information-theoretical as well as from a metabolic point of view [15].

The timed precision of individual spikes from *real* neurons is the subject of much discussion, though increasingly a remarkable precision of single spike-timings in experiments is reported. We remark that distributed spike-time vectors as we employ are relatively insensitive to (uncorrelated) noise on individual spike-times.

To detect more than 4 *similar* conjunctions simultaneously, say for further integration, a solution would be to

use multiple copies of the same detector. Importantly, this would only apply to detectors for features-conjunctions that are often present in numbers larger than the capacity of our framework. If seen as a biological model, the prediction then is that the density of feature-detectors sensitive to particular conjunctions is proportional to the probability of co-occurrence of multiple such conjunctions. As such, having to allocate and learn multiple detectors for often occurring conjunctions seems quite feasible.

## VI. CONCLUSIONS

In this paper, we have demonstrated how the temporal dimension of individual spikes combined with the introduction of a novel local feature-binding operator can be employed to detect feature-conjunctions from position-invariant (aggregate) feature-detectors, in the presence of other conjunctions. The weights for feature detection were set similar to those obtained with temporal Hebbian learning in [8], [10], suggesting that the architecture could thus be learned. The incorporation of unsupervised learning in the framework is thus a logical addition. As noted in [1], the issue of dynamic binding and structured representations is important in the field of neural networks and (sub)symbolic AI. We believe that as such, the framework developed should enable new ways of dealing with these issues. We are exploring this angle.

## REFERENCES

- [1] Ch. von der Malsburg. The what and why of binding: The modeler’s perspective. *Neuron*, 24:95–104, 1999.
- [2] D.O. Hebb. *The Organization of Behaviour*. Wiley, NY, 1949.
- [3] J.A. Fodor and Z.W. Pylyshyn. Connectionism and cognitive architecture: a critical analysis. *Cognition*, 28:3–71, 1988.
- [4] M.C. Mozer. *The Perception of Multiple Objects*. MIT Press, MA, USA, 1991.
- [5] D.A. Rachkovskij and E.M. Kussul. Binding and normalization of binary sparse distributed representations by context-dependent thinning. *Neural Computation*, 13:411–452, 2001.
- [6] S. J. Luck and E. K. Vogel. The capacity of visual working memory for features and conjunctions. *Nature*, 390:279–81, 1997.
- [7] S.J. Thorpe and J. Gautrais. Rapid visual processing using spike asynchrony. In M.C. Mozer, M.I. Jordan, and Th. Petsche, editors, *NIPS 9*, pages 901–08. MIT Press, 1997.
- [8] T. Natschläger and B. Ruf. Spatial and temporal pattern analysis via spiking neurons. *Network: Comp. Neural Syst.*, 9(3):319–338, 1998.
- [9] J.J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376:33–36, 1995.
- [10] S.M. Bohte, J.N. Kok, and H. La Poutré. Unsupervised classification in a layered network of spiking neurons. *IEEE Trans. Neural Networks*, to appear. An abstract has appeared in the proceedings of IJCNN’2000, vol IV, 249–55.
- [11] S.M. Bohte, J.N. Kok, and H. La Poutré. Spike-prop: error-backpropagation in multi-layer networks of spiking neurons. *Neurocomputing*, to appear. An abstract has appeared in the proceedings of ESANN’2000, pp 419–425.
- [12] W. Maass and C. M. Bishop. *Pulsed Neural Networks*, volume 1. MIT Press, Cambridge, MA, 1999.
- [13] S.M. Bohte, J.N. Kok, and H. La Poutré. Binding with sparse spike-time vectors. in preparation, www.cwi.nl/~sbohte, 2001.
- [14] M.N. Shadlen and J.A. Movshon. Synchrony unbound: A critical evaluation of the temporal binding hypothesis. *Neuron*, 24:67–77, 1999.
- [15] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.