Towards bridging the Gap between Biological and Computational Image Segmentation

Iasonas Kokkinos, Rachid Deriche, Théodore Papadopoulo, Olivier Faugeras, Petros Maragos

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Towards bridging the Gap between Biological and Computational Image Segmentation

Iasonas Kokkinos — Rachid Deriche — Théodore Papadopoulo — Olivier Faugeras — Petros Maragos

N° 6317

Octobre 2007

Thème BIO

ISSN 0249-6399 ISRN INRIA/RR--6317--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Towards bridging the Gap between Biological and Computational Image Segmentation

Iasonas Kokkinos [*] , Rachid Deriche [†] , Théodore Papadopoulo [†] , Olivier Faugeras [†] , Petros Maragos [*]

Thème BIO — Systèmes biologiques
Projet Odyssée

Rapport de recherche n° 6317 — Octobre 2007 — 111 pages

Abstract: This report presents a joint study of biological and computational vision.

First we briefly review the most common models of neurons and neural networks and the function of cells in the V1/V2 areas of the visual cortex. Subsequently, we present the biologically plausible models for image segmentation that have been proposed by Stephen Grossberg and his collaborators during the previous two decades in a series of papers. We have implemented the B.C.S. (*Boundary Contour System*) and F.C.S. (*Feature Contour System*) models that form the basic building blocks of this model of biological vision, known as FACADE (*Form And Colour and DEpth*) theory. During their implementation, we faced several problems, like a large number of parameters and instability with respect to these; this was not traded off with a higher performance when compared to classical computer vision algorithms.

This has led us to propose a simplified version of the B.C.S./F.C.S. system, and to explore the merits of using nonlinear recurrent dynamics. The biologically plausible model we propose is paralleled with classical computational vision techniques, while a link with the variational approach to computer vision is established.

By interpreting the network's function in a probabilistic manner we derive an algorithm for learning the network weights using manually determined segmentations excerpted from the Berkeley database. This facilitates learning the terms involved in the variational criterion that quantifies edge map quality from ground truth data. Using the learned weights our network outperforms classical edge detection algorithms, when evaluated on the Berkeley segmentation benchmark.

[*] Computer Vision, Speech Communication and Signal Processing Group, School of E.C.E., National Technical University of Athens, Iroon Polytexneiou 9, 157 73, Zografou, Athens, {jkokkin,maragos}@cs.ntua.gr

[†] Projet Odyssée, INRIA Sophia-Antipolis, 2004, route des Lucioles, BP 93, 06902 Sophia-Antipolis, France, {Rachid.Deriche,Theodore.Papadopoulo,Olivier.Faugeras}@sophia.inria.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Key-words: Computer Vision,Biological Vision, Neural Networks, Boundary Contour System, Feature Contour System, Line Process, Surface Process, Variational Techniques for Computer Vision, Edge Grouping, Perceptual Grouping, Mean Field Theory, Boltzmann Machines, Learning

Résumé : Ce rapport prsente une étude conjointe de la vision biologique et de la vision algorithmique. Nous nous intéressons plus particulièrement aux modèles biologiquement plausibles liés au processus de segmentation d'images tel que proposé par S. Grossberg et ses collègues.

Dans une première partie rédigée sous forme de tutoriel, nous abordons le problème de la modélisation du comportement et de la dynamique d'un neurone. Nous abordons ensuite le cas plus complexe d'un réseau de neurones avant de nous focaliser plus particulièrement sur la classe des neurones qui interviennent dans le cortex visuel et plus spécifiquement dans les zones corticales V1/V2.

Nous résumons ensuite de manière synthétique les travaux de S. Grossberg et ses collègues sur la modélisation biologiquement plausible du processus de segmentation. La mise en oeuvre de ses modèles B.C.S. (*Boundary Contour System*) et F.C.S. (*Feature Contour System*) qui forment la base du modèle de vision biologique FACADE (*Form And Colour and DEpth*) est présentée et discutée. Nous mettons en lumière certaines difficultés posées par la mise en oeuvre de ces modèles et proposons ensuite quelques modifications qui les simplifient tout en permettant de mieux les controler et d'améliorer sensiblement la qualité des résultats de segmentation obtenus. Le modèle simple et biologiquement plausible de segmentation que nous proposons est ensuite mis en parallèle pour comparaison avec certaines approches classiques proposées en Vision Algorithmique. Un lien avec les approches variationnelles plus récemment introduites en Vision conclut enfin ce rapport illustré par plusieurs exemples de rsultats obtenus sur diverses images réelles.

D'une interprtation probabiliste de la fonction opre par le rseau, une mthode d'apprentissage supervise des coefficients de ce rseau est propose en se basant sur des images segmentes manuellement extraites de la base de donnes d'images segmentes de Berkeley. Cela permet notamment de dterminer partir de donnes terrain les termes du critre variationnel qui quantifie la qualit des cartes de contours. En utilisant ces coefficients appris, notre rseau, appliqu au benchmark de segmentation de Berkeley, se comporte bien mieux que les algorithmes de dtection de contours classiques.

Mots-clés : Vision par Ordinateur, Vision Biologique, Réseaux des Neurones, Boundary Contour System, Feature Contour System, Processus de Lignes, Processus des Surfaces, Approches Variationnelles en Vision par Ordinateur, Groupement Perceptuel.

Contents

1 Introduction

A considerable part of research in the computer vision community has been devoted to problems of low and mid-level vision, and specifically image segmentation, that can be summarized as the problem of breaking an input image into a set of homogeneous pieces. Even though this may seem to be a trivial task for a human, since we perform this task effortlessly, it is an intrinsically difficult one, as 3 decades of research have proven.

Despite the progress that has been made in the last decades, the human visual system outperforms the state-of-the-art in image segmentation, since it is robust to noise, clutter, illumination changes, etc; therefore, its mechanisms could serve as a pool of ideas and a point of reference for computer vision research.

In this report, we shall try to establish a link between systems that have been proposed as modelling the mechanisms of low and mid-level biological vision tasks like image segmentation and some well-established computer vision techniques. Our starting point will be the system developed by S. Grossberg and his collaborators through a series of papers; this system is based on both edge and surface-based information to perform segmentation, and results in the formation of piecewise continuous surfaces. Later on we shall present a recurrent variation of these networks and exploit the link between recurrent networks and variational techniques for image segmentation. Even though these links have been established in previous work [93, 91, 66], our work elaborates on these connections and tries to exploit the interplay between the neural architectures and the computation they perform.

The plan of this presentation is the following: in section 2, we present in a purely tutorial manner some elementary background material on neural computation: the dynamics of a neuron cell, the dynamics of an ensemble of cells and the most common neurons that are involved in the visual pathway are briefly presented. In section 3, we present the architecture proposed by S. Grossberg for the problems of edge detection and image smoothing and our comments on this model. In section 4, we describe a simpler, yet more efficient model, which uses recurrent intra-layer connections and less processing stages and yields better results than the original model when applied to both synthetical and real images. An interpretation of the proposed model in computer vision terms is presented that offers a different perspective on our model, making a link with variational techniques. In Section 5, based on a probabilistic interpretation the network's function we derive an algorithm for learning the network weights using manually determined segmentations. Using the learned weights our network outperforms classical edge detection algorithms; for training and testing we use the Berkeley segmentation dataset. In the last section, we summarize the most important contributions of our work and present possible directions for future research.

In Appendix A, the link between recurrent networks, variational techniques and statistical physics is reviewed. In Appendix B, we prove that the network we proposed decreases a Lyapunov function and in Appendix C some implementation details are given.

One of our objectives has been to make this report self-contained, and accessible to someone who may not have a background in biological vision; therefore a large part of this report serves a tutorial purpose and a well knowledgeable reader can go directly to sections 4&5 and Appendixes B and C to read about the model we propose.

The notation used in this report is as follows:

- Vectors and matrices are denoted by bold capital letters. If we have multiple matrices with similar role but different elements, we shall use superscripts to differentiate among them; e.g. if $\mathbf{W}$ is the kernel of an oriented edge detection filter, $\mathbf{W}^\theta$ will be used to indicate the kernel used for edge detection at direction θ.
- Time varying quantities are written with capital letters. We shall use the ∞ subscript for their steady-state values, in case it is not clear from the context.
- Brackets are used to index with discrete variables $[i, j, k, \ldots]$, parentheses for continuous $(t, v, \ldots)$.
- Constants, parameters etc. are denoted by small letters, usually from the beginning of the alphabet. Elements of constant vectors and matrices (e.g. connection weights) are indexed using subscripts.
- In section 2, subscripts are commonly used to label variables (e.g. V_m: membrane voltage, etc.). Sometimes capital letters will be used for constants, due to convention, e.g. V_{exc} denotes a steady excitation voltage, R a constant resistance.
- Whenever some quantities are involved in identical equations curly braces are used to compactly write these equations; e.g. in (34) we mean that the equation holds for both S^+ and S^-.

2 Neural Computation & Biological Vision Essentials

Our goal in this section is to introduce some widely used concepts and terms from models of neurons, neural networks and biological vision, trying to clarify the distinctions and the similarities between the models presented hereafter. The first subsection deals with the function of an isolated neuron and the way it reacts to an excitatory/inhibitory input, the second subsection examines the behavior of an ensemble of neurons and their collective computational properties; in the last subsection those visual cortex cells that are most commonly deemed important for the purpose of image segmentation are briefly presented.

2.1 Single Cell Dynamics

In this subsection some of the aspects of the dynamics of a single cell will be presented. Starting with the simplest of all models, that of a passive membrane, we will subsequently study the Hodgkin-Huxley equations and how they account for the creation of spikes, and present the two most common approximations to the dynamics of a neuron, namely the integrate-and-fire and the mean firing rate models of a neuron. This presentation is by no means complete; most of the exposition here follows [61] & [17] and the interested reader can find a wealth of material in there.

2.1.1 Single Neuron Terminology

In fig. 1 we see a sketch of a neuron: Many *axons* leave the *soma* (body) of a neuron, and end

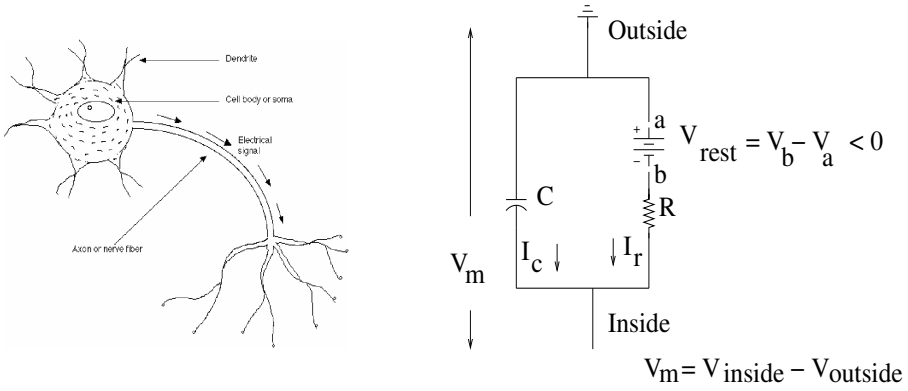


Figure 1: (a) A sketch of a neuron, from `www.cog.brown.edu/courses/cg0001/lectures/visualpaths.html` (b)The passive membrane model of a neuron

up near other neurons, whose state they influence at their *synapses* (contacts) located at the *dendrites* (treelets) of the 'receiver' neuron. The 'emitter' neuron is called *presynaptic* and the 'receiver' is called *postsynaptic*. The state of a neuron can be observed by its potential, which is negative ($\simeq -70mV$) when the neuron is in isolation, i.e. when it receives no input from other neurons. This is achieved by a complicated mechanism that constantly pumps *ions* (charged elements, e.g. K^+ - Potassium, Na^+ - Sodium, Cl^- - Chloride) in and out of the neuron's soma through *channels* located at the neuron's *membrane*, keeping its potential negative in equilibrium. The communication between neurons is effectuated when the presynaptic neuron generates a *spike*, i.e. rapidly increases and then lowers its potential, which travels along its axons; when this spike arrives at the synapses, it results in the release of a chemical, termed *neurotransmitter*. This causes a change in the postsynaptic neuron's conductance to certain ions, which in turn results in an increase, or decrease in the postsynaptic neuron's potential, depending on the neurotransmitter released. We say that the synapse is *excitatory* in the first case and that the postsynaptic neuron is *depolarized* and that the synapse is *inhibitory* in the second case and the cell is *hyperpolarized*.

Models of increased sophistication have been introduced for neurons and their interactions. A common mathematical model that is used in all of them is that of an electrical circuit, due to both its suitability for modeling a neuron (which is a cell with varying potential) and the relative ease with which the behavior of an electrical circuit can be understood.

2.1.2 The Passive Membrane Model

The passive membrane is probably the simplest model of a neuron, shown in fig. 1(b). The fact that a neuron in steady state is charged negatively inside relative to its outside is modeled by an electrical circuit with a capacitor C parallel to a voltage source, equal to the resting potential V_{rest}. The capacitor models the neuron's membrane that is charged by the voltage source. Ionic currents across the neuron membrane surface are modeled by a resistor in series with the voltage source, which accounts for the increase/decrease in the ionic currents due to a change in the neuron's potential. At steady state $I_r = 0 \rightarrow V_m = V_{rest} \simeq -70mV$.

Using this simple RC circuit for a neuron injecting a current into the neuron can be modeled by a current source parallel to the other network elements, as seen in fig. 2. A current

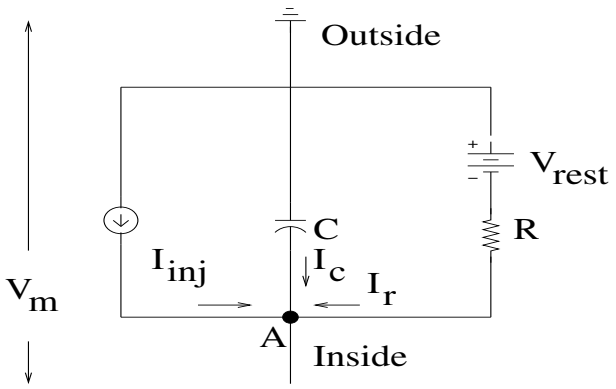


Figure 2: Injecting current into a neuron

that results in an increase in the membrane potential (i.e. the membrane is depolarized) is called by convention a positive current and, conversely, a negative current hyperpolarizes the membrane. Using the passive membrane model and by applying Kirchoff's current law at node A in fig. 2, we find the following equation for the membrane potential:

$$\begin{aligned} -I_C - I_R &= I_{inj} \rightarrow \\ C\frac{\mathrm{d}V_m}{\mathrm{d}t} + \frac{V_m - V_{rest}}{R} &= I_{inj} \rightarrow \\ \tau\frac{\mathrm{d}V'}{\mathrm{d}t} &= -V' + RI_{inj}, \\ \text{where} \quad V' &= V_m - V_{rest}, \quad \tau = RC \end{aligned} \tag{1}$$

where $V_{rest} = -70mV$. [1] Equation (1) is known as the *membrane equation* and can be analyzed using linear system analysis techniques like the Impulse/Step Response and

[1] The polarity of the source is shown in the diagrams for illustration purposes; as above, in all of the following equations, we shall write down the equations for the electrical circuits as if all the voltage sources

Fourier Analysis. Specifically, if we consider a step current applied to the system at $t = 0$, i.e. $I_{inj} = I_0 H(t)$ where H is the step (or Heavyside) function:

$$H(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

then by inspection the response of the system (the membrane potential) can be seen to equal

$$V_m(t) = V_{rest} + RI_0(1 - e^{-t/\tau}), \quad t \geq 0$$

The potential of the membrane asymptotically reaches the value $V_{rest} + RI_0$ with a speed that decreases as RC increases. In general, the response of the system to an input $I(t)$ equals

$$V_m(t) = V_{rest} + \int_{-\infty}^{t} I(u)h(t-u)\mathrm{d}u \quad \text{where} \quad h(t) = \frac{1}{C}e^{-t/\tau}$$

i.e. $h(t)$ is the impulse response of the system. Analysis of the system in the frequency domain reveals that the system acts as a low pass filter, since its frequency response is

$$\tilde{V}(f) = \frac{R\tilde{I}_{inj}}{1 + i2\pi f\tau} \quad \rightarrow \quad \|\tilde{V}(f)\| = \frac{R\tilde{I}_{inj}}{\sqrt{(1 + (2\pi f\tau)^2)}}$$

This is the kind of behavior we were expecting to see from a system like that of fig. 2: rapid fluctuations in the injected current I_{inj} are not fully 'followed after' by V_m, due to RC circuit, so the membrane potential is determined by the low frequency components of the injected current.

Synaptic Excitation and Inhibition of a Passive Membrane

Our analysis up to now was concerned with the way a neuron would respond to an injected current, assuming there is a single current through the membrane, which is modeled by a voltage source in series with a resistance. This ionic current's role is to bring the membrane potential back to its steady state value and is therefore termed a *leakage* current. Apart from this current there are however other ions that are being constantly pumped in and out of a neuron. The difference in the concentrations inside/outside the neuron of these ions determines the potential of the neuron and any neural mechanism that changes the neuron's potential using chemical synapses achieves this by changing the membrane's permeability to these ions. This is achieved at neuron synapses, and therefore this type of excitation/inhibition is called *synaptic*. In an electrical circuit this is modeled by adding voltage sources in parallel to the circuit, in series with modifiable conductances, as shown in fig. 3. The equations for the membrane potential that are derived by Kirchhoff's current

had their positive pole looking into the inner side of the neuron and use negative voltage values for sources of different polarity (e.g. $V_{rest} \simeq -70mV$, here). This is a commonly used convention in electrical circuit models of neurons.

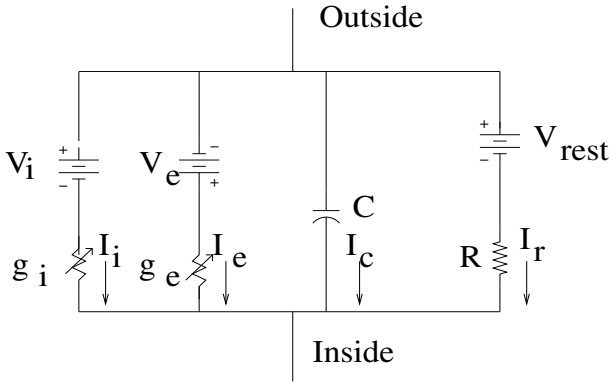


Figure 3: Synaptic excitation and inhibition of a passive membrane.

law are:

$$\begin{aligned} -I_c &= I_i + I_e + I_r \rightarrow \\ C\frac{\mathrm{d}V_m}{\mathrm{d}t} &= g_e(V_{exc} - V_m) + g_i(V_{inh} - V_m) + (V_{rest} - V_m)/R \qquad (2) \\ &= -gV_m + I_d, \quad \text{where} \qquad (3) \\ g &= g_e + g_i + 1/R, I_d = g_e V_{exc} + g_i V_{inh} + V_{rest}/R \end{aligned}$$

where $V_{exc} \geq V_{rest}$, $V_{inh} \leq V_{rest}$. g_e and g_i model the permeability of the neuron to specific ions which determine whether the neuron will be hyperpolarized or depolarized. Note that for steady g and I_d (3) is equivalent to (1), after a change of constants. I_d is called the *driving current* and is independent of V. The excitatory input delivered from neighboring neurons can be modeled by an increase of g_e, so that $(V_{exc} - V_m)$ plays a greater role in the evolution of V_m in (2); conversely the negative input is modeled by an increase of g_i.

A special case is when $V_{inh} = V_{rest}$, or, more generally, when there is another variable conductance g_{sh}, with corresponding $V_{sh} = V_{rest}$, so that equation (2) is rewritten as [12]:

$$\begin{aligned} C\frac{\mathrm{d}V_m}{\mathrm{d}t} &= g_e(V_{exc} - V_m) + g_i(V_{inh} - V_m) + g_{sh}(V_{sh} - V_m) + (V_{rest} - V_m)/R \qquad (4) \\ &= -gV_m + I_d, \quad \text{where} \\ g &= g_e + g_i + g_{sh} + 1/R, I_d = g_e V_{exc} + g_i V_{inh} + V_{rest}(g_{sh} + 1/R) \end{aligned}$$

In that case the dynamics in (4) due to the term $g_{sh}(V_{sh} - V_m)$ are called *shunting inhibition* dynamics. A distinctive feature of shunting inhibition is that its effect cannot be seen in the absence of excitatory input; however high g_{sh} may become, we have $g_{sh}(V_{sh} - V_m) = 0$, so there can be no inhibition. On the contrary, *hyperpolarizing inhibition* acting by $g_i(V_{inh} - V_m)$ can decrease the potential of a neuron in the absence of excitation. Intuitively shunting inhibition acts by absorbing the potential that would otherwise be developed by

the neuron. To clarify the following formulae, assume $V'_m = V_m - V_{rest}$ or, equivalently, that $V_{rest} = 0$. If excitatory input is delivered to the cell by increasing g_e, the effect of shunting inhibition is to decrease the steady-state voltage (found by setting $\frac{\mathrm{d}V'_m}{\mathrm{d}t} = 0$) from $V' = \frac{g_e V_{exc}}{g_e + 1/R}$ to $V' = \frac{g_e V_{exc}}{g_e + g_{sh} + 1/R}$. The effect of shunting inhibition is approximately the *division* of a neuron's potential by g_{sh}. This has been proposed as a mechanism for divisive normalization that can explain many phenomena in the visual system [12, 13] and explains its capability to operate in a wide range of scales without saturating. Some recent objections [51] brought up the problem that the *steady-state voltage* of a neuron is what is being divided and not it's *firing rate* which is the most commonly used output of a neuron. According to [51], the effect of shunting inhibition on the firing rate is subtractive rather than divisive; keeping this in mind, we will still use divisive normalization in our applications, whatever the biophysical mechanism that performs it, and use shunting inhibition as a 'mathematical trick' to achieve divisive normalization using continuous evolution equations like (2).

2.1.3 Hodgkin & Huxley Model

Up to now an important simplification that has been made was that the ionic currents through the neuron membrane are linearly dependent on the difference between the equilibrium potential V_{rest} and the membrane potential V_m. This was modelled by using time- and voltage- *independent* resistances R and conductances g_e, g_i, and results in a simple linear behavior. However, true neurons exhibit more complex behavior, like the emission of short voltage pulses when presented with a strong enough input, known as *spikes* or *action potentials*. The model of Hodgkin and Huxley explains the generation of spikes, using a set of coupled O.D.E.s that capture much more accurately the behavior of a neuron. Explaining in detail these equations is out of the scope of this report, but we think it would be useful to simply explain how a spike can be generated.

The model of Hodgkin and Huxley [50] explains the generation of a spike by analyzing the temporal variation of the permeability of the membrane's conductance to the two ions K^+ and Na^+. The permeability of the neuron's membrane to these two ions is voltage dependent, so a change in the neurons potential will influence the ionic currents, which in turn influence the neuron's potential and so on. Specifically, the inward current that results in the depolarization of the neuron is given by

$$I_{Na}(t) = \bar{g}_{Na} m^3 h (V_m(t) - E_{Na})$$

the outward current that hyperpolarizes the neuron is given by

$$I_K(t) = \bar{g}_K n^4 (V_m(t) - E_K)$$

while there is also a leakage current given by

$$I_L(t) = \bar{g}_L (V_m(t) - E_L)$$

The evolution of the membrane potential V_m is then given by

$$C\frac{\mathrm{d}V_m}{\mathrm{d}t} = -\bar{g}_{Na} m^3 h (V_m(t) - E_{Na}) - \bar{g}_K n^4 (V_m(t) - E_K) - \bar{g}_L (V_m(t) - E_L) + I_{inj} \qquad (5)$$

In the above equations, m, h, n are voltage dependent quantities corresponding to the m, h, n *gating particles* that were introduced by Hodgkin and Huxley and $\bar{g}_{\{Na,K,L\}}$ are constants, equal to the maximal attainable conductances of the neuron for each ion. I_L is used to make up for what the K, Na ionic currents do not model, and $\bar{g}_L, E_L$ are determined so that the model has a prescribed behavior. I_{inj} represents all the input that is delivered to the neuron, whether synaptic or injected, from its environment

The gating particles are random variables, taking the values 0 and 1 (closed/open gate); all the gating particles corresponding to an ionic current must be open (i.e. 3 m-particles and 1 h-particle for Na ions and 4 n-particles for K ions) if the corresponding conductance is to be open. Interpreting the m, h, n quantities as the probabilities of the corresponding gating particle to be open, then the above equation can be explained as saying that on average, the ionic current $I_{\{K,Na\}}$ will be equal to $\bar{g}_{\{K,Na\}}(V_m - E_{\{K,Na\}})$ times the probability of finding the conductance $\bar{g}_{\{K,Na\}}$ open, i.e. n^4 / m^3h respectively.

The *activation variables* m, n are increasing functions of the depolarization $(V_m - V_{rest})$ while the *inactivation variable* h is a decreasing function of it. The dynamics of n can be described by either of the following, equivalent, equations

$$\begin{aligned}
\frac{\mathrm{d}n}{\mathrm{d}t}(V_m) &= \alpha_n(V_m)(1-n) - \beta_n(V_m)n \quad \leftrightarrow \\
\tau_n \frac{\mathrm{d}n}{\mathrm{d}t}(V_m) &= n_\infty - n, \quad \tau_n(V_m) = \frac{1}{\alpha_n(V_m) + \beta_n(V_m)}, \quad n_\infty(V_m) = \frac{\alpha_n(V_m)}{\alpha_n(V_m) + \beta_n(V_m)}
\end{aligned}$$

The same equations are used for m, h and the difference in their behavior lies in the α, β terms. The functions that were empirically derived by Hodgkin and Huxley are [61]:

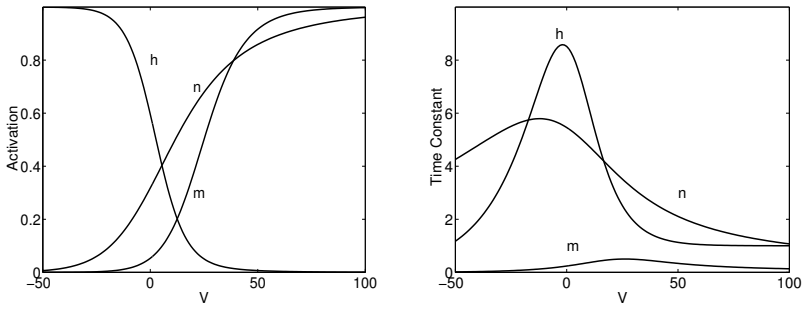
$$\begin{array}{llll}
\alpha_n(V_m) = \frac{10-V_m}{100(e^{(10-V_m)/10}-1)}, & \beta_n(V_m) = .125e^{-V_m/80}, \\
\alpha_m(V_m) = \frac{25-V_m}{10(e^{(25-V_m)/10}-1)}, & \beta_m(V_m) = 4e^{-V_m/18}, \\
\alpha_h(V_m) = .07e^{-V_m/20}, & \beta_h(V_m) = \frac{1}{(e^{(30-V_m)/10}+1)}
\end{array}$$

The resulting graphs of $\{n, m, h\}_\infty$ and $\tau_{\{n,m,h\}}$ as functions of V_m are shown in fig. 4, where we observe that:

- n_∞ is an increasing function of V_m; the steady state value n_∞ is reached with a relatively slow speed.
- h_∞ is a decreasing function of V_m, as we would expect for an inactivation particle, while the steady state value is reached with a relatively slow speed, slower than the n speed below a certain voltage and faster after that voltage.
- m_{infty} is an increasing function of V_m, and the steady-state is reached rapidly, with a time constant that is about 1/10 th of the time constants of n, h

Given the above, if a relatively faint and short current pulse I_{inj} is injected into the neuron, the behavior of the model will be:

- I_{inj} causes an increase in membrane potential and the m particles open faster than the n and h particles while for a short time n and h can be considered stationary. This



(a) (b)

Figure 4: (a) The steady state values and (b) the time constants as functions of V_m for the three gating particles

results in an increase in $g_{Na}m^3h$ and therefore an increase in the inward, depolarizing, current I_{Na}.

- The membrane potential increases (due to I_{Na}) which results in an increase in the outward current $I_K = \bar{g}_K n^4(V_m - E_K)$. This does not have to do with $\bar{g}_K n^4$, which is supposed to be stationary, but with the increase of $V_m - E_K$.
- The two currents I_{Na}, I_K, quickly balance each other, bringing the neuron back to its resting potential

In summary, the injected current pulse results in a short-time and short-magnitude increase in the neuron's potential, like the passive membrane model would predict (even though not in the same way). In case a stronger input current pulse is injected to the neuron, the time course of the events is the following:

- The m particles rapidly open, letting a depolarizing current I_{Na} in.
- n, h stay initially approximately stationary, so the hyperpolarizing current $I_K = \bar{g}_K(V_m - E)$ will increase only due to the increase in V_m.
- I_{inj} is assumed to be strong enough so that the increase in I_K does not immediately outweigh the increase in I_{Na} induced by I_{inj}. This further depolarizes the neuron, which results in a further increase in m. This results in a 'loop' between the previous steps and is witnessed as a rapid increase in V_m.
- After a period of time h will have decreased enough to block the I_{Na} current (due to the m^3h factor) and n will have increased enough to let the I_K current increase (due to the n^4 factor) and bring the membrane back to its resting state. Actually, blocking the I_{Na} current results for a short time in the hyperpolarization of the membrane, which is brought to its resting potential shortly afterwards.

Summarizing, this succession of events results in a rapid increase in the neuron's potential, that describes the generation of a spike, i.e. the stereotypical rapid increase of the neuron potential to a fixed value and the subsequent decrease. This procedure is independent of the magnitude or duration of I_{inj}, provided it is strong enough to initiate the pulse generation process -or, citing [56], 'how fast the bullet travels has nothing to do with how hard you pull the trigger'.

This analysis can be used to understand the transformation of an input stimulus to a firing rate: suppose a constant current is injected into the neuron, say I_{inj} - up to now we dealt with a current pulse. As long as the potential of the neuron is below a certain threshold, called the spike initiation threshold, V_{th}, I_{inj} drives V_m to a steady value, in a way similar to that described for the passive membrane model. If V_m exceeds V_{th} a spike will be produced and V_m will be reset to a value V_0. Then this process starts again, producing another spike, resulting in a cycle with a period equal to the time it takes to bring V_0 to V_{th}. Apart from all the other, neuron-specific parameters, this period depends on the intensity of I_{inj}. A strong current drives V_m quickly to V_{th} -i.e. it results in a high firing rate, and conversely, for a faint current it will take longer to reach V_{th}. If I_{inj} is below a certain threshold the neuron acts like in the passive membrane model: the steady-state potential, which is below V_{th}, is reached after a transient period of time, and no spike is generated.

2.1.4 Integrate-and-Fire Models

Even though the model of Hodgkin-Huxley has helped deeply understand the behavior of a neuron, more economical and straightforward models are desired, both for the purpose of analysis and simulation. A widely used model, that mimics the generation of pulses, as well as the sub-threshold behavior of the Hodgkin-Huxley model is the Integrate-and-Fire (IaF for short henceforth) model of a neuron, that is widely used in neuronal modeling.

The simplest model is the *perfect*, or *non-leaky* Integrate-and-Fire unit, while a more realistic model is the *leaky* Integrate-and-Fire unit. For simplicity, we shall no longer use the resting potential voltage source, V_{rest}, assuming all the voltage sources involved in the circuits have been appropriately modified to account for this change. The equation describing the subthreshold behavior of a perfect IaF unit is:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I_{inj}$$

A spike is generated when $V = V_{th}$, whereupon V is reset to 0. The time instants t_1, t_2 when two consecutive spikes occur are related by:

$$\int_{t_1}^{t_2} I(t)\mathrm{d}t = CV_{th}$$

The resulting spiking frequency for a constant injected current I_{inj} can thus be found as:

$$f_{spike} = 1/\tau_{spike} = \frac{1}{CV_{th}/I_{inj}} = \frac{I_{inj}}{CV_{th}} \tag{6}$$

This model is rather unrealistic, since the firing rate of a neuron cannot become arbitrarily large, as equation (6) would suggest and it is also known that I_{inj} should surpass a certain threshold to initiate a spike. In the more realistic setting of the leaky IaF model, the subthreshold behavior of the model is described by:

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{V(t)}{R} + I(t)$$

which has the solution:

$$V(t) = IR(1 - e^{-t/\tau}) + V(0)e^{-t/\tau} \tag{7}$$

If $V(t)$ exceeds V_{th} a spike is generated, and V is reset to 0; however if $I < V_{th}/R$, no spike is generated, as would be desired for a weak enough current. As before, we have for the time between two consecutive spikes occurring at t_1, t_2, that:

$$V(t_2) = V_{th}, \quad V(t_1) = 0 \overset{(7)}{\rightarrow} T_{th} = t_2 - t_1 = -\tau \log(1 - \frac{V_{th}}{IR})$$

Based on the Hodgkin-Huxley model, there is a refractory period after each spike, during which the cell ignores all input it receives; we can account for this as well, taking the interspike interval to equal $\tau_{ref} + T_{th}$ which results in the following relation:

$$f = \frac{1}{\tau_{ref} - \tau \log(1 - \frac{V_{th}}{IR})}$$

This model, even though slightly more complicated than the passive membrane model and much less complicated than the Hodgkin- Huxley model can account for the generation of spikes, and for the transformation of an input stimulus to a sequence of spikes. Other variants have been introduced (see e.g. [61] and references therein) to allow the model to account for more phenomena.

2.1.5 Mean Firing Rate Models

The most popular model for the analysis of an ensemble of neurons, is the Mean Firing Rate (MFR) model; this model was one of the very first to be used by people in the neural networks community, a long time before the model of Hodgkin and Huxley was presented. According to this model, the neuron's output can be interpreted as an analog value, that is coded by a 'digital' sequence of spikes. As the model's name implies we ignore the specific timing of the spikes that are emitted from a neuron, and replace it by a mean firing rate; this mean can be interpreted as the mean over a population of neurons satisfying the same dynamics.

Given that the spike generation mechanism is ignored in the MFR model, a common choice for describing the neuron's state is the *generator potential* which is the voltage V that would be developed by a passive membrane, if the spike generation mechanism could be blocked. The firing rate U of the neuron can then be related to its generator potential V by a function of the form $U = g(V)$; some common choices for g are:

- A quadratic-above-zero function, used in [12]

$$g(V) = \max(V, 0)^2$$

- A clipping function, with low/high thresholds T_L/T_H, as used in [68]

$$g(V) = \begin{cases} 0, & V < T_L \\ V - T_L, & T_L < V < T_H \\ T_H - T_L, & V > T_H \end{cases}$$

- A sigmoid function, with slope parameter $\beta > 0$, used e.g. in [53]

$$g(V) = \frac{1}{1 + e^{-2\beta V}}$$

Generally such functions are increasing, positive above a certain point, modeling the necessity to surpass a threshold in order to have a positive firing rate and equal to zero below that point, since there cannot be a negative firing rate.

Since the MFR model is commonly used for networks of neurons, one needs to model also the interactions of neurons; a common assumption is that the effect of a presynaptic neuron with firing rate equal to U can be modeled as injecting a current $I_{inj} = w \cdot U$ into the postsynaptic neuron, where w is the synaptic strength - negative w stands for an inhibitory synapse and positive for an excitatory. It is also assumed that there are no interactions among the synapses and that the total current delivered to the neuron is the sum of the individual currents. This yields the following expression for the membrane potential V of the postsynaptic neuron, as a function of the presynaptic neuron firing rates U_n, $n = 1, \ldots, N$ and membrane potentials V_n:

$$\begin{aligned} C\frac{\mathrm{d}V)}{\mathrm{d}t} &= -\frac{V}{R} + \sum_{n=1}^{N} w_n U_n \\ &= -\frac{V}{R} + \sum_{n=1}^{N} w_n g(V_n) \\ \text{where } U &= g(V) \end{aligned} \tag{8}$$

This is the commonly known *additive model* of a neuron and is widely used in the neural networks community.

Another variant of the MFR model commonly used by S. Grossberg and his collaborators in modeling the dynamics of an ensemble of cells uses synaptic excitation/inhibition to determine the generator potential of a neuron instead of the simple injected-current model. Specifically, the voltage V of the membrane is given by:

$$\tau\frac{\mathrm{d}V}{\mathrm{d}t} = -V g_{leak} + (V_e - V)g_e + (V_i - V)g_i \tag{9}$$

$$\text{where} \qquad g_e = \sum_{n=1}^{N} w_n^e U_n, \quad g_i = \sum_{n=1}^{N} w_n^i U_n \quad U_n = g(V_n) \Rightarrow$$

$$\tau \frac{\mathrm{d}V}{\mathrm{d}t} = -V g_{leak} + (V_e - V) \sum_{n=1}^{N} w_n^e U_n + (V_i - V) \sum_{n=1}^{N} w_n^i U_n \tag{10}$$

In the above equations, V_e, V_i are respectively the maximal/minimal attainable values for V, while g_e, g_i, model the neuron's variable permeabilities to various ions. These are assumed to be determined by the excitatory/inhibitory input the neuron receives from its neighbors $U_{1\ldots N}$ as in (10) using w^i, w^e to model the effect of the activity U_n of each neighboring neuron on the neuron permeabilities. g_{leak} is used to model the leakage current of the neuron.

A more elaborate model is that used in [12], where a separate term $g_{sh} = \sum_{i=1}^{N} w_n^{sh} g(V_n)$ for shunting inhibition is introduced, that acts in a complementary way to hyperpolarizing inhibition. In that case the evolution equations become

$$\tau \frac{\mathrm{d}V}{\mathrm{d}t} = -V g_{leak} - V \sum_{i=1}^{N} w_n^{sh} g(V_n) + (V_e - V) \sum_{n=1}^{N} w_n^e g(V_n) + (V_i - V) \sum_{n=1}^{N} w_n^i g(V_n) \tag{11}$$

Equations (10) and (11) have the implication that V cannot surpass V_e or V_i, while in the additive model V can become arbitrarily large, depending on the r.h.s. of (8). Using (8), one can model divisive normalization, as was previously discussed, which allows a network of neurons to function in a wide range of input stimuli without any of its neurons saturating. A more extended account of the properties of equation (10) is given in [26], §21-24.

A note is necessary about the possible misinterpretation of the above equation as being somehow related to the Hodgkin-Huxley equation (5) since they both somehow describe networks, use some conductances etc. and are mathematically similar; this similarity is superficial, since equation (5) models the membrane current (modeled by V_m/R in (9)) while the latter models the effect of external (synaptic) input to the neuron's potential V. Equation (9) describes the evolution of a *non-spiking* neuron's generator potential in response to the input it receives, which is transformed into a *firing rate* by some function g; no spike generation is modeled, so these are totally different equations. Apart from that, they are mathematically different as well, since the coefficients in (9) are not voltage independent.

Despite its common application, the potential-based model has its shortcomings, since it performs a low-pass filtering of its input as equation (8) shows. It has been argued that a more realistic model can be derived using a neuron's synaptic current as the cause of spike generation, rather than its voltage, which is a low-passed version of the former. The above considerations lead to the use of the synaptic current model presented in [17] which offers an alternative to the classical potential based MFR models.

2.2 Network Dynamics

The next step after modeling the behavior of a single neuron is understanding the behavior of a system of neurons, that form a neural network; in this section we will briefly review some common neural network models. Again, the purpose of this section is tutorial; for the interested reader, an excellent textbook on neural dynamics is [47] while a wealth of information about biological aspects of neural computation can be found in [17]. A useful resource has been the web-page [85] and the references therein.

The most common single-neuron model used in the neural networks community is the additive variety of mean firing rate neurons, which leads to mathematically tractable models. Even though this model greatly facilitates analysis, it throws away any neuron timing information [25], that has been proposed as being crucial to the solution of many vision-related problems related to the binding problem which is at the heart of the segmentation problem.

For the sake of clarity, we summarize the symbols that will be used in the following equations:

U_i:	output of neuron i, to be interpreted as mean firing rate of neuron
V_i:	internal state of neuron i (generator potential)
I_i:	input from other layers that is independent of the neuron's state and outputs
$w_{i,j}$:	synaptic connection strength between neurons N_i and N_j; a negative value stands for inhibition of N_i by N_j and a positive value for excitation of N_i by N_j.
g(U):	the function transforming a neuron's potential into a firing rate.

Matrix notation will be sometimes used for convenience, e.g.:

$$\mathbf{U} = \mathbf{W} \cdot \mathbf{V} \quad \leftrightarrow U_i = \sum_{n=1}^{N} w_{n,i} V_n, \quad i = 1, \ldots, N$$

When expressing $\mathbf{U}$ as a function of $\mathbf{V}$ we shall write $\mathbf{U} = \mathbf{g}(\mathbf{V})$ meaning $U_i = g(V_i), i = 1 \ldots, N$.

2.2.1 Feedforward Dynamics

The simplest type of dynamics occurs when the network operates in a purely feedforward manner, with lower layers sending their output to higher layers, and no connections in the opposite direction, shown schematically in fig. 5(a). This structure is the one employed by many neural networks models for pattern recognition like Multi-Layer Perceptrons and Radial Basis Functions. In this case, the network dynamics are given by the following type of equations:

$$\begin{aligned} \tau \frac{\mathrm{d}V_i^{l+1}}{\mathrm{d}t} &= -V_i^{l+1} + \sum_{j=1}^{N} w_{i,j} g(V_j^l) \rightarrow \\ V_{i,\infty}^{l+1} &= \sum_{j=1}^{N} w_{i,j} g(V_j^l), \quad U_i^{l+1} = g(V_i^{l+1}) \end{aligned}$$

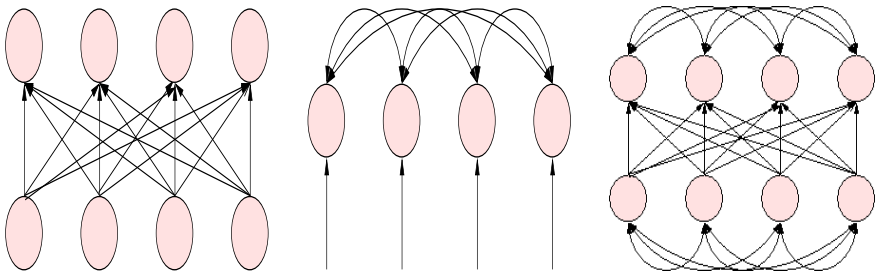


Figure 5: (a) Feedforward network (b) Single and (c) Multiple Layer recurrent networks

where $V^l_{1\ldots N}$ are the activations of layer-l neurons. The dynamics of such networks are restricted and we can use immediately the steady-state value, due to the decoupling of the network interactions in layers. Most of the research effort in the field of feedforward networks has focused on learning the weights of the connections between the networks in order to minimize some classification performance criterion, while their dynamical aspects are ignored. These networks have reduced computational power, compared to their recurrent counterparts.

2.2.2 Recurrent dynamics

A more interesting case arises in networks where there are connections among neurons located at the same processing stage as shown in fig. 5(b),(c), which results in *recurrent* dynamics. There the system as a whole behaves in ways that are global compared to the local nature of the computations that are performed on each network node. Such connections, termed *horizontal* or *lateral* are common in the cortex, and a variety of functions has been proposed as being implemented by such connections.

Linear dynamics

Assume that each neuron sums the input it receives from all of its neighbors, and produces its output as a linear function of this sum. In this case there is decreased biological plausibility, since the outputs can be negative, or tend to infinity, depending on the inputs a neuron receives; however, it is important to see how a system can behave in a *global* way, by performing *local* computations. From that perspective, we should use the term 'computational unit' instead of neuron, in order to avoid confusions.

The dynamics of such a single layer network can be described by a system of linear O.D.E.s:

$$\tau \frac{\mathrm{d}V_i}{\mathrm{d}t} = -V_i + \sum_{j=1}^{N} w_{i,j} U_j + I_i \overset{\text{g:linear}}{=} -V_i + \sum_{j=1}^{N} w_{i,j} V_j + I_i, \quad i = 1, \ldots N \tag{12}$$

where I_i is external input from another layer and is considered constant.

In case the $N \times N$ connection matrix $\mathbf{W} = (w_{ij})$ has N (distinct) eigenvalues λ_i and eigenvectors $\mathbf{E}_i$, these form a full basis on which the vector $\mathbf{V}(t)$ can be expressed as

$$\mathbf{V}(t) = \sum_{i=1}^{N} C_i(t)\mathbf{E}_i$$

Substituting into equation (12), we get a set of decoupled O.D.E.s for the expansion coefficients $C_i(t)$:

$$\begin{aligned}
\sum_{i=1}^{N} \frac{\mathrm{d}C_i(t)}{\mathrm{d}t}\mathbf{E}_i + \sum_{i=1}^{N} C_i(t)\mathbf{E}_i &= \mathbf{W}\sum_{i=1}^{N} C_i(t)\mathbf{E}_i + \mathbf{I} \\
&= \sum_{i=1}^{N} \lambda_i C_i(t)\mathbf{E}_i + \mathbf{I} \overset{\mathbf{E}_i\mathbf{E}_j=\delta i,j}{\rightarrow} \\
\frac{\mathrm{d}C_i(t)}{\mathrm{d}t} + C_i(t) &= \lambda_i C_i(t) + \mathbf{E}_i^T\mathbf{I} \quad i = 1, \ldots, N
\end{aligned}$$

$$\Rightarrow C_i(t) = \frac{\mathbf{E}_i^T\mathbf{I}}{1-\lambda_i}\left(1 - \exp\left(-\frac{t(1-\lambda_i)}{1}\right)\right) + C_i(0)\exp\left(-\frac{t(1-\lambda_i)}{1}\right) \tag{13}$$

Each of these equations describes the evolution of $\mathbf{V}$ along one direction (the one defined by the eigenvector $\mathbf{E}_i$), $C_i(t)$. The evolution of C_i depends on $\lambda_i - 1$: a negative value results in $C_i(t)$ converging from any initial condition $C_i(0)$ to its steady state $\frac{\mathbf{E}_i^T\mathbf{I}}{1-\lambda_i}$, while a positive value of $\lambda_i - 1$ results in C_i diverging, and subsequently $\mathbf{V}$ diverging in the direction $\mathbf{E}_i$. In case $\lambda_i < 1 \quad \forall i$, the steady state value of $\mathbf{V}$ can be written as:

$$\mathbf{V}_{\text{inf}} = \sum_{i=1}^{N} \frac{\mathbf{E}_i\mathbf{I}}{1-\lambda_i}\mathbf{E}_i$$

From the above equation we can see that the direction $\mathbf{E}_i$ with maximum λ_i dominates the final outcome: the term that gets weighted most in the above sum is the projection of the external input $\mathbf{I}$ on this direction. The network can thus be seen as a system that favors certain features of the input, by enhancing the projections of the input on some directions and suppresses others. The network thus functions *globally* in a way that cannot be explained by analyzing the behavior of each single computational unit: to say it more technically, the eigenvectors of W determine the behavior of the system and not each of its columns separately. This is in accordance with the common maxim in neural networks that 'the whole is more than the parts'.

2.2.3 Nonlinear Recurrent Networks

When the firing rate of a neuron is no longer supposed to be a linear function of its input, the dynamics of the system become nonlinear and more complicated since we can no longer use

eigenvalues and eigenvectors as was done in the previous section. The main technique used for the global analysis of nonlinear systems, whenever it is applicable, is the introduction of a *Lyapunov function* that facilitates their analysis.

We will confine ourselves to networks with *symmetric connections*, i.e. $w_{i,j} = w_{j,i}$, that allow the introduction of such a function; even though networks with symmetric connections are of reduced biological plausibility [17] and flexibility [68], this assumption greatly illuminates recurrent network behavior. We proceed by presenting models of increasing complexity, presenting initially discrete Hopfield networks, subsequently continuous Hopfield networks and finally a wide class of nonlinear recurrent networks, that were studied by Cohen and Grossberg.

Discrete Hopfield Networks

Discrete Hopfield networks were the first recurrent neural networks that were studied in terms of Lyapunov functions; they were introduced in [52] and since then have been extensively studied and extended (e.g. [47], [5]). The model of a neuron that Hopfield used was the McCullogh-Pitts model, i.e. a computational unit that adds its inputs (voltages) and is active in case the sum of its inputs exceeds a threshold $-I_i$:

$$U_i(t+1) = H\left(\sum_{j=1}^{N} w_{i,j}U_j(t) + I_i\right) \tag{14}$$

where H is the step function. In this equation, U_i is the output of the i^{th} neuron, and can take values only in $\{0, 1\}$. Hopfield networks were proposed as models of associative memories, with memory patterns $\mathbf{E}_i$ stored as stable points (point attractors) of the system in (14):

$$\mathbf{E}_i = \mathbf{H}(\mathbf{W}\mathbf{E}_i + \mathbf{I})$$

A major difference between the dynamics of the systems (14) and (12) is that the introduction of the nonlinearity in the former does not allow the system to converge to a linear combination of its outputs, as the former does: it can be shown [47] that the network system will converge to the steady-state that is closest in Hamming distance to the initial state of the system. This is in sharp contrast with the linear behavior that is reflected in equation (12), where the steady state is a linear combination of the system's inputs.

The model of a neuron used is among the simplest and can be classified as a mean firing rate model, using the Heavyside function to associate its generator potential $\sum_{j=1}^{N} w_{i,j}U_j(t) + I_i$ with its firing rate; it is not the architecture of the model that was the novelty, but the introduction of the *Lyapunov function*

$$E = -\frac{1}{2}\sum_{i,j}^{N} w_{i,j}U_iU_j + \sum_{i=1}^{N} I_iU_i \tag{15}$$

of the system (14). E can be easily seen to be a non-increasing function of time if $w_{i,j}$ is symmetric: supposing the value of a single neuron, i is updated each time, the change ΔE

in E due to the change ΔU_i is

$$\Delta E = -\left[\sum_j U_j w_{i,j} + I_i\right] \Delta U_i$$

However, by (14), ΔU_i is positive if the term $U_j w_{i,j} + I_i$ is positive, so ΔE can never be positive. Given that E is a bounded from below function since $0 \leq U_i \leq 1 \quad \forall i$, any sequence of changes in the state of the network will eventually lead to a steady point.

By adding to E a constant greater than its minimum, we get a Lyapunov function of the system (14). A Lyapunov function of a system is an energy-like i.e. positive definite function of the state of a system, that does not increase as the system evolves according to its dynamics. The existence of such a function means that the system will converge, since it descends on a bounded from below function. A system may accept a variety of Lyapunov functions while it may also accept none. The introduction of a Lyapunov function facilitates analysis of a system's stability and behavior in a *global* way, contrary to the local behavior that can be analyzed using the inherently local linearization techniques. Systems much more sophisticated than (14) had been studied before, but the introduction of (15) facilitated their analysis on a firmer setting.

The input to a Hopfield network can be considered either its initial state, where it functions as an associative memory, or the vector $\mathbf{I}$ where it can be seen as a minimizer of the $\mathbf{I}$ dependent energy (15).

Continuous Hopfield Networks

Continuous Hopfield networks were presented in [53] and their behavior was analyzed in terms of Lyapunov functions as well. They retain the collective non-linear behavior of their discrete ancestors, while being closer to biological models of neurons. Specifically, the outputs of continuous Hopfield networks are allowed to take continuous values, by replacing the step-function H in (14) by a continuous sigmoid function g, shown in fig. 6. β plays

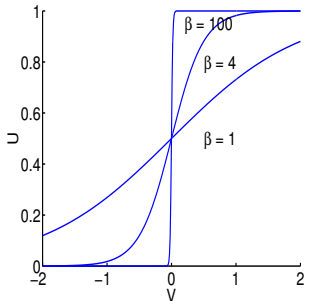


Figure 6: $g(V) = \frac{1}{1+\exp(-2\beta V)}$ for various values of β

the role of a steepness parameter, and for $\beta \to \infty$ we get the original step-function H, used

in (14). Using $\beta \neq 0$ is much closer to the way we think of a neuron in the MFR model, since the firing rate of a neuron is a smooth, increasing function of the input it receives, and not the binary result of thresholding the input. The passive, smoothing behavior of the membrane was incorporated in this model resulting in the following dynamics for the network :

$$\begin{aligned} C\frac{\mathrm{d}V_i}{\mathrm{d}t} &= -\frac{V_i(t)}{R} + \sum_{j=1}^{N} w_{i,j}U_j(t) + I_i \\ U_j &= g(V_j), \quad j = 1, \dots N \end{aligned} \tag{16}$$

This system accepts the Lyapunov function

$$E = -\frac{1}{2}\sum_{i,j=1}^{N} w_{i,j}U_iU_j + \sum_{i=1}^{N}\frac{1}{R}\int_{1/2}^{U_i} g^{-1}(U)\mathrm{d}U - \sum_{i=1}^{N} I_iU_i \tag{17}$$

since we have

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}t} &= -\frac{1}{2}\sum_{i,j=1}^{N} w_{i,j}\frac{\mathrm{d}U_i}{\mathrm{d}t}U_j - \frac{1}{2}\sum_{i,j=1}^{N} w_{i,j}U_i\frac{\mathrm{d}U_j}{\mathrm{d}t} + \frac{1}{R}\sum_{i=1}^{N} g^{-1}(U_i)\frac{\mathrm{d}U_i}{\mathrm{d}t} - \sum_{i=1}^{N} I_i\frac{\mathrm{d}U_i}{\mathrm{d}t} \\ &\overset{w_{i,j}=w_{j,i}}{=} -\sum_{i=1}^{N}\frac{\mathrm{d}U_i}{\mathrm{d}t}\left(\sum_{j=1}^{N} w_{i,j}U_j - \frac{1}{R}V_j + I_i\right) \\ &\overset{(16)}{=} -\sum_{i=1}^{N} g'(V_i)/C\left(\sum_{j=1}^{N} w_{i,j}U_j - \frac{1}{R}V_j + I_i\right)^2 \leq 0 \end{aligned}$$

since g is a nondecreasing function. Using $g^{-1}(x) = \frac{1}{\beta}\ln(x/(1-x))$ and integrating, the term $\sum_{i=1}^{N}\frac{1}{R}\int_{1/2}^{U_i} g^{-1}(U)\mathrm{d}U$ can be written as:

$$\frac{1}{\beta R}\sum_{i=1}^{N}(1-U_i)\ln(1-U_i) + U_i\ln(U_i) = \frac{1}{\beta R}\sum_{i=1}^{N}\sum_{S_i\in ON,OFF} p_{S_i}\ln(p_{S_i})$$

This can be seen as a negative entropy function, where U_i is taken as the probability of the random variable S_i (S for state) being ON (i.e. the i^{th} neuron firing) and $1-U_i$ is the probability of S_i being off (i.e. the i^{th} neuron being silent). The addition of this term punishes neuron outputs U close to 0 or 1, that is, binary decisions. This is the result of using a sigmoid function for g, which requires a strong positive or negative input to get a binary output. Using continuous Hopfield networks for energy minimization problems was first proposed in [54] and is reviewed in Appendix A.

Cohen and Grossberg's Recurrent Networks

In [14] Cohen and Grossberg came up with a Lyapunov function to analyze a much broader class of networks than those presented in the previous sections. Even though S. Grossberg had proposed these models of networks during the previous two decades, the use of the specific Lyapunov functions for their analysis was first presented in [14]. It was shown that networks that can be written in the form:

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} = a_i(V_i)\left[b_i(V_i) - \sum_{j=1}^{N} c_{i,j} d_j(V_j)\right] \tag{18}$$

accept the following global Lyapunov function:

$$E = -\sum_{i=1}^{N} \int_{-\infty}^{V_i} b_i(x_i) d_i'(x_i) \mathrm{d}x_i + \frac{1}{2} \sum_{i,j=1}^{N} c_{i,j} d_i(V_i) d_j(V_j) \tag{19}$$

if the coefficient matrix $C = \| c_{i,j} \|$ and the functions a_i, b_i and d_j obey some conditions including *positivity* $a_i(V_i) \geq 0$, *monotonicity* $d_j'(V_j) \geq 0$ and the matrix $C = \| c_{i,j} \|$ is a symmetric matrix of non-negative constants.

A broad class of systems can be expressed in the form (18). For example in [32], the system (16) was expressed in the form (18), by taking

$$\left.\begin{array}{ll} a_i(V_i) & = \frac{1}{C} \\ b_i(V_i) & = -\frac{1}{R} V_i + I_i \\ c_{i,j} & = -w_{i,j} \\ d_j(V_j) & = g(V_j) \end{array}\right\} \Rightarrow (18) \leftrightarrow (16)$$

and the Lyapunov function (19) is in that case:

$$E = -\frac{1}{2} \sum_{i,j=1}^{N} w_{i,j} U_i U_j - \sum_{i=1}^{N} I_i U_i + \frac{1}{R} \sum_{i=1}^{N} \int_0^{V_i} V g'(V) \mathrm{d}V, \quad U_i = g(V_i) \tag{20}$$

The only apparent difference between (20) and (17) lies in the last term, which is actually the same quantity, as can be seen from the relation $V = g^{-1}(U)$; hence $g'(V)\mathrm{d}V = \mathrm{d}U$ and the limits of integration become $\int_{1/2}^{U_i}$. Apart from these comparisons, another model used by S. Grossberg in [14] for a neuron is also of interest to us:

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} = -a_i V_i + (b_i - V_i)[I_i + f(V_i)] - (V_i + c_i)[J_i + \sum_{j=1}^{N} w_{i,j} g_j(V_j)] \tag{21}$$

This is, apart from some differences, similar to the synaptic excitation/inhibition mean firing rate model (10), using the following correspondence of terms

(21)	(10)	
a_i	$\frac{1}{R}$	leakage conductance
b_i	V_e	excitation potential
$c_i(>0)$	$-V_i$	inhibition potential
I_i	$\sum_{n\neq i} w_n^e U_n$	excitation from other neurons
$f(V_i)$	$w_i^e U_i$	self-excitation
$J_i + \sum_{j=1}^N w_{i,j} g_j(V_j)$	$\sum_{n=1}^N w_n^i U_n$	inhibitory input from other neurons

The main difference between (10) and (21) lies in that in (10) excitatory input $\sum_{n\neq i} w_n^e U_n$ comes from neighbors that are temporally varying and may be affected by the value of the current neuron V_i, while in (21) the excitatory input I_i is considered as a constant, coming from some other layer. However the model (10) is used by S. Grossberg as well, mainly in his work concerning biological vision.

As discussed in [14], (21) can be written in the form (18) under some suitable redifinition of terms

$$\begin{aligned} a_i(V_i) &= V_i \\ b_i(V_i) &= \frac{1}{V_i}\left[a_i c_i - (a_i + J_i)V_i + (b_i + c_i - V_i)(I_i + g_i(V_i - c_i))\right] \\ c_{i,j} &= w_{i,j} \\ d_j(V_j) &= g_j(V_j - c_j) \end{aligned}$$

It was shown in [14] that the system (18) accepts a Lyapunov function even if g is not invertible, as long as g is nondecreasing; it is therefore not necessary to use a sigmoid function, but e.g. the clipping function can be used instead.

2.3 Visual System Essentials

The human visual system is organized in separate but communicating layers that perform increasingly sophisticated operations, starting from cells responding to intensity variations (in the retina) and ending at highly specified cells that respond to a very distinct class of visual inputs like a hand or a face. The areas that are of primarily interest for the purpose of image segmentation are [56]:

- The retina
  which is the rear of the eye, and acts like the 'camera' of the visual system.
- The Lateral Geniculate Nucleus (L.G.N.)
  which regulates the flow from the retina to the cortex, with magnocellular and parvocellular cells sending to the visual cortex fast & coarse and slow & fine signals respectively.
- Area V1 (a.k.a. Area 17, Striate Cortex)
  where elementary feature detection takes place. It is there where cells have been detected responding maximally to simple patterns like step edges and bars at prescribed orientations.

- Area V2
  where the same features as the ones in area V1 are detected, but at a higher level, e.g. there are cells responding to illusory contours, and with a higher degree of invariance.

In the following, we shall briefly present some important and well studied cells as well as some mathematical models that are used to approximate their behavior. These formulae are neither the only ones that can model the observed behavior of these cells nor perfectly correct; most of them are based on the feedforward model of computation that has been advocated by Hubel & Wiesel, but there is certainly place and evidence for more complex models, using recurrent connections among cells of the same and different layers. We should therefore keep in mind that the following formulae are more of a qualitative nature, rather than precise models; in our implementations we use recurrent connections and nonlinear functions, that deviate from the linear filtering-like presentation that follows. The presentation in this subsection follows mainly [56] and [17].

2.3.1 On-Off/Off-On Cells

On-Off cells respond maximally when a bright dot surrounded by a dark region is presented to their receptive field [2], while Off-On cells respond maximally to a dark dot surrounded by a bright region.

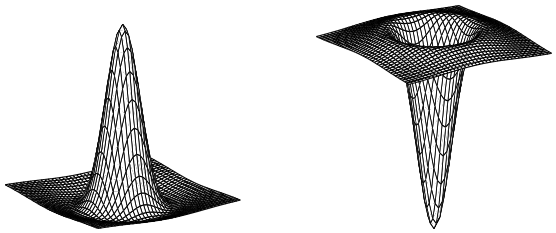
These cells are encountered in the retina as well as in the L.G.N. and can be thought of as an economical way of presenting an image to the cortex: no other information is useful, apart from the location and the magnitude of the change in the input image intensity. Contrary to what one might think when ignoring the visual system, a retinal cell will not respond when uniform light is flushed onto it, however bright the light.

A commonly used model for an On-Off cell is a Difference of Gaussians (DoG) filter, where the cells response is calculated by convolving its input with two Gaussians, one with a large spread (corresponding to the background region) and another with a smaller spread (corresponding to the foreground region), taking their difference and rectifying:

$$\begin{aligned} X^{On-Off} &= [I * G_{\sigma_1} - I * G_{\sigma_2}]^+ \\ X^{Off-On} &= [I * G_{\sigma_2} - I * G_{\sigma_1}]^+, \\ \text{where} \quad G_{\sigma_i}(x,y) &= \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{x^2+y^2}{2\sigma_i^2}\right), \quad \sigma_1 < \sigma_2 \quad [.]^+ = \max(.,0) \end{aligned}$$

The receptive fields of an On-Off and an Off-On cell modeled as a difference of Gaussians are shown in fig. 7. Using suitably selected spread ratios for the two Gaussian filters ($\frac{\sigma_2}{\sigma_1} \simeq 1.4$), this filter can be used as a good approximation to the Laplacian of Gaussian (LoG) filter, that is routinely used in image processing for edge detection. The DoG filter approximation does not fully account for the behavior of On-Off cells cells: these cells are known to perform contrast normalization, i.e. they are capable of responding in the same way in environments

[2] The receptive field of a cell refers to the area of the retina in which appropriate light stimulation evokes a response in the cell, as well as to the pattern of light stimulation that evokes such a response.



(a) (b)

Figure 7: The receptive fields of (a) On-Off and (b) Off-On cells modeled using Difference of Gaussians filters

of low and high contrast, which could be modelled by dividing the filter's output by a local average of neighboring cell responses. This can be accomplished with shunting inhibition [12, 31] using lateral connections [43], but is not feasible by the additive variety of MFR neurons. This is a simple case where using the model (11) instead of the simpler (8) may account better for the behavior of the visual system cells.

2.3.2 Simple Cells

Simple Cells are located in area V1 of the cortex, and they perform the most elementary feature detection tasks: simple cells respond maximally to oriented bars or step edges at a specific orientation and location. For each location and orientation we can think that there is a simple cell responding maximally to a step-like increase in image intensity, another responding to decrease in intensity, and two simple cells responding to a bright bar on a dark background and a dark bar on a bright background in that orientation. Of course this is not a strict rule, since simple cells come in many varieties, but it gives an idea about the role and functionality of these cells.

The most common and best studied model of the receptive fields of simple cells uses 2D Gabor filters, as proposed in [16]; the outputs of simple filters are calculated by convolving the input image with an appropriate Gabor filter and half-wave rectifying the output. Even though it is not true that simple cells come exclusively in sinus-cosinus pairs [16], it is most common to use such even/odd symmetric pairs, e.g. for cells detecting changes in the horizontal direction:

$$G^{Even}(x,y) = \cos(2\pi kx)\frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\left\{\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2}\right\}\right) \quad (22)$$

$$G^{Odd}(x,y) = \sin(2\pi kx)\frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\left\{\frac{x^2}{2\sigma_x^2}+\frac{y^2}{2\sigma_y^2}\right\}\right) \quad (23)$$

k is the spatial frequency of the sinusoidal input that maximally activates the cell, σ_x determines the spatial frequency selectivity of the filter and σ_y its angular selectivity. Some commonly used relations for these parameters are [66]:

$$\frac{\sigma_x}{\sigma_y} = \frac{1}{2}, \quad b \in [0.5, 2.5], \text{where} \quad b = \log_2 \left(\frac{k\sigma_x + \sqrt{2\ln(2)}}{k\sigma_x - \sqrt{2\ln(2)}} \right)$$

b is called the *bandwidth* of the filter [17] and is defined as $\log_2(K_+/K_-)$ where $K_+ > k$ and $K_- < k$ are the frequencies of the sinusoidal inputs which produce one-half the response amplitude of an input with $K = k$. Actually, the filter in (23) does not have a zero DC response, as is natural for even-symmetric Gabor filters. This is undesirable, since this filter is used to model a cell that should not respond when it is presented with constant input. A remedy to this problem was suggested in [45] and in [67]: the latter, which seems simpler, consists in subtracting from the even-symmetric filter a Gaussian filter aG_{σ_x,σ_y}, of the same spread as the Gaussian in (23). The modified filter becomes:

$$G'^{Even}(x,y) \quad = \quad (\cos(2\pi kx) - a)\frac{1}{2\pi\sigma_x\sigma_y} \exp\left(- \left\{ \frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} \right\} \right) \tag{24}$$

In this way, the filter's shape stays approximately the same, while its DC response becomes zero if we set as in [67] $a = \mathcal{F}[G^{even}](0,0)$ where $\mathcal{F}$ is the Fourier Transform of the filter.

Alternatively, if the outputs of the previous processing step (On-Off/Off-On cells) are used as inputs to simple cells, the behavior of simple cells can be approximated by convolving the On-Off and Off-On responses with suitably chosen Difference-of-offset Gaussian (DooG) filters. We consider the four varieties of simple cells that respond to vertically oriented inputs, shown in fig. 8. The behavior of these cells can be modeled by convolving their

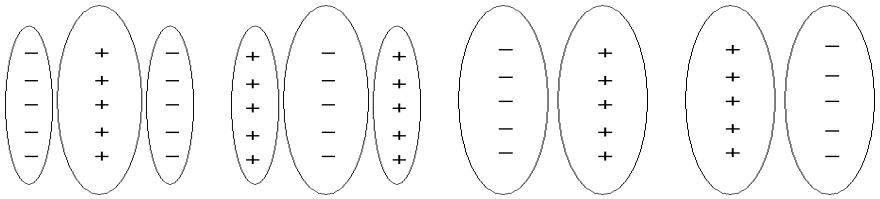


Figure 8: (a) On-Off- (b) Off-On cells feeding into a simple cell detecting a bright vertical bar on a dark background (c) On-Off- (d) Off-On cells feeding into a simple cell detecting an increase of image intensity from left to right (a step edge); + denotes an excitatory connection and − an inhibitory connection

inputs (On-Off/Off-On cells outputs) with appropriately elongated and offset Gaussians: the ellipsoid in the center of fig. 8(a),(b) is given by

$$G_C(x,y) = \frac{1}{2\pi\sigma_{c,x}\sigma_{c,y}} \exp\left(- \left\{ \frac{x^2}{2\sigma_{c,x}^2} + \frac{y^2}{2\sigma_{c,y}^2} \right\} \right)$$

and its two side lobes are given by

$$G_{S_L/S_R}(x,y) = \frac{1}{2\pi\sigma_{s,x}\sigma_{s,y}} \exp\left(-\left\{\frac{(x \pm (\sigma_{c,x}+\sigma_{s,x}))^2}{2\sigma_{s,x}^2} + \frac{y^2}{2\sigma_{s,y}^2}\right\}\right)$$

while for the case of the step edge (fig. 8(c),(d)) we can use for the two lobes:

$$G_{L/R}(x,y) = \frac{1}{2\pi\sigma_{r,x}\sigma_{r,y}} \exp\left(-\left\{\frac{(x \pm \sigma_{r,x}/2)^2}{2\sigma_{r,x}^2} + \frac{y^2}{2\sigma_{r,y}^2}\right\}\right)$$

The following notation has been used:

$\sigma_{c,\{x,y\}}$	Spread of central lobe in x / y directions (bar edge)
$\sigma_{s,\{x,y\}}$	Spread of the side lobes in x / y directions (bar edge)
$\sigma_{r,\{x,y\}}$	Spread of Left/Right lobe in x / y directions (step edge)

We can rewrite the above equations for an arbitrary orientation θ, by making a change of coordinates of the filters

$$G(x,y) \rightarrow G(x\cos(\theta) + y\sin(\theta), y\cos(\theta) - x\sin(\theta))$$

The responses of the cells are:

$$\begin{aligned}
L_\theta^{s_1} &= \left[X^{On-Off} * G_c + X^{Off-On} * (G_L + G_R) - \right. \\
&\quad \left. [X^{Off-On} * G_c + X^{On-Off} * (G_L + G_R)]\right]^+ \qquad (25)\\
L_\theta^{s_2} &= \left[X^{Off-On} * G_c + X^{On-Off} * (G_L + G_R) - \right. \\
&\quad \left. [X^{On-Off} * G_c + X^{Off-On} * (G_L + G_R)]\right]^+ \qquad (26)\\
L_\theta^{s_3} &= \left[X^{On-Off} * G_L + X^{Off-On} * G_R - (X^{Off-On} * G_L + X^{On-Off} * G_R)\right]^+ \qquad (27)\\
L_\theta^{s_4} &= \left[X^{Off-On} * G_L + X^{On-Off} * G_R - (X^{Off-On} * G_L + X^{On-Off} * G_R)\right]^+
\end{aligned}$$

where $L_\theta^{s_i}$ stands for the i^{th} variety of simple cells in the θ orientation. In each equation, the first two terms model the excitatory synaptic input and the last two the inhibitory input.

2.3.3 Complex Cells

Complex Cells are found in area V1 of the striate cortex, and as their name implies, they respond in a more complicated way to their input: complex cells are *phase insensitive*, so they are insensitive to the sign of intensity change in their receptive fields. Apart from that, complex cells may receive input from both eyes, i.e. are binocular, contrary to simple cells that are monocular. Even though complex cells discard the phase sensitivity of simple cells, they retain their orientational sensitivity and respond to a limited area of the visual field.

Such cells could be assumed to be pooling the outputs of multiple phase-sensitive simple cells, and simply adding their inputs, or taking their maximum. However, there is evidence (see e.g.[17], p. 74-76) that the output of a complex cell can be approximated by taking the

sum of the squares of an even-symmetric and an odd-symmetric Gabor filter, centered at the same location and with the same orientational sensitivity. Even though the physiological mechanism which could accomplish this is not evident, we can use it in our simulations by squaring and adding the rectified outputs of 4 simple cells, 2 of them corresponding to the even symmetric filter and 2 to the odd-symmetric filter:

$$C_\theta[i,j]) = [L^1_\theta[i,j]]^2 + [L^2_\theta[i,j]]^2 + [L^3_\theta[i,j]]^2 + [L^4_\theta[i,j]]^2$$

The pair of filters used constitutes a quadrature pair and the sum of their squares is a local estimate of the signal energy in that direction.

2.3.4 End-Stopped (Hypercomplex) Cells

End-Stopped Cells -also referred to as Hypercomplex cells- are found in the striate cortex and have the following distinctive behavior: they respond maximally when their stimulus is an edge of a specific orientation but also of specific length and their activity decreases when they are presented with a longer/shorter edge. Such cells have been conjectured to implement a neural mechanism for curvature estimation in the visual system, since we could 'build' such a cell to respond maximally when it is tangent to a circle with a specific curvature, by appropriately tuning spatially its excitatory and inhibitory areas. It has also been argued [45] that the function of these cells is the detection of border terminations, in case we take their single-stopped variety i.e. end-stopped cells that are insensitive to edge length on one of their receptive fields sides and sensitive on the other. Such cells could be used both as image feature detectors and as cues for the creation of illusory contours: when many such cells are active along a line, this signals the existence of an occluder that has caused the observed edge terminations. Such cells have been modeled by derivatives of complex cell activations in [45, 46].

2.3.5 Blob Cells

Blob cells have no orientational sensitivity, receive monocular input and carry only color-related information. Even though they are not directly involved in the estimation of contours, they have been proposed by S. Grossberg to be at the basis of the mechanism that is used to *fill-in* the interior of uniform regions, that is not represented anywhere else in the visual system. This mechanism deals with the fact that even though only differences in the image excite On-Off and Off-On cells -and subsequently the rest of the aforementioned cells, we perceive continuous surfaces and not simply their borders. The image brightness information has to be available somewhere, in order to be further processed; this is where blob cells could be helpful, as a 'buffer' of surface-related (and not form-related) information.

3 Stephen Grossberg's Model of Low & Mid Level Vision

The model proposed by S. Grossberg and his collaborators in a series of papers (see articles in [30],[31], as well as extensions and reviews in [28, 27, 37, 33, 39]), known as the FACADE (Form And Colour and DEpth) theory of vision is probably one of the few models to account for such a wide variety of visual functions, based solely on neural mechanisms: starting with edge detection, it proceeds with contour grouping [36, 35], surface & depth perception [41, 37, 40] and binocular vision [33, 27, 34, 40], while it has been used as a preprocessing step for motion analysis [38].

In this versatile model of vision most of the ideas are relatively straightforward, assuming one has a background in neural computation and biological vision; as a whole, however, the system becomes complicated, in terms of both its functionality and its analysis. This is natural, though, for any model of something as complicated as our visual system; the model's ability to explain a plethora of psychophysical phenomena [33] in a unified way offers support for its plausibility and motivation for studying it in depth, trying to relate and compare it with computer vision techniques.

We shall focus on the parts that form the basic building blocks of this model, namely the Boundary Contour System (B.C.S.) and the Feature Contour System (F.C.S.); these have been used to model monocular image perception [28, 39] and by adapting them and their interactions to fuse information from two images at multiple depths they can be used to model binocular image perception [27, 33]. The interactions between these two systems are shown in the block diagram in fig. 9. In the monocular case, the Boundary Contour

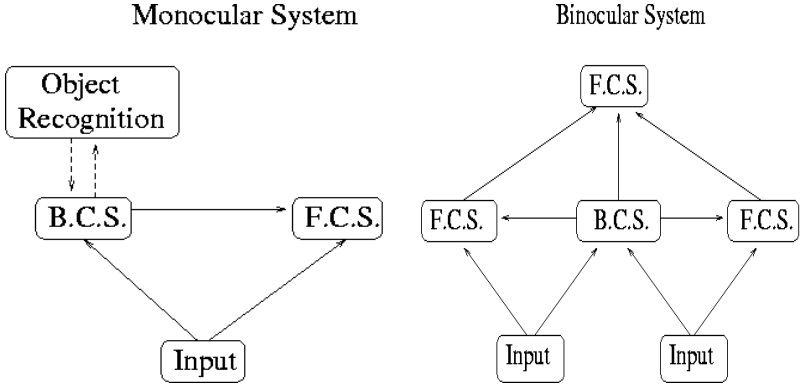


Figure 9: The interactions between the B.C.S./F.C.S. components

System detects the coherent contours in the image, and sends their locations to the Feature Contour System. Then, the F.C.S. diffuses isotropically its image-derived input apart from the areas where there is input from the B.C.S., signaling the existence of an edge. This

results in an anisotropic smoothing procedure, where the degree and direction of anisotropy is determined by the B.C.S.. An optional component proposed in [28, 33] is an Object Recognition system, that recognizes the objects in the image from their outlines and helps in the formation and linking of their boundaries. In the binocular case, the left and the right image edges are fused into a binocular edge map, that is sent as input to two separate F.C.S.s; each of them forms a monocular image that is consistent with the other one, and they are subsequently fused in a binocular F.C.S.. This is the case presented in [27], while in the complete FACADE theory the binocular case is somewhat more complicated [33, 34].

Before presenting in detail the B.C.S. and F.C.S., it would be useful to have in mind a 'road map' of the monocular system's architecture, shown in fig. 10. The stages marked

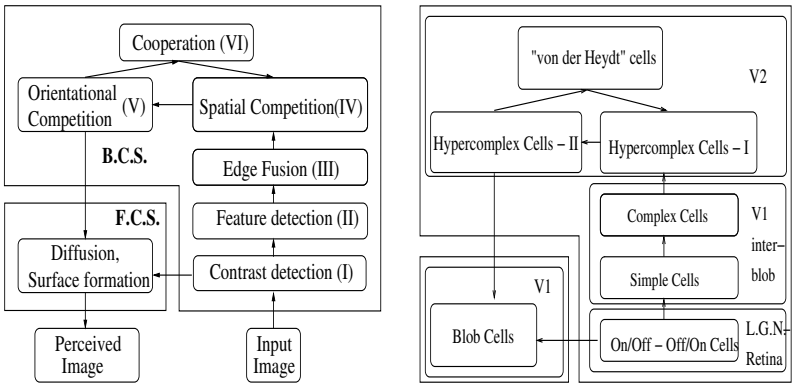


Figure 10: (a) A block diagram of the B.C.S./F.C.S. architecture. (b) corresponding areas in the visual system

with I-III act in a purely feedforward manner, each layer feeding its output to the next and as a whole they perform an edge detection task; the more interesting part of the model is the loop between stages IV-VI whose role is to detect and enhance the most salient edges, and this is where illusory contours and perceptual groupings arise. The interaction of the B.C.S. with the F.C.S. results in the formation of the perceived image.

Before proceeding with the presentation of the B.C.S./F.C.S. architecture, it should be noted that the components of these systems have been subject to many modifications, but the architecture and the main concepts have not changed significantly. Therefore we shall neither present a single proposed variety nor all of them; instead, we have used these components that seemed to combine simplicity and utility and were up-to-date. Several variations that have been proposed were tried and we present the ones that collectively gave the best performance. Many thresholding and compression steps that have been used by S. Grossberg and his collaborators at various stages of the network are not mentioned, since they do not form, in our opinion, an essential part of the B.C.S./F.C.S. architecture. Values for the used constants are given in Appendix C, while our comments are gathered at subsection 3.4. We should also mention that this section is based on the papers cited

previously; in a recent paper [83] changes to the B.C.S. architecture were proposed which are similar to the ones we have used in our model, presented in the following section, we keep most of the 'discussion' subsection for the B.C.S. model and not for the work in [83].

3.1 The Boundary Contour System (B.C.S.)

The Boundary Contour System detects, enhances and groups the edges that exist in an image, forming continuous borders by a recurrent among layers process. Every processing stage involved in this computation has been proposed with a specific area and function of the visual cortex in mind, starting from the retina and ending at area V2. For illustration purposes, the effect of each stage on a simple input image, shown in fig. 11, will be shown.

3.1.1 Stage I: Contrast detection

This is the function performed by cells in the retina and the L.G.N.; the equations used by S. Grossberg [41, 39] are of the shunting type, which have been presented in section 2.1.2:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}U^{On-Off}[i,j] &= -aU[i,j] + (b - U[i,j])C_{i,j} - (U[i,j] + d)E_{i,j} \\
\frac{\mathrm{d}}{\mathrm{d}t}U^{Off-On}[i,j] &= -aU[i,j] + (b - U[i,j])C_{i,j} - (U[i,j] + d)E_{i,j}
\end{aligned}$$

where $C_{i,j}$ is the total excitatory and $E_{i,j}$ the total inhibitory input that a retinal cell located at position $[i,j]$ receives, b is the maximal attainable output and $-d$ is the minimal. $C_{i,j}$ and $E_{i,j}$ are computed by convolving the input image I with Gaussian filters of different spreads, σ_1, σ_2:

$$\begin{aligned}
C_{i,j} &= \sum_{k=-N}^{N}\sum_{l=-N}^{N} I[i-k, j-l]G_{\sigma_1}(k,l) \\
E_{i,j} &= \sum_{k=-N}^{N}\sum_{l=-N}^{N} I[i-k, j-l]G_{\sigma_2}(k,l)
\end{aligned}$$

where $\sigma_1 < \sigma_2$ for an On-Off cell and $\sigma_1 > \sigma_2$ for an Off-On cell.

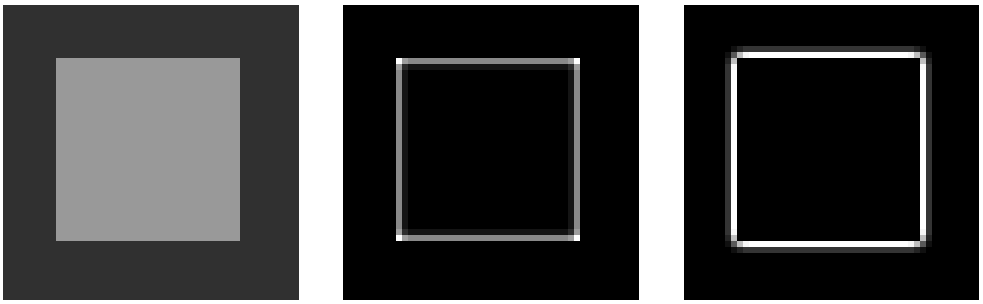
The fact that all the input the cell receives is from a previous layer (in this case the input image) allows us to pass as output to the next stage the rectified state solution:

$$\begin{aligned}
U_{\infty}^{On-Off}[i,j] &= \left[\frac{bC_{i,j} - dE_{i,j}}{a + C_{i,j} + E_{i,j}}\right]^{+} \\
U_{\infty}^{Off-On}[i,j] &= \left[\frac{bE_{i,j} - dC_{i,j}}{a + E_{i,j} + C_{i,j}}\right]^{+}
\end{aligned}$$

The divisor in the above equations accounts for contrast normalization that is performed by the visual system which allows it to operate over a large scale of inputs without saturating;

if we multiply the image I by a constant c the new response U' shall be related to U by $U' = \frac{1}{1+a/c}U \simeq U$ if $a << c$, while for a linear model it would be $U' = cU$.

In fig. 11(b),(c) we show the outputs of On-Off/Off-On cells when presented with the test image; in this and the following images the outputs of the neurons are shown normalized in the range $[0, 1]$.



(a) (b) (c)

Figure 11: (a): input image,(b) On-Off cell responses,(c) Off-On cell responses

3.1.2 Stage II: Elementary Feature detection

This is the process accomplished by simple cells in area V1 of the visual cortex; using the feedforward connections from the previous layer as in section 2.3.2 we can model simple cells detecting step edges at orientations $\theta = k\pi/N, k = 1 \dots N$:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}V^{\theta,+}[i,j] &= -V^{\theta,+}[i,j] + C^{\theta}_{i,j} - E^{\theta}_{i,j} \\ \frac{\mathrm{d}}{\mathrm{d}t}V^{\theta,-}[i,j] &= -V^{\theta,-}[i,j] + E^{\theta}_{i,j} - C^{\theta}_{i,j} \end{aligned}$$

As in section 2.3.2, θ denotes the orientation at which a simple cell detects changes and $+/-$ denote whether it is sensitive to an increase or a decrease in On-Off cell activations in its preferred orientation. $C^{\theta}_{i,j}, E^{\theta}_{i,j}$ are the total excitation/inhibition (inhibition/excitation) cell $V^{\theta,+}$ ($V^{\theta,-}$) at location $[i,j]$ receives:

$$\begin{aligned} C^{\theta}_{i,j} &= \sum_{l,m=-N}^{N} F^{\theta}_1(l,m)U^{On-Off}[i-l,j-m] + F^{\theta}_2[l,m]U^{Off-On}[i-l,j-m] \\ E^{\theta}_{i,j} &= \sum_{l,m=-N}^{N} F^{\theta}_1[l,m]U^{Off-On}[i-l,j-m] + F^{\theta}_2[l,m]U^{On-Off}[i-l,j-m] \end{aligned}$$

$$F_1^\theta[l,m] = \frac{e^{-\left\{\frac{((l-c)\cos(\theta)+m\sin(\theta))^2}{2\sigma_x^2}+\frac{((l-c)\sin(\theta)-m\cos(\theta))^2}{2\sigma_y^2}\right\}}}{2\pi\sigma_x\sigma_y} \quad (28)$$

$$F_2^\theta[l,m] = \frac{e^{-\left\{\frac{((l+c)\cos(\theta)+m\sin(\theta))^2}{2\sigma_x^2}+\frac{((l+c)\sin(\theta)-m\cos(\theta))^2}{2\sigma_y^2}\right\}}}{2\pi\sigma_x\sigma_y} \quad (29)$$

The expression $F_1^\theta[l,m]$ is the value of a 2-D Gaussian with variances along the principal axes σ_x, σ_y, $\sigma_y > \sigma_x$ offset by c to the right and rotated by θ. These filters correspond to the ones presented in section 2.3.2: e.g. for $V^{\theta,+}$, C^θ is the total excitation received from On-Off/Off-On cells 'appropriately' active within its receptive field (that is, with the signs shown in fig. 12), and E^θ is the inhibition from 'inappropriately' active cells (with opposite signs). The only difference between V^+ and V^- lies in that what excites the one inhibits the other.

In most of the B.C.S. papers only odd-symmetric filters are used, that can detect step edges; however in a recent publication [34], the use of even-symmetric filters was proposed.

The outputs to the next stage are given by rectifying the steady state voltages:

$$\begin{aligned} V_\infty^{\theta,+}[i,j] &= \left[C_{i,j}^\theta - E_{i,j}^\theta\right]^+ \\ V_\infty^{\theta,-}[i,j] &= \left[E_{i,j}^\theta - C_{i,j}^\theta\right]^+ \end{aligned}$$

3.1.3 Stage III: Edge Fusion, Cue Integration

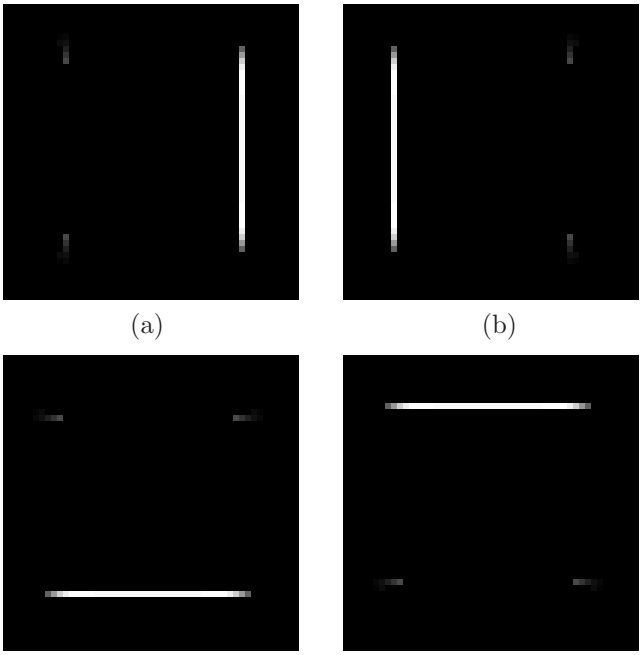
This is the process that is accomplished by complex cells in layer V1 of the visual cortex; it is known that complex cells receive binocular input and are phase and color insensitive, which suggests that they pool information from multiple simple cells. In the monocular case, the B.C.S. model accounts for this by adding the outputs of simple cells detecting edges at the same orientation and different directions (increase/decrease):

$$W_\infty^\theta[i,j] = V^{\theta,+}[i,j] + V^{\theta,-}[i,j]$$

If color images are being processed, the related equation is

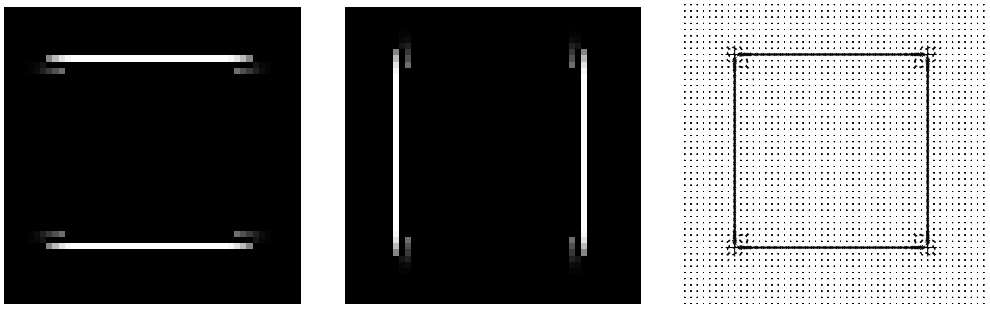
$$W_\infty^\theta[i,j] = V_w^{\theta,+}[i,j] + V_w^{\theta,-}[i,j] + V_b^{\theta,+}[i,j] + V_b^{\theta,-}[i,j] + V_r^{\theta,+}[i,j] + V_r^{\theta,-}[i,j]$$

where V_w, V_b, V_r are the responses of the simple cells that receive input from Black/White, Blue/Yellow and Red/Green On-Off cells respectively. For the binocular case a more complicated equation is used [34], which includes a recurrent term, that accounts for cross-disparity competition.



(a) (b) (c) (d)

Figure 12: (a),(b) Vertical simple cell responses (dark-to-light/light-to-dark), (c),(d) Horizontal simple cell responses (dark-to-light/light-to-dark)



(a) (b) (c)

Figure 13: (a) Vertical complex cell responses, (b) Horizontal complex cell responses (c)needle diagram of complex cell outputs.

Up to the third processing stage, most of the steps are fairly simple and have a clear-cut interpretation. Even though not everyone uses this exact structure and set of equations to

simulate the way the output of complex cells is derived, there is some general consensus about which layer does what. The subsequent stages are the most original stages of the system proposed by S. Grossberg and his collaborators, and our presentation shall therefore be more detailed from now on.

3.1.4 Stage IV: Spatial Competition

The edge maps derived from the previous stages are usually relatively broad, since complex cells with similar spatial and orientational selectivity receive similar inputs and therefore have similar responses. However, the visual system is known of being capable to detect edges with high precision, actually with higher resolution than it's receptive field sizes would suggest. This property, termed *hyperacuity*, is accounted for in the B.C.S. model by the 4^{th} stage of computation where spatial competition among neurons with the same orientation preference takes place; the image processing analog is edge thinning. Parallel to that, this stage is used as a preliminary step for *illusory contour formation*: cells whose orientational preference is orthogonal to an observed line ending become active using a mechanism termed *end-stopping*; such cells signal the possibility of an occluding contour causing the observed ending. Later on, (in stage VI), if more such cells are found along a smooth direction, they are grouped together, giving rise to an *illusory contour* that 'explains' the co-occurrence of the line endings. Complementary to that, in this stage *top-down* signals from grouping processes (stage VI) are used to give rise to illusory contours: a cell that is lying on a perceptually strong contour may get excited even though there may be no contrast (that is, *bottom-up* excitation) at its location in the input image.

In the B.C.S. model these processes are conjectured to be accomplished by end-stopped cells; it is common to model the responses of end-stopped cells as derivatives of complex cell activations [45, 46] and it is assumed they perform more elaborate computations than complex cells. However not all the functions described above are included in the usual definition and interpretation of the function of end-stopped cells (see sec.2.3.4, [56, 45, 46]), which are usually viewed as detectors of features like points of specific curvature, corners, etc.

The equation used at this level is

$$\begin{aligned} \frac{\mathrm{d}X^{\theta}[i,j]}{\mathrm{d}t} &= -aX^{\theta}[i,j]+(b-X^{\theta}[i,j])C^{\theta}[i,j]-(X^{\theta}[i,j]+d)E^{\theta}[i,j]+I+\delta Z^{\theta}[i,j] \\ C^{\theta}[i,j] &= W^{\theta} * G_{\sigma_1} \\ E^{\theta}[i,j] &= W^{\theta} * G_{\sigma_2}, \quad \sigma_1 < \sigma_2 \end{aligned} \tag{30}$$

In the above equation:

- $C^{\theta}(i,j)$ and $E^{\theta}(i,j)$ stand for the excitation and inhibition each cell receives from the complex cells in its orientation, and are calculated by convolving complex cell outputs, W^{θ}, with Gaussian filters of different spreads, (small for excitation, large for inhibition). The idea behind using Gaussian filters with different spreads is that a

cell that is more active than cells in its neighborhood will get more excitation than inhibition, since its excitatory signal comes from a more active area than the inhibitory, while a cell that is less active than its neighbors will be suppressed.

- I is a constant excitatory input (termed *tonic input*) that keeps every cell 'active' above a zero activity: this is used to implement the end-stopping mechanism proposed by S. Grossberg. This mechanism works as follows: using a positive offset, I, for the activity of all cells, the only cells that have zero activity at their steady-state are those that receive more inhibition, due to the mechanism presented above, than tonic input. Such cells appear at line endings, since they receive only inhibitory input, due to the surrounding activity; there is no excitatory input, as the edge terminates *next* to them. This results in zero activity next to a line ending in the line direction and more activity (equal to the tonic input) in the perpendicular dimension. This mechanism provides the following stages with the necessary occlusion information to build subjective contours. This tonic input term has not been used in any of the simulations of the B.C.S./F.C.S. system presented in this report because we encountered many problems with it even for simple synthetic images.
- At last, the feedback term, Z^θ comes from stage VI, signaling the existence of a perceptually important edge and is weighted by δ. This feedback term drives the process (30) to perceptually important boundaries, helping to disambiguate and link fragmented edges.

The output is again computed by rectifying the steady-state value of the system (30), assuming the evolution process is much faster than the rate of change of the feedback signal:

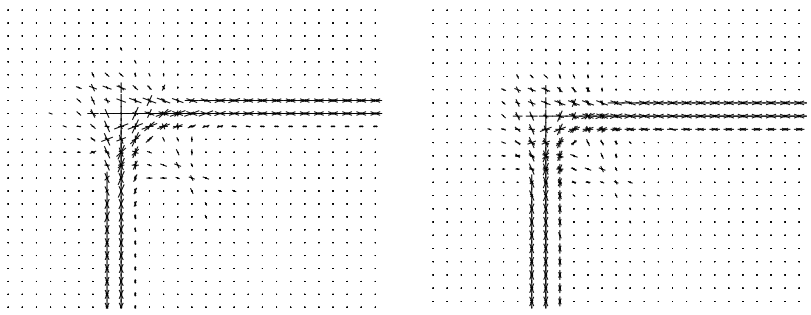
$$X^\theta_\infty[i,j] = \left[\frac{bC^\theta[i,j] - dE^\theta[i,j]}{a + C^\theta[i,j] + E^\theta[i,j] + I + \delta Z^\theta[i,j]}\right]^+$$

3.1.5 Stage V: Orientational Competition

This stage follows spatial competition and helps select the strongest orientation at each image point, while completing the end-stopping mechanism of stage IV. In the previous stage competition was among cells with the same orientation at different locations. In this stage, competition is among cells with the same location and different orientations. The connections are such that each previous stage cell $X^{\theta_1}[i,j]$ excites $Y^{\theta_2}[i,j]$ cells located close (in orientation) to itself and inhibits cells at the perpendicular and its neighboring orientations. The equation used is:

$$\frac{\mathrm{d}}{\mathrm{d}t}Y^\theta[i,j] = -aY^\theta[i,j] + (b - Y^\theta[i,j]) \sum_{k=0}^{N-1} e_{\theta,\frac{k\pi}{N}} X^{\frac{k\pi}{N}}[i,j] - (Y^\theta[i,j] + d) \sum_{k=0}^{N-1} i_{\theta,\frac{k\pi}{N}} X^{\frac{k\pi}{N}}[i,j]$$

where $\{0,\ldots,(N-1)/N \cdot \pi\}$ are the orientations used by our network. The connection strengths $i_{\theta_i,\theta_j}, e_{\theta_i,\theta_j}$ are such that e_{θ_i,θ_j} is positive for small angle differences between



(a) (b)

Figure 14: (a) The outputs of stage IV cells for the input image, compared to (b) the outputs of complex cells. A B.C.S. operating at a larger scale than those for the rest of the images has been used, in order to have fat complex cell outputs, that could be thinned.

θ_i, θ_j, e.g. $|\angle\theta_1, \theta_2| \leq 22.5^0$) and zero for larger angles. Conversely, i_{θ_1,θ_2} is maximal when $|\angle\theta_1, \theta_2| = \pi/2$ and minimal when $\theta_1 = \theta_2$. Again, the steady state solution is used as input to the next stage:

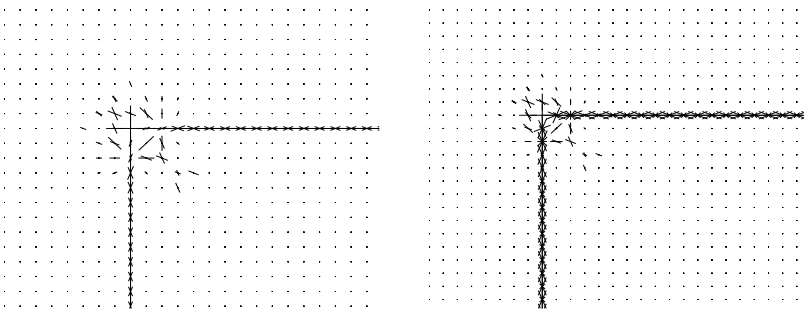
$$Y_\infty^\theta[i,j] = \left[\frac{bC^\theta - dE^\theta}{a + C^\theta + E^\theta}\right]^+ \quad C^\theta = \sum_{k=0}^{N-1} e_{\theta,\frac{k\pi}{N}} X^{\frac{k\pi}{N}}[i,j], \quad E^\theta = \sum_{k=0}^{N-1} i_{\theta,\frac{k\pi}{N}} X^{\frac{k\pi}{N}}[i,j] \quad (31)$$

On the one hand this process helps remove edges that appear in multiple orientations in one single location by favoring the strongest among them, and it helps end-stopped cells perpendicular to ending lines become more pronounced: as was described in the previous stage, cells with orientation parallel to a line ending are not active, due to the inhibition from their active *spatial* neighbors. Therefore, they cannot contribute any inhibitory term in the evolution equation of the cells with orientation perpendicular to the line, which receive mainly excitation, and thereby become enhanced.

3.1.6 Stage VI: Spatial Cooperation

After the previous competitive stages, a cooperative stage is used to detect perceptually important edges, using the previous stage outputs. For this purpose, a mechanism was proposed that gives a measure of edge saliency for all orientations at each point by integrating evidence in favor of a curve from its neighborhood. This role was assigned to cells that have been found to respond to illusory contours in area V2 [48]; such cells respond even when there is no image variation at the center of their receptive field, provided there are well aligned stimuli in the rest of their receptive fields; the existence of such cells had been conjectured by S.Grossberg before their discovery.

The proposed way to gather information in favor of an edge is to convolve the output of stage V cells (Y^θ)with appropriate filters, so that active cells lying on smooth contours



(a) (b)

Figure 15: (a) The outputs of stage V cells compared to the outputs of (b) stage IV cells for the input image. The finest scale results are shown.

passing through the point of interest increase the evidence in favor of a contour passing through it. In fig. 16 we show the weights given to the outputs of stage V cells of all orientations in the neighborhood of an horizontal cell, located at the center: The 'evidence'

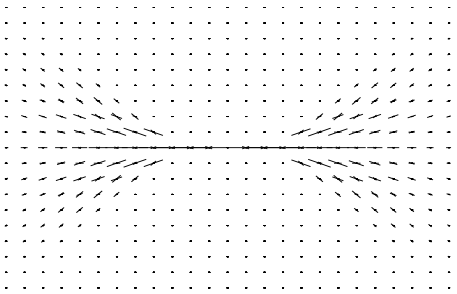


Figure 16: The shape of the *lobes* used for saliency detection in the horizontal direction, at stage VI. The length of the needles corresponds to the weights assigned to edge elements in the specific orientation and position.

offered by a stage-V cell at location (p, q), with orientation θ in favor of an edge passing through an horizontal cell, located at $(0, 0)$ is calculated by multiplying its output with the following quantity:

$$w[p, q, \theta] = \exp(-\beta(p^2 + q^2)) \exp\left(-\mu\left(\frac{q}{p^2}\right)^2\right) \cos^{\gamma}\left(\theta - \mathrm{sgn}(p) \arctan\left(\left|\frac{2q}{p}\right|\right)\right) \quad (32)$$

The formula used is similar to the one in [39], apart from some minor modifications made for the sake of simplicity. The term $\exp(-\beta(p^2 + q^2))$ punishes the distance from cell $(0, 0)$,

the term $\exp\left(-\mu\left\{\frac{q}{p^2}\right\}^2\right)$ punishes non-collinearity of the cell (p,q) positions with the horizontal cell's orientation, while the term $\cos^\gamma\left(\theta - \mathrm{sgn}(p)\arctan\left(\left|\frac{2q}{p}\right|\right)\right)$ punishes highly curved edges (see [75], p.440-441, where this relation is explained for somehow different connection strengths). The parameters μ, β, γ determine the shape of the so-called *lobes*. These functions are expressing in a heuristic way the desired nature of the interconnections between the neurons and should not be taken as a strict rule; in section 4.6.2 a well-founded mathematical model of such connections is reviewed. The activation, h, of each lobe is calculated as the weighted sum over orientations and neighboring positions of the activations $Y^\theta(i,j)$ of stage V cells; for the left/right lobes of a cell with horizontal preference, located at $(0,0)$ we have:

$$\begin{aligned} h^{0,+}[0,0] &= \sum_{p>0,q,\theta} w[p,q,\theta]Y^\theta[p,q] \\ h^{0,-}[0,0] &= \sum_{p<0,q,\theta} w[p,q,\theta]Y^\theta[p,q] \end{aligned}$$

A rotation of the coordinate system by θ can give the corresponding formula for $h^{\theta,\pm}$.

Stage VI cells become active when they receive positive input from both lobes. A product, or a minimum operation, is necessary, since by addition this process will activate cells close to line endings, irrespective of whether they are between line endings (as they should be) or not. This is somehow tricky to write down in 'biological notation', since the common model is that a cell adds its inputs and does not multiply them, as is needed to satisfy the above description. Strangely enough, an addition of the thresholded outputs of the lobes has been commonly used which is like using an OR operator instead of an AND operator (if any of the two lobe outputs is positive and above threshold, the output of the stage IV neuron will be positive). However, in a recent publication [72] an equation was used which implements an AND-like function of the two lobe outputs (appropriately modified for this presentation):

$$\frac{\mathrm{d}Z^\theta[i,j]}{\mathrm{d}t} = -aZ^\theta[i,j] + (b - Z^\theta)H^\theta[i,j]$$

where $H^\theta[i,j]$, the excitatory input to cell i,j,θ is given by

$$\begin{aligned} H^\theta[i,j] &= \left[f(h^{\theta,+}[i,j]) + f(h^{\theta,-}[i,j]) + h^{\theta,+}[i,j] + h^{\theta,-}[i,j] - 2\right]^+ \\ f(x) &= x/(a+x), a << 1 \quad \rightarrow f(x) \simeq H(x) \end{aligned}$$

The term in the brackets is positive only when both $h^{\theta,+}$, $h^{\theta,-}$, are positive, i.e. when both lobes are activated, since in that case $f(h^{\theta,+}) \simeq 1, f(h^{\theta,-}) \simeq 1$ and $H^\theta[i,j] \simeq h^{\theta,+}[i,j] + h^{\theta,-}[i,j]$. Again the steady-state solution is used:

$$Z^\theta_\infty[i,j] = \left[\frac{bH^\theta[i,j]}{a + H^\theta[i,j]}\right]^+ \tag{33}$$

In [75] a different neural architecture was proposed, that implements a multiplication-like operation, using two processing layers instead of a single one; in our implementations we calculate the output directly as the multiplication of the rectified outputs of each lobe i.e. $Z^{\theta}(i,j) = [h^{\theta,-}[i,j]]^{+}[h^{\theta,-}[i,j]]^{+}$. Qualitatively the behavior of the system stays the same, but we avoid many of the intricacies of the above approaches.

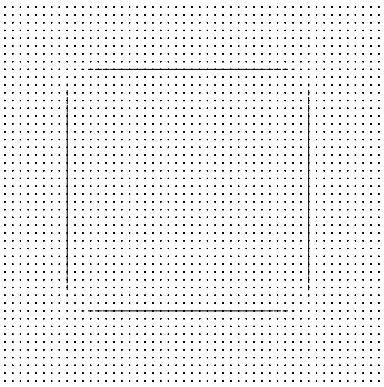


Figure 17: The outputs of stage VI cells

3.1.7 Summary

In summary, the B.C.S. is in charge of edge detection, linking and enhancement and boundary formation, i.e. it processes all the *form*-related information in the image. Quoting Grossberg, we can summarize the function of the VI-IV loop saying that 'it senses perceptual groupings with enough inertia to reach non-zero steady state, 'quenching' insufficient edges' and that it 'receives 'analog' input and has 'digital' output'. This is a distinctive behavior in the evolution of cooperative systems, where the stronger hypotheses -corresponding to states of the system with higher probability- 'exaggerate' the evidence that supports them and dominate the evolution of the system, at the expense of weaker alternatives. The mechanism for achieving this is the use of feedback and the cooperative behavior of the loop between the stages IV-VI.

3.2 The Feature Contour System (F.C.S.)

The F.C.S operates in a complementary way to the B.C.S.; while the B.C.S. detects, thins and amplifies coherent edges, the F.C.S. spreads the activations of On-Off/Off-On cells in an isotropic way, apart from the places where a corresponding B.C.S. cell is active, where the diffusion process is stopped. This is performed in a separate cell array, termed *syncytium* by S. Grossberg, and it is conjectured that it is related to the blob cells in area V1. These cells, termed *Filling-In Domains- FIDO's* are supposed to interact by influencing

each other's potential in a diffusive way (each cell's potential is set to the average of its neighbors) unless a B.C.S. signal blocks their interaction. The fact that there are 3 different color channels with 2 different types of cells (On-Off/Off-On) is accounted for by using 6 syncytia, when processing color images, and 2 when processing gray-level images. The B.C.S. signals delivered to the syncytia are common, since at complex cells fusion of color edges is performed. The equations used for the activation of the On-Off/Off-On syncytia cells $S^{\{\pm\}}$ for the grey-level image case are:

$$\frac{\mathrm{d}}{\mathrm{d}t}S^{\{\pm\}}[i,j] = -aS^{\{\pm\}}[i,j] + cX[i,j]^{\{\pm\}} + \sum_{[p,q]\in N_{i,j}} (S^{\{\pm\}}[p,q] - S^{\{\pm\}}[i,j])P_{[p,q],[i,j]} \quad (34)$$

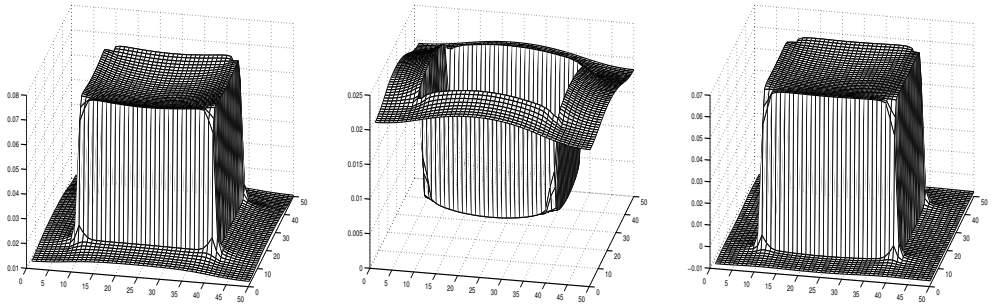
$$S_{\infty}^{\{\pm\}}[i,j] = \frac{cX^{\{\pm\}}[i,j] + \sum_{[p,q]\in N_{i,j}} S_{\infty}^{\{\pm\}}[p,q]P_{[p,q],[i,j]}}{a + \sum_{[p,q]\in N_{i,j}} P_{[p,q],[i,j]}} \quad (35)$$

$$P_{[p,q],[i,j]} = \frac{\delta}{1+\epsilon(Y[p,q]+Y[i,j])}, \quad N_{i,j} = \{[i-1,j],[i+1,j],[i,j-1],[i,j+1]\}$$

In these equations $X^{\{\pm\}}[i,j]$ stand for the On-Off / Off-On syncytia cell outputs, $Y[p,q] = \sum_{\theta} Y^{\theta}[p,q]$, where $Y^{\theta}[p,q]$ is the output of the B.C.S. system after the competition process at stage V, and the quantity $P_{[p,q],[i,j]}$ is a decreasing function of $Y[p,q]$ signaling the absence of an edge between the pixels $[p,q]$ and $[i,j]$. Ignoring the a, X terms, the steady state potential $S[i,j]$ for cell $[i,j]$ in (35) can be seen as a weighted average of the steady-state values of its neighboring cells $S[p,q]$; $P_{[p,q],[i,j]}$ gives higher weights to neighbors that have no edges separating them from pixel $[i,j]$ and lower for values when an edges interferes, reflecting the fact that diffusion is inhibited across edges. The results of the F.C.S. for the input image are shown in fig. 18.

The output of the F.C.S. system is the difference between the steady state values of the On-Off/Off-On syncytia $O_{i,j} = S_{i,j}^{+} - S_{i,j}^{-}$. Equation (34) allows the formation of a smooth, global percept due to the diffusion process, that retains however the forms in the image, because of the B.C.S. intervention. The similarity with anisotropic diffusion [79] is striking, even though there are some differences: (a) the edges here are known a-priori, and are not estimated in parallel with the diffusion process (b) the diffused quantity is the On-Off cell activations and not the image intensity. In fig. 18 the steady states values of the F.C.S. syncytia in response to the input image are shown.

The output of the F.C.S. is the perceived image; actually, what is proposed is that the existence of an edge cannot be perceived, unless it can stop the spreading of the On-Off cell activations around it during the diffusion process. This claim is summarized by the aphorism 'all edges are illusory' [37] meaning that the visual system cannot differentiate between illusory and contrast based contours, but perceives them only when they succeed in bounding the diffusion process inside a closed border. This conjecture facilitated the explanation of various depth-related and binocular rivalry phenomena [27, 33] while by-passing many problems in the B.C.S. system wherever spurious edges were formed, by assuming that they would not be finally perceived by the visual system.



(a) (b) (c)

Figure 18: (a) On-Off syncytium (b) Off-On syncytium (c) difference of two syncytia (perceived surface)

3.2.1 B.C.S./F.C.S. and the COCE illusion

The F.C.S. diffusion scheme described above has been used to explain a wide variety of brightness illusions; for a recent and comprehensive article see [60]. We shall briefly show how this model deals with the Craik- O'Brien Cornsweet - COCE illusion, presented in [41], in order to demonstrate how the B.C.S./F.C.S. can account for a brightness illusion.

The image shown on the left in fig. 19 is perceived as a bright rectangle next to a darker rectangle, on a dark background. However, by covering a few pixels in the middle of the image as is seen on the right of fig. 19, we can see no important difference in the brightness of the rectangles, even though it is the same image (the reader can verify this by putting a pencil in the middle of fig. 19(a). The only change is along the line joining the two

(a) (b) (c)

Figure 19: (a) The image used for the COCE illusion and (b) the same image, when the middle line is covered by a dark band (c) A cross section of image (a) along the horizontal axis

rectangles, where the left rectangle is brighter than the right, as shown in a cross-section in fig. 19(c). If the perceived surface brightness were equal to the mean image intensity within each box, the difference due this narrow width change would not be perceivable in the rest of the boxes. However, the visual system processes only contrast information; therefore, the image area determining the perceived brightness is almost 50% different due to the intensity variation in the line separating the two 'boxes', as shown in fig. 20. In fig. 21 the way the

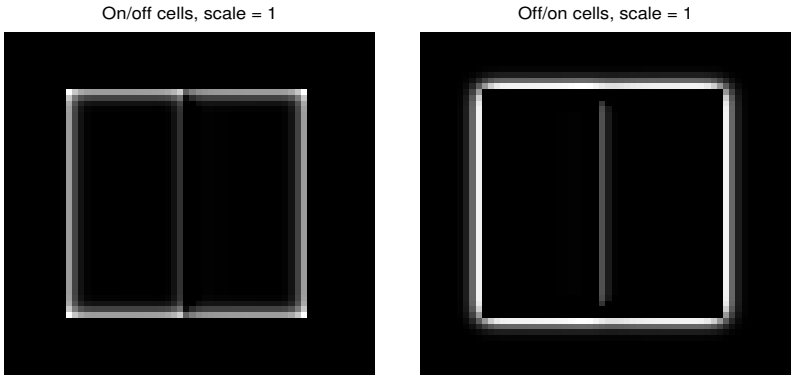


Figure 20: On-Off/Off-On cell responses for the image shown previously

B.C.S./F.C.S. responds to the input image is shown, which helps see both the competition-cooperation loop in action, and the importance of using a diffusion process (filling-in) to model brightness perception.

3.3 Color Channels and Multiple Scales

The model is extended in a straightforward way to deal with color images, which are treated in the Black/White, Red/Green, Blue/Yellow channels: three types of On-Off/Off-On cells are used, which send their output to different simple cells; the outputs of the latter are fused at complex cells, according to equation (30). Color information is stored at separate F.C.S. syncytia, two for each color channel. There is a common B.C.S. output, wherein the color sensitive F.C.S. syncytia diffuse separately their activity, resulting in the form-sensitive perception of a color image.

Multiple scales are treated separately, and the only difference in the equations of the system is that the sizes of the filters are appropriately scaled. For each scale there is one B.C.S. layer and one F.C.S. layer (each with two, or more, filling-in syncytia). The perceived image is derived by an addition of each layer's outputs.

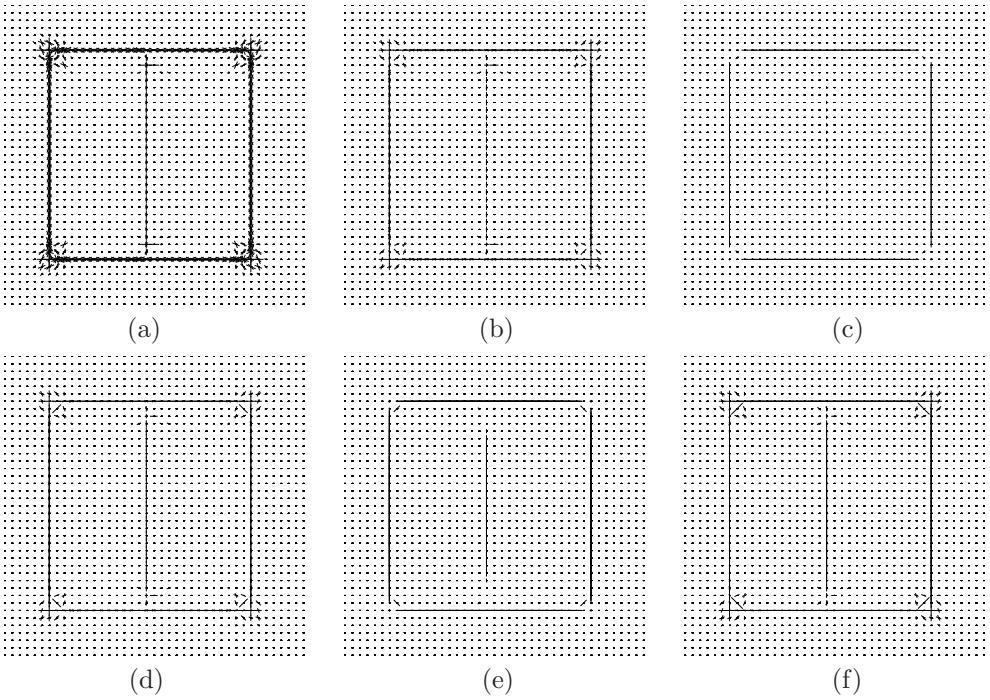
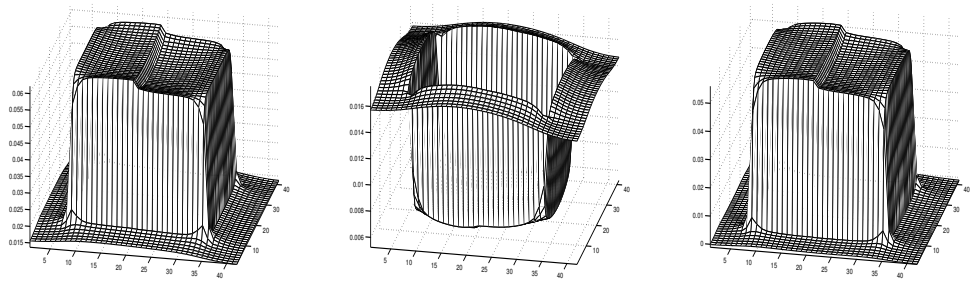


Figure 21: (a) Complex cell outputs (magnified ×4) (b) Stage IV outputs (iteration 1, magnified ×4) (c) Feedback signal, iteration 1 (d) Stage V outputs (iteration 2) (e) Feedback signal, iteration 3; The feedback signal in the middle has been enhanced compared to (c); this is the effect of 'closing the loop' between stages IV-VI (f) Stage V outputs, iteration 4.

3.4 Comments on the B.C.S./F.C.S. model

There are many intuitively appealing ideas in this system, but it has been hard to exploit all of its potential global characteristics, probably due to various choices in the system's subparts. In our implementations we faced many problems, since so many parameters and interdependent stages are involved, that the whole system becomes hard to tame; it has not been possible to achieve a consistent behavior of the system even for a limited variety of images. We shall initially mention our concerns about each specific part of the system, proposing what we suggest works better, as well as some comments about some changes in the overall architecture at the end. Our proposed model is presented in the following section.

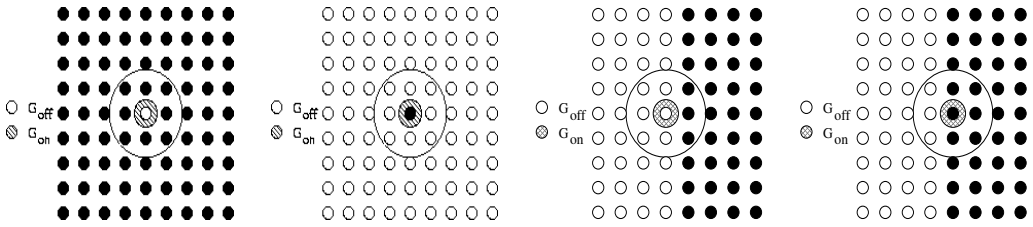


(a) (b) (c)

Figure 22: (a) On-Off syncytium (steady-state) (b) Off-On syncytium (c) On-Off minus Off-On syncytia outputs (perceived surface)

3.4.1 Stages I-III

Even though there cannot be any doubt about whether the functions performed at these stages have a correlate in the visual system, there are some subtleties concerning the specific equations used to model the functions of these stages. As a simple 'sanity check', consider the functions determining the activations of On-Off/Off-On cells. A reasonable assumption is that a white spot on a black background elicits the same On-Off cell response with that of an Off-On cell when presented with a black spot on a dark background as shown in fig. 23(a),(b); this should be the case also with On-Off/Off-On cells lying on opposite sides of a step edge, shown in fig. 23(c),(d). One could argue that no two cells are the same



(a) (b) (c) (d)

Figure 23: The test stimuli used in the 'sanity check' described in the text

etc., but since we are making the simplification of using On-Off cells as contrast detectors, this is something reasonable to ask for. According to (3.1.1) the steady-state outputs for an On-Off/Off-On cell are given by:

$$U^{On-Off}[i,j] = \left[\frac{bC_{i,j} - dE_{i,j}}{a + C_{i,j} + E_{i,j}}\right]^+ \quad U^{Off-On}[i,j] = \left[\frac{bE_{i,j} - dC_{i,j}}{a + E_{i,j} + C_{i,j}}\right]^+$$

where $C = G_{\sigma_1} * I$, $E = G_{\sigma_2} * I$, $\sigma_1 > \sigma_2$. Assuming, without loss of generality, that the On-Off cell excitatory Gaussian has one pixel wide 'support' and the input spot is one pixel wide we have for a On-Off cell located ideally above a white-on-black spot at $(0, 0)$ (fig. 23(a)):

$$U^{On-Off} = \frac{b \cdot 1 - d \cdot G_{\sigma_2}[0,0]}{a + 1 + G_{\sigma_2}[0,0]}$$

and for an Off-On cell located on a black-on-white spot (fig. 23(b)):

$$U^{Off-On} = \frac{b \cdot (1 - G_{\sigma_2}[0,0]) - d \cdot 0}{a + (1 - G_{\sigma_2}[0,0]) + 0} \tag{36}$$

Taking $U^{On-Off} = U^{Off-On}$ results in $(b - d)a = (b + d)(1 - G_{\sigma_2}[0, 0])$.
For the case of On-Off /Off-On cells lying on opposite sides of a step edge (fig. 23(c)/(d)), we have

$$U^{On-Off} = \frac{b \cdot 1 - d(s_{1/2} + c)}{a + 1 + (s_{1/2} + c)} \quad U^{Off-On} = \frac{b \cdot (1 - (s_{1/2} + c)) - d \cdot 0}{a + (1 - (s_{1/2} + c)) + 0} \tag{37}$$

where c and $s_{1/2}$ are given by

$$c = \sum_{k=-N}^{N} G_{\sigma_2}[0, k], \quad s_{1/2} = \sum_{m=1}^{N} \sum_{k=-N}^{N} G_{\sigma_2}[m, k] \tag{38}$$

Equations (37) are identical to the previous set of equations, if we replace $G_{\sigma_2}[0, 0]$ by $s_{1/2}+c$, so the condition obtained by setting $U^{On-Off} = U^{Off-On}$ is $(b-d)a = (b+d)(1-(s_{1/2}+c))$, which necessitates that $s_{1/2} + c = G_{\sigma_2}[0, 0]$; this is however impossible since $c \geq G_{\sigma_2}[0, 0]$ by definition and $s_{1/2} > 0$. One could use different parameters b', d' (instead of b, d) for the Off-On model, but this is a counterintuitive choice that serves only to make the model 'pass the test'. It seems also likely that a third example would make this variation break down as well.

The problem emerges from the introduction of the normalization factor in the computation of the output, whose purpose is to allow the system to function in the same way, under varying input magnitudes [26]. If instead of the feedforward model of retinal cells presented previously one uses recurrent connections, as used e.g. in the classical model of retinal cells of Hartline & Ratliff [43], the above problems do not occur, while normalization is still performed.

In [39] three different models of On-Off cells were presented and two more proposed; we consider such a plethora of models rather discouraging, given that none of them is finally favored by the authors, while they face the problems that were presented above in one form or the other. Another problem we encountered has to do with the emergence of spurious simple cell responses and is presented in [72], so we refer the interested reader for details there.

If we ignore the need for normalization and use only the feedforward input sent to each cell as determining its output, we can view the cascade of Stages I and II as performing something qualitatively similar to Gabor-filtering the image; take for example equation (27):

$$Y^{s_3}_{\pi/2} = [(X^{On-Off} * G_L + X^{Off-On} * G_R) - (X^{Off-On} * G_L + X^{On-Off} * G_R)]^+ \quad (39)$$

This can be written as

$$\begin{aligned} Y^{s_3}_{\pi/2} &= \max((X^{On-Off} - X^{Off-On}) * G_L + (X^{Off-On} - X^{On-Off}) * G_R, 0) \\ &= \max((X^{On-Off} - X^{Off-On}) * (G_L - G_R), 0) \\ &= \max(I * (G_{\sigma_1} - G_{\sigma_2}) * (G_L - G_R), 0) \end{aligned} \quad (40)$$

since

$$X^{On-Off} - X^{Off-On} = [I * (G_{\sigma_1} - G_{\sigma_2})]^+ - [I * (G_{\sigma_2} - G_{\sigma_1})]^+ = I * (G_{\sigma_1} - G_{\sigma_2}) \quad (41)$$

The Fourier transform of the combined filter $(G_{\sigma_1} - G_{\sigma_2}) * (G_L - G_R)$ is the product of the Fourier transforms of its constituents, which are given by

$$\begin{aligned} \mathcal{F}[G_{\sigma_1} - G_{\sigma_2}] &= \mathcal{F}\left[\frac{\exp\left(-\left\{x^2/2\sigma_1^2 + y^2/2\sigma_1^2\right\}\right)}{2\pi\sigma_1^2}\right] - \mathcal{F}\left[\frac{\exp\left(-\left\{x^2/2\sigma_2^2 + y^2/2\sigma_2^2\right\}\right)}{2\pi\sigma_2^2}\right] \\ &= \exp\left(-\left\{\frac{\omega_x^2\sigma_1^2 + \omega_y\sigma_1^2}{2}\right\}\right) - \exp\left(-\left\{\frac{\omega_x^2\sigma_2^2 + \omega_y^2\sigma_2^2}{2}\right\}\right) \end{aligned} \quad (42)$$

$$\begin{aligned} \mathcal{F}[G_L - G_R] &= \mathcal{F}\left[\frac{\exp\left(-\left\{(x-c)^2/2\sigma_x^2 + y^2/2\sigma_y^2\right\}\right)}{2\pi\sigma_x\sigma_y}\right] - \mathcal{F}\left[\frac{\exp\left(-\left\{(x+c)^2/2\sigma_x^2 + y^2/2\sigma_y^2\right\}\right)}{2\pi\sigma_x\sigma_y}\right] \\ &= \exp\left(-\left\{\frac{\omega_x^2\sigma_x^2 + \omega_y^2\sigma_y^2}{2}\right\}\right)\exp(-jc\omega_x) - \exp\left(-\left\{\frac{\omega_x^2\sigma_x^2 + \omega_y^2\sigma_y^2}{2}\right\}\right)\exp(jc\omega_x) \\ &= -2j\sin(\omega_x)\exp\left(-\left\{\frac{\omega_x^2\sigma_x^2 + \omega_y^2\sigma_y^2}{2}\right\}\right) \end{aligned} \quad (43)$$

The first filter has only a real component, the second only an imaginary and their product only imaginary; these are shown in fig. 24. For an appropriate choice of $\sigma_1, \sigma_2, \sigma_x, \sigma_y$, this is like the Fourier Transform of a Gabor filter, shown in fig. 24(d); of course not any combination of parameters leads to Fourier Transforms similar to those of Gabor filters. The same analysis can be applied to the even-symmetric case of the filters detecting bars (25), by suitably choosing the σ's. Given the optimality results concerning Gabor filters [16] we would prefer to use these to model the cascade of On-Off and Simple Cells, since they have been extensively studied as models of simple cells, and less parameters are involved compared to modeling separately each stage. One should, however, keep in mind the fact

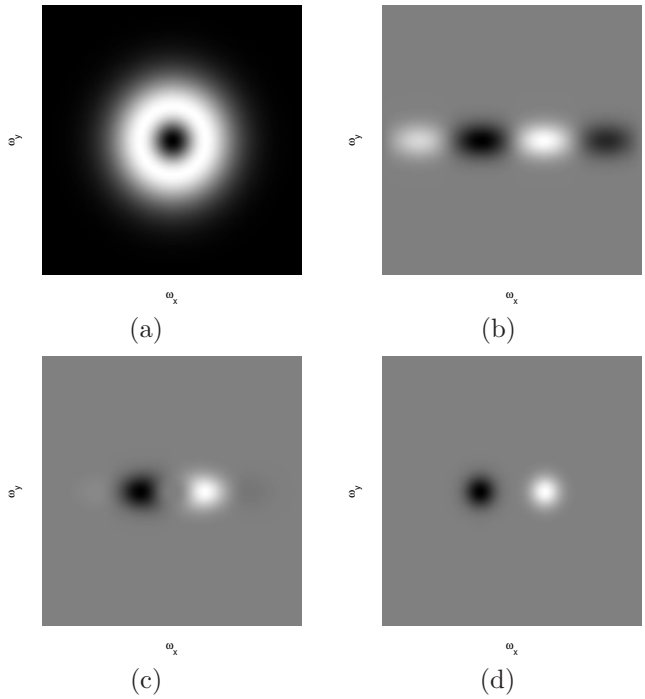


Figure 24: (a) Real component of Fourier Transform of $G_{\sigma_1} - G_{\sigma_2}$, $\sigma_1 = .3, \sigma_2 = .6$ (b) Imaginary component of FT of $G_L - G_R$, $s_x = .1, s_y = 1, c = 1/2$ (c) Imaginary component of FT of $(G_L - G_R) * (G_{\sigma_1} - G_{\sigma_2})$ (d) Imaginary component of FT of Gabor filter, $\sigma = 1, f = 1$

discussed in subsection 2.3.3 that we have to modify the classical even-symmetric Gabor filters to have zero DC response, as done e.g. in [67]. Finally we mention that using only odd-symmetric filters that detect step edges the system is bound to have poor performance at detecting roof edges [78, 65].

To account for the normalization in the shunting equations of the B.C.S. model one could use the system proposed in [12, 13] to normalize the outputs of even symmetric and odd symmetric filters using shunting inhibition, as was discussed in 2.1.2. This system uses recurrent connections between neurons of the same layer, and can operate, without saturating, over a wide range of input stimuli. This will be presented in the following section.

3.4.2 Stages IV-V

In our implementation of the orientational and spatial competition mechanism we faced significant problems in determining the constants, like the spreads of the filters, the tonic

input strength etc. We tried using adaptive expressions, e.g. using fractions of median, mean or maximum values, local estimates of edge variance, but they all failed when presented with new examples. When we say failed we mean that they did not manage to strike a balance in achieving the following, reasonable goals for any edge detection/enhancing system: (a) Sharp edges (b) No self-creation of edges (c) No shrinking of edges while correctly detecting line endings. It is probably a hint of the difficulty of the problem that we have never seen the same parameters used in two different publications, while many times the equation is modified by the omission of one term or the other, depending on the subject of the specific paper where it appears, even though its 'semantics' remain the same. In papers with real-world applications [39, 72] the parameters are set to peculiar values, in order to achieve the desired performance.

We believe the above problems arise mainly due to two causes:

- Edge thinning and line-end detection are incompatible.
- Edge thinning is an inherently recurrent process.

We explain these points in detail below

Edge thinning and line-end detection are incompatible: The use of a single type of cells to perform both processes of line-end detection and edge thinning seems to be too demanding: the edge thinning process can be accomplished using filters that change perpendicular to the edge direction (so as to detect and 'punish' broad edges) while the end-stopping mechanism is detecting changes along the border's direction: using an isotropic Gaussian, as is done in stage IV filter detects changes in both directions, however it does not do *well* in detecting any of them.

Apart from that, end-stopped cells should become active at edge terminations (for the case of asymmetric end-stopped cells) and places of increased edge curvature (symmetric). However, using the tonic input term in (30) results in having end-stopped cells active both at edge terminations *and* on every side of an edge, in the perpendicular direction. The reason why this happens is that during the thinning process cells that are near an edge's peak and have the same orientation as the edge get inhibited, while the ones that are perpendicular stay active, as they receive no excitation/inhibition, apart from their (excitatory) tonic input. In this way, cells with orientational preference perpendicular to an edge are active in its surroundings, all along the edge. Apart from having no biological evidence that end-stopped cells do behave this way this has implications in the subsequent edge grouping process. To by-pass the resulting problems *another* mechanism is proposed, named spatial impenetrability [28]; we believe that this is more of a heuristic solution to a problem posed by the model itself than an actual biological mechanism for a problem of the visual system.

A less confusing approach to line-ending detection seems to us to be the one used in [45, 46]: the end-stopped cells are used exclusively for the detection of line endings and high curvature points, freeing them from the need to be active all along a border. This is achieved by applying relatively simple filters (differences of offset gaussians) on the complex cell outputs, and separating the filters into groups, according to their orientational selectivity

and whether they detect the left or the right part (bottom/top etc.) of the ending edge. These are subsequently input to a line-ending grouping process in order to detect illusory contours, at a later processing stage. The edge thinning process can be performed separately; actually it is not mentioned by the authors in [45], who simply choose the maximum response of the complex cell activations as input to their line-ending detection process.

Edge thinning is an inherently recurrent process: Apart from the feedback term, that comes from another layer and is considered constant, all other connections are feedforward, so that it is possible to use the steady state solution. However, horizontal connections exist in each processing stage in the visual pathway, while many recent successful models of the behavior of cortical cells [12, 13, 7] have been using recurrent connections. Even though S. Grossberg has been a pioneer in research for recurrent network dynamics [29, 14] recurrent networks have been employed mainly in his work concerning models of memory and pattern recognition [32, 27, 31]; however in [80] a variation of the B.C.S./F.C.S. model using recurrent interactions was proposed and applied to 1-D signals, while in [83] a novel architecture that uses recurrent connections for spatial and orientational competition was presented, which was however not in the usual B.C.S./F.C.S. framework.

Using feedforward connections the whole process boils down to a combination of linear filtering, compressing and thresholding the previous stage's output, which cannot provide an efficient edge thinning mechanism. Edge thinning can be easily thought of as a recurrent process; intuitively, it is accomplished by taking away small pieces of edges until we can no longer remove any without breaking an edge and changing the topology of the edge map. In case we try to do this in a feedforward manner by convolving with a Difference of Gaussians and thresholding, we have to choose a priori the width of the inner and outer Gaussian, which are related to the assumed shape of the input edge and which determine the sharpness of the output edge. Even if these quantities are known, the performance of this process is no better than that of a band-pass filter and hence cannot result in sharp (high-frequency) outputs. One possible solution would be to re-filter the output at a next layer and so on; however this can be done in a single layer using recurrent connections between neighboring neurons, which is an architecture commonly observed in the visual cortex and studied extensively in artificial neural networks. With the same rationale, one could perform the spatial competition process in parallel with the orientational competition process, resulting in a synchronous thinning of the edges, both in space and orientation.

To clarify this point, in fig. 25(b) we see the evolution in time of the outputs of a recurrent system of neurons (located along axis X) in response to the input it receives from a previous stage, shown in fig. 25(a). The one-dimensional input can be considered as the profile of complex cells along a cross-section perpendicular to the direction at which edge detection is performed. Initially ($t = 0$) the output of the system is broad, which means we have a fat edge, while our goal is to derive a well localized edge. Since this model will be presented in detail in the following section for the 2-D case, we simply demonstrate its behavior here. Using various possible edge profiles, we observed that when the input stimuli are narrow enough recurrent networks tend to converge to 'binary' states where only one of the neurons is active in a neighborhood and the other become suppressed, i.e. they exhibit a winner-take-

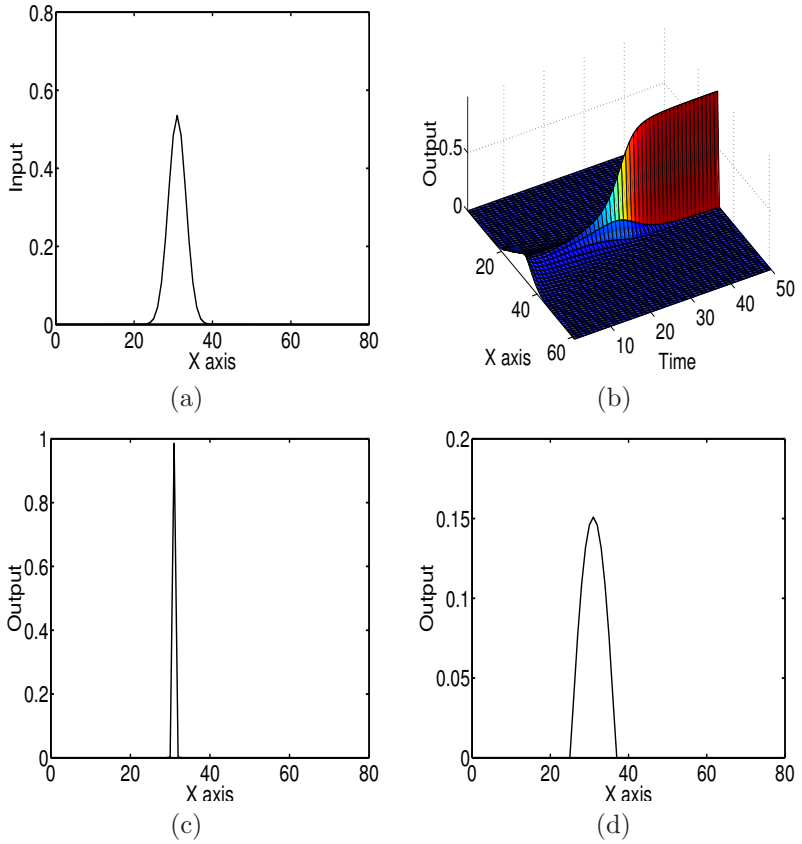


Figure 25: (a) Input profile (b) Evolution in time of recurrent network outputs (c) Steady-state solution of recurrent network (d) Steady state solution of feedforward network (the best obtained, by thresholding the outputs of convolution with Difference-of-Gaussian filters)

all behavior. Using inputs of width larger than that predicted by the recurrent connection strengths results in worse performance, which remains, however, always better than that of the feedforward system. What narrow and broad means is relative to the connection pattern among neurons: if distant neighbors are connected, broad stimuli can be dealt with, while if only interactions among very proximal neighbors are allowed, the system cannot receive a broad input and have a sharp output.

3.4.3 Stage VI

The use of feedback *after* the convergence of each layer to its steady-state may seem plausible in the case only feedforward connections are used, where at each competition stage a

linear filtering-like operation is applied. However, in case recurrent connections are used, one should compute the feedback term and let it interact in parallel with the competition process, in order to prevent it from being driven to a local minimum, that ignores higher-level information. Apart from the above, we consider more appealing the simpler and well founded model of Elastica, presented in 4.6.2, as a basis for the calculation of lobe shapes than the ad-hoc formulae presented in the previous subsection.

3.4.4 Feature Contour System

As far as the F.C.S. is concerned, it was not clear from Grossberg's publications whether the B.C.S. outputs are placed in between the F.C.S. cells, or are considered to be on top of them. Commonly in computer vision [24, 93] the line process is placed in between surface cells, and this is what the the architecture proposed in [28] sounds like. However the formula (36) implies B.C.S. cells are placed on top of the F.C.S. cells. Placing the B.C.S. cells between F.C.S. cells gives better results and is more rational (a step edge, like those detected by the B.C.S. network is between two pixels, not on top of one), so we used this in our implementations.

Another technical, but important problem is that unless the edge map is perfectly thinned, the activations of F.C.S. neurons get 'trapped' between adjacent edges (that correspond however to a single image discontinuity), and the diffusion process does not spread the activation of the cells closest to the edges.

For example, as shown in the competition results of fig. 15, it was not possible to eliminate the diagonal edges near the corners using the specific spatial/orientational competition process described previously. Some of these edges are due to the changes in image intensity (like the $\pi/4$ edge in the top left corner), while some others are due to the used model of simple cells that causes spurious edges, as was mentioned earlier. The steady state values of the diffused syncytia reflect this problem, where the activations near corners are much larger than those of their neighbors inside the square, even though there is no true line separating them. For special settings of the parameter values the desired performance can be achieved for a specific image (as was done to produce fig. 18), but the problem remains. This problem occurs mainly because On-Off cell activations (the input to F.C.S.) are highly localized near edges, so the B.C.S. results have to be perfect (no spurious edges, one-pixel wide edge maps) to achieve the desired behavior. If the image intensity is used instead of On-Off cell activations, these problems are by-passed, even though many of the interesting phenomena of brightness perception cannot be explained. An approach lying somewhere in the middle is that used in [80], where both luminance and contrast driven signals are used.

3.4.5 Overall architecture

Apart from the above comments, we should mention the misuse of multiple scales, the absence of feedback from the F.C.S to the B.C.S. in the monocular case and the complexity of the B.C.S. system as a whole.

Multiple Scales

The way multiple scales are treated in the B.C.S./F.C.S. is contrary to the way most people working in computer vision would think about multiscale analysis; the only reference where we found some cooperation between scales is in [34] and this is done only as late as at stage VI. Decoupling the processing at different scales and superimposing the steady state-outputs of the F.C.S. systems seems to be an inefficient approach, that can add little to the system's performance. Coarse-scale edges should be used to drive the competition at finer scales, adding robustness to noise, while avoiding local minima. This could be implemented by adding a constant or gain modulating input in the competition process (30) of fine scales that comes from coarser scale processes, so that locations where coarse scale edges have been detected could be favored.

Feedback from the F.C.S. to the B.C.S.

In the B.C.S./F.C.S. model the role of the F.C.S. is secondary, relative to that of the B.C.S.: the B.C.S. detects and localizes the edges present in the image, and the F.C.S. simply diffuses its activity inside the compartments formed by the B.C.S. The flow of information from the F.C.S to the B.C.S. has been proposed only for binocular vision, in the framework of the FACADE theory and its main computational role in that case is in the formation of amodally completed surfaces.

However, the F.C.S. could be let to interact with the B.C.S., so that it can play a role in the contour formation process, enhancing the more visible contours and exploiting region-based information. Apart from that, empirically, by ignoring the F.C.S. outputs in the contour formation process we have observed that contours tend to get deformed in favor of smoother ones, due to the influence of the feedback term, even when there exists a sharp edge in the image.

Less processing stages

The B.C.S. system seems initially to be complicated and loaded with many functions, even though the ideas behind it are simple and appealing. A simpler and easier controllable architecture would be achieved by devoting a layer (a) to edge detection and normalization, (i.e. merging stages I-II to a single one) a layer (b) to spatial and orientational competition and a layer (c) corresponding to the high level grouping processes. Another (optional) layer (d) can be used for line-ending detection, which will feed into layer's (c) grouping process, like in [45, 46]. With the exception of layer (d), which has not been implemented, this is the architecture of the model we propose in the following section.

3.4.6 Discussion

Apart from these concerns, it should be made clear that this system is impressive not only because of having been introduced 20 years ago, but also because it is a biologically plausible architecture encompassing a variety of functions in a single and coherent model; specifically, what we found more interesting about this system is that

- The whole system is biologically plausible; apart from the exact form of the equations, or the connections etc., every processing stage comes with an area of the visual cortex to implement it. The architecture of the system is similar to that proposed later by other authors e.g. [66, 68, 75] to model biological segmentation mechanisms and their major differences lie in the network dynamics equations [68], the role [66] and the necessity or not [68] of the F.C.S., as well as the role of a high-level grouping process [75].
- Several aspects of this model appeared very original, when they were introduced later with the proper way to a proper audience by different people. For example, the concept of a bipole-like field was re-introduced later by Zucker & Parent [76, 94] and Field et al. [20] and used by Heydt et al. [45, 46], Guy &Medioni [42] and other researchers [89, 68, 75] to derive edge salience maps. Anisotropic diffusion [79] was a breakthrough in the multiscale analysis and denoising of images, and is closely related to the BCS-FCS interaction proposed previously.

Apart from the above, we believe the most important point is that this model has been extended using the same mechanisms of parallel and distributed cooperation and competition to perform various other mid-level vision tasks, like motion processing, depth analysis and binocular vision. A wide variety of psychophysical phenomena can be explained in its terms ([41, 60, 27, 38]), lending support in favor of its mechanisms. Therefore, understanding the mechanisms of this model can give us a hint for the mechanisms of human vision.

4 A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks

In this section, we propose a model that is based on the same ideas that determined the architecture of the B.C.S./F.C.S. model, yet is more easily controllable and yields better results, when applied to both synthetic and real world images. Our initial efforts concentrated on modifying the original B.C.S./F.C.S. model little by little in order to improve its performance, but it turned out to be easier to 'reinvent' the model. This model components can substitute the B.C.S/F.C.S. components in the rest of S. Grossberg's work on biological vision, so it can serve as a starting point for the incorporation of ideas from the FACADE model of vision in robust and efficient computer vision algorithms. This section complements the exposition of the model proposed initially in [63].

The analogy of the model we propose, shown schematically in fig. 26, with the original B.C.S./F.C.S. architecture is evident, however there is a significant degree of novelty concerning each of the components individually and the system as a whole. The architecture is close to that proposed in the work of T.S. Lee [66], which, however, did not include bottom up edge detection and high level edge grouping processes, while the system was focused on texture segmentation. Our model performs both contour detection and image smoothing so it is different from other biologically plausible models like [75, 68, 46] that deal exclusively with boundary processing.

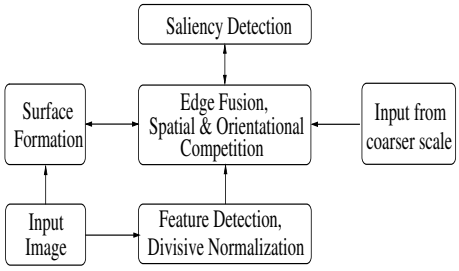


Figure 26: The architecture of our model

4.1 Stage I': Feature Extraction, Divisive Normalization

The first two stages (contrast & feature detection) of the B.C.S. are merged into one, using the Gabor filterbank described in [66]; this filterbank includes even-symmetric filters with zero DC components, which is necessary for modeling simple cells, while the parameters of the filters are chosen to comply with measurements of simple cell receptive fields.

To incorporate contrast normalization, we used the mechanism of shunting inhibition [12, 13], where the feedforward input to each cell is determined from the beginning using a convolution with a Gabor filter, while the shunting term is dynamically changing, based on

the activations of neighboring cells. Concerning the plausibility of this mechanism, in [12] a mechanism is proposed to justify the use of a constant feedforward term when using synaptic excitation and inhibition. Further, it has been questioned [51] whether shunting inhibition is a mechanism that can account at a physiological level for divisive normalization; however it still provides a mathematical way to introduce this mechanism in the evolution equations of a mean firing rate neuron. The neuronal mechanism by which shunting inhibition is performed had been proposed by S. Grossberg (see e.g. [26]); the main difference lies in that in the work of S. Grossberg the shunting term is computed using the outputs of the same cells that contribute to the feedforward term, while in [12] it is computed from the outputs of cells at the same processing layer with the neuron whose output is being normalized. This avoids the problems concerning normalization presented in 3.4 as can be seen by examining the same problems.

The outputs of the neurons contributing to the shunting inhibition of neuron $[i, j]$ are weighted with a Gaussian function centered at $[i, j]$ with a spread equal to the larger spread of the elongated Gaussian used by the Gabor filters. Contributions from different orientations than the neuron's orientation preference θ are given equal weights, so that the equation driving the activity of a simple cell becomes:

$$\frac{\mathrm{d}X^{\theta}_{\{o,e\}}}{\mathrm{d}t} = -aX^{\theta}_{\{o,e\}} + (c - X^{\theta}_{\{o,e\}}) \underbrace{[F * \Psi^{\theta}_{\{o,e\}}]}_{\text{feedforward term}} - X^{\theta,\sigma}_{\{o,e\}} \underbrace{\sum_{\theta,o,e} G * Y^{\theta}_{\{o,e\}}}_{\text{shunting term}} \tag{44}$$

where $\Psi^{\theta,\sigma}_{\{o,e\}}$ are odd/even symmetric Gabor filters, with orientation preference θ, while $Y_{\{o,e\}} = \max(X_{\{o,e\}}, 0)$ are the varying with t outputs of odd/even cells. In case multiple scales are used, the above equation is replicated with appropriately scaled filters.

4.2 Stage II': Contour Formation, Edge Thinning

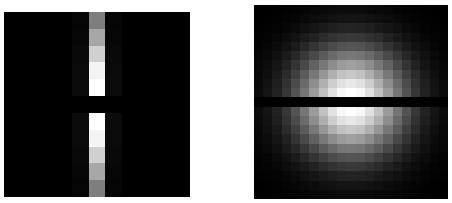
In this stage, simultaneous competition in space and orientation is accomplished using lateral connections. The notation that will be used for these connection weights is $W^{\theta,\phi}_{[i,j]}[k, l]$ which expresses the strength with which the neuron at $[k, l]$ with orientation ϕ inhibits the one at $[i, j]$ with orientation θ.

Qualitatively, connection strengths among neurons of the same orientation should be such that a broad edge is punished while thin edges are not. As was discussed in section 3.4, this cannot be accomplished using isotropic Gaussian functions for the connection strengths among neurons, since this would imply that neurons which are consistent, i.e. have the same preferred orientation and lie along the same line, would inhibit each other. Initially we experimented with connection strengths of the form of an elongated Gaussian, whose major axis is perpendicular to the preferred orientation; this way fat edges could be avoided, allowing a single neuron to be active in its neighborhood. For example for an horizontally oriented neuron located at $[i, j]$ the inhibitory connection strengths with neurons of the same

orientation would be

$$W^{0,0}_{[i,j]}[k,l] = (1-\delta_{i,j}[k,l]) \exp\left(-\left\{\frac{(i-k)^2}{2\sigma_1^2}+\frac{(j-l)^2}{2\sigma_2^2}\right\}\right), \quad \sigma_1 < \sigma_2 \ . \tag{45}$$

shown in fig. 27. The term $(1-\delta_{i,j}[k,l])$ guarantees that a neuron does not inhibit itself.



(a) (b)

Figure 27: Shape of lateral connections between neurons of the same orientation, as (a) according to (45) (b) according to (46)

In [83] a somehow 'smoother' type of interconnections has been proposed for an analogous stage, where two terms come into play: a convolution with an isotropic Gaussian filter that acts in an inhibitory manner and a convolution with a template similar to the bipole filters used in our model at Stage III'; the idea behind this is that even though inhibition comes from an isotropic term, this is counterbalanced by an excitatory term that comes from active neurons in an appropriate constellation. In fig. 27(b) we show an approximation of this type of connections which is used in our model and is given by:

$$\begin{aligned} W^{\theta,\theta}_{[i,j]}[k,l] &= G_{[i,j]}[k,l] - bG^{\theta}_{[i,j]}[k,l] \qquad (46)\\ &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma^2}\right)\\ &\quad -\frac{b}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{(\cos(\theta)(i-k)+\sin(\theta)(j-l))^2}{2\sigma_1^2}-\frac{(\sin(\theta)(i-k)-\cos(\theta)(j-l))^2}{2\sigma_2^2}\right)\\ &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma^2}\right)\\ &\quad -\frac{b}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{(\Pi_{\perp\theta}[(i-k),(j-l)])^2}{2\sigma_1^2}\right)\exp\left(-\frac{(\Pi_{\parallel\theta}[(i-k),(j-l)])^2}{2\sigma_2^2}\right) \end{aligned}$$

where σ_1 is set much smaller that σ_2, so that the non-inhibition effect is kept only for neurons lying along thin lines. What these connection weights guarantee is that due to $G_{[i,j]}[k,l]$ in general a neuron close to another with the same orientation inhibits its neighbor; however, due to the second term, $-bG^{\theta}_{[i,j]}[k,l]$, neurons lying on a line with the same orientation θ as the one which they prefer do not inhibit each other as strongly.

The lateral inhibitory connections among complex cells of different orientational preferences are determined not only by the spatial but also by the orientational relationship between two neurons; two neurons with perpendicular orientation preferences can interact in a different way compared to those which are almost parallel. Such interactions for a neuron located at pixel $[i, j]$ with orientation ϕ and a neuron located at $[k, l]$ with orientation θ can be expressed as

$$W^{\phi,\theta}_{[i,j]}[k, l] = a(\phi, \theta) \exp\left(-\left\{\frac{(i-k)^2 + (j-l)^2}{2\sigma^2}\right\}\right) .$$

Using a radially symmetric function instead of an elongated Gaussian or something similar is useful from a practical point of view (it gives good results) and is exploited in Appendix B to derive a Lyapunov function for the network. According to the B.C.S. model and as described in a previous section, $a(\phi, \theta)$ should be chosen to be maximal for $\theta \perp \phi$ (e.g. using $a(\phi, \theta) = |\sin(\theta - \phi)|$), so that a vertical edge appearing at the same location with an horizontal edge suppresses it, while an edge with an almost similar orientation should not have a significant inhibitory effect. However, we found out that this way it is hard to derive an edge map that is clean in orientation, while corners are broken up. Adapting to our model the scheme in [94], where only the closest neighbors (in orientation) inhibit each other, we finally chose to maximize inhibition for neighbors of similar orientations. The expression $a(\phi, \theta) = a + (1 - a)\cos(\theta - \phi)$ has been used, where a is a constant term (e.g. $1/2$) that guarantees that there will be some inhibition among neurons responding even to perpendicular directions. The angles in the expression above express the orientation preferences of the interacting neurons and have no relation with steerable filters [22], where the angles involved in the expressions correspond to the polar coordinates of pixel $[i, j]$.

The bottom-up input to this stage is estimated by combining the outputs of odd/even symmetric cells in an orientational energy-like term [4, 78], which is used as feedforward input:

$$E^{\theta,\sigma} = \sqrt{(U^{\theta,\sigma}_o)^2 + (U^{\theta,\sigma}_e)^2} . \tag{47}$$

The difference with the model proposed in [78] lies in that simple cell outputs have already been normalized before, which results in a contrast invariant edge detector. The merits of incorporating both even and odd-symmetric filter responses in an edge detector are well-known [78, 4, 44, 65]; such a scheme has been used to model the behavior of complex cells also in [4, 44] and biological evidence in favor of it is reviewed in [17] p. 74-76.

The steady state outputs of the immediately coarser scale, $U^{\theta,\sigma+1}$, are used as a constant term that favors the creation of edges at specific locations. The feedback term $T^{\theta,\sigma}(t)$, that is calculated at the following stage, is used from the beginning of the evolution process, facilitating the timely integration of high-level information; otherwise, if the system is let to converge before using the feedback term, it may be driven to a local minimum and it may be hard to drive it out of it. The evolution of the B.C.S. neuron outputs U has been coupled with the evolution of the F.C.S. outputs, S; the magnitude of the directional derivative $|\nabla S_{\theta^\perp}|$ of S perpendicular to θ helps the formation of sharp edges at places where the gradient of the

reconstructed image is highest. Thereby the occasional shifting of edges due to higher level (Stage III') cues, or the breaking up of corners due to orientational competition is avoided.

The evolution equation for this stage's neuron $[i, j, \theta]$ potential, $V^\theta[i, j]$ is written as:

$$\frac{\mathrm{d}V^\theta}{\mathrm{d}t} = -aV^\theta + (c - V^\theta)I^\theta - (V^\theta + d)\sum_{\theta'}\sum_{k,l} W^{\theta,\theta'}_{[i,j]}[k, l]U^{\theta'}[k, l] \quad (48)$$

$$I^\theta(t) = [c_1E^\theta + c_2U^{\theta,\sigma+1} + c_3T^\theta(t) + c_4|\nabla S_{\theta^\perp}(t)|^2] \; . \quad (49)$$

where $U^\theta[i, j] = g(V^\theta[i, j])$, with :

$$g(V) = \frac{1}{1 + \exp(-c_b(V - 1/2))} \quad (50)$$

The cues determining the excitatory input to the neuron, $I^{\theta,\sigma}$, are (a) bottom up: E^θ, (b) region based: $|\nabla S_{\theta^\perp}(t)|$ (c) coarser scale: $U^{\theta,\sigma+1}$ and (d) top-down: T^θ.

All the quantities in the evolution equation should have a σ superscript to denote the scale to which the neurons correspond, but this has been omitted for the sake of simplicity. It is also assumed that c_b, c and d as well as all the c_i coefficients are positive numbers.

4.3 Stage III': Salience Computation.

The outputs of this layer are calculated as the product of two lobe responses, which are continuously updated, resulting in a process that is parallel and continuously cooperating with Stage-II'. Even though it is not clear how multiplication can be performed in a single cell, there is strong evidence in favor of multiplication being used in mechanisms like gain modulation [84]; see also [80, 75] for a neural 'implementation' of multiplication.

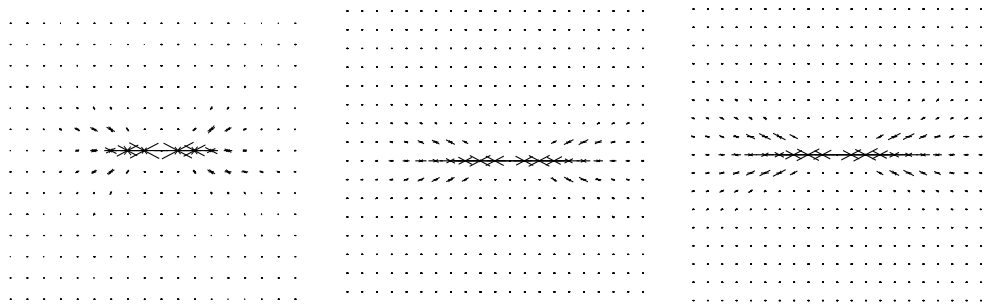
Instead of using formula (32) for the computation of the interaction weights among neurons we preferred using the Elastica model of curves, presented in a following section, as was done in [89]. This requires using only two easily interpretable parameters, one corresponding to the scale of the lobes, and another to their spread. In fig. 28 the resulting lobes for different choices of the parameters are shown.

4.4 Feature Contour System

To avoid the problems presented in section 3.4 the brightness values of the image have been used, instead of the On/Off- Off/On outputs, which allows us to compare the results of our model with other algorithms that use the image intensity as their input, e.g. [79].

A less important architecturally, but significant practically modification was that we consider the B.C.S. neurons as being located between F.C.S. neurons, as in [93], shown also in fig. 29. This facilitates the exploitation of the oriented line-process neurons by blocking the diffusion among two pixels only when there is an edge (close to) *perpendicular* to the line joining them. More formally, this can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}S(i, j) = \sum_\theta (\nabla_{\theta^\perp} S) \cdot (1 - (U^\theta)) \quad (51)$$



(a) (b) (c)

Figure 28: The lobes derived from the Elastica model of curves for parameters (a) $\lambda = 10, \sigma = 0.4$, (b) $\lambda = 10, \sigma = 0.2$,(c) $\lambda = 20, \sigma = 0.2$

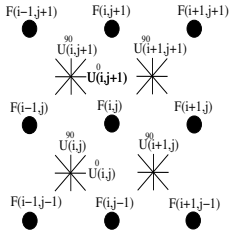


Figure 29: Relative Positions of Surface (F) and Line (U) process neurons

where, as in [79] $S(i, j)$ is always the subtracted quantity in $\nabla_\theta S$. In the implementation these equations have been modified to account for the discrete nature of the neighborhoods and the relative positions of the F.C.S./B.C.S. nodes, but the idea is the same: block diffusion only across edges and not along them.

The leakage term $-aS$ has been dropped since we did not include a data fidelity term $+cF$ in the evolution equation, so there is no steady current that can keep the surface process neuron active. Even though using a data fidelity term is possible, we only initiated the S neurons with the corresponding image intensities, to compare our model with the classical anisotropic diffusion of Perona & Malik [79] on equal grounds.

4.5 Experimental Results & Model Evaluation

The model presented here has been tested on a variety of images, including both synthetic and real world images, taken from the image database of [71]. Its parameters have been tuned initially to respond in a reasonable way to synthetic stimuli, and subsequently it has been tested on real images using the same parameter set. Its performance compares favorably to classical edge detection and diffusion algorithms [11, 78, 82] used in computer

vision. We do not claim however that it is superior to the state-of-the-art in edge detection or anisotropic diffusion [18, 87, 78, 82, 71]; we consider it more important that the modifications we introduced resulted in a simpler and efficient biologically plausible model, that does not largely deviate from the original B.C.S./F.C.S. architecture. We have used a small fraction of the number of parameters used e.g. in [39]; using the original B.C.S./F.C.S. model we did not manage to achieve the same results, even when optimizing its parameters for every single image.

For all the images the same set of parameters has been used, while modifying them did not result in significant changes in the system's performance. It should be noted that our model operates at specific scales, so it may not respond in the same way to stimuli of arbitrary size, while a careful treatment of discretization problems is required.

We first demonstrate the merits of using divisive normalization at an early processing stage; comparing fig. 30(d) to fig. 30(e) we see that after divisive normalization edges of low-contrast regions, like the background pergola or the folds of the shirt become more vivid, while regions of high contrast, like the bush on the left remain at the same level of activation. Thereby contrast adaptation is performed, which allows the system to treat high- and low-contrast regions on equal grounds. All the operations that lead to this behavior are local and biologically plausible.

In fig. 31 we show how our system can perform illusory contour detection, with various synthetic stimuli. These examples help mention some of the discretization problems faced by our network, as e.g. in the top right image, where the fact that only 8 orientations are used results in a octagon-like illusory contour rather than a circle.

In the following pages the results of our system for real-world images are shown and its performance is compared to the Canny edge-detection scheme with approximately the same number of pixels and Perona-Malik anisotropic diffusion, with approximately the same level of smoothing. The middle row in each page compares the B.C.S. derived edges (left) with the Canny derived edges (right). We notice that our system does a good job at detecting smooth and long contours, and giving a piecewise smooth surface process. Specifically, in the first page one can notice the columns and the tree, in the second page the background pergola, the woman's hands, in the third and the fourth the backs of the dear and the kangaroo and so on. This is reflected in the F.C.S. outputs as well, where diffusion is inhibited along low contrast contours, if they are long and smooth enough to be detected by the B.C.S. When using Canny edge detection at a coarse scale, edge localization deteriorates significantly, while low contrast regions still do not give good results. At finer scales many edges are detected at textured regions by Canny edge detection, while our model avoids these due to its contrast normalization function. In favor of our model one should note at last that the outputs of the system have not been thinned 'explicitly' by a computer vision algorithm, but by the winner-take-all nature of the network; therefore, some edges are fatter than one-pixel wide, so the pixel count is somehow inflated for our model. We should note here that the benchmark results presented in section 5 validate the claim that the results of the BCS model are systematically better than those of Canny edge detection at a fine scale, but for a coarser scale this is no longer valid, at least using the performance measures used in [70].

By favoring smooth and long contours we bias our system outputs, as one can see for example in the image of the dear, where the edge starting from its leg is unified with the one due its head's shadow. In general, however, we believe it is a bias that pays off. Notice at last the problem of using a small number of orientations, where e.g. in the stones image the contours in the top left part of the image have an orientation of $\pi/8$, even though the input image contour orientations are rather smaller.

We should also mention that the term 'efficient' in the preceding text has been used in the sense of robustness to parameter changes, and of giving good results; as far as time-efficiency is concerned, our system is rather slow (it takes about 30 minutes on a Pentium-4 at 1.6 Ghz to run the system on 3 scales for a 321×481 image). The main bulk of the processing time is at the saliency computation part, where convolutions with large filters (e.g. 90×90) are required and where important savings could be achieved by recursively implementing the convolution operations [19] or adapting to our problem the efficient PDE algorithm described in [90], or by designing a deformable filterbank for the lobes [77].

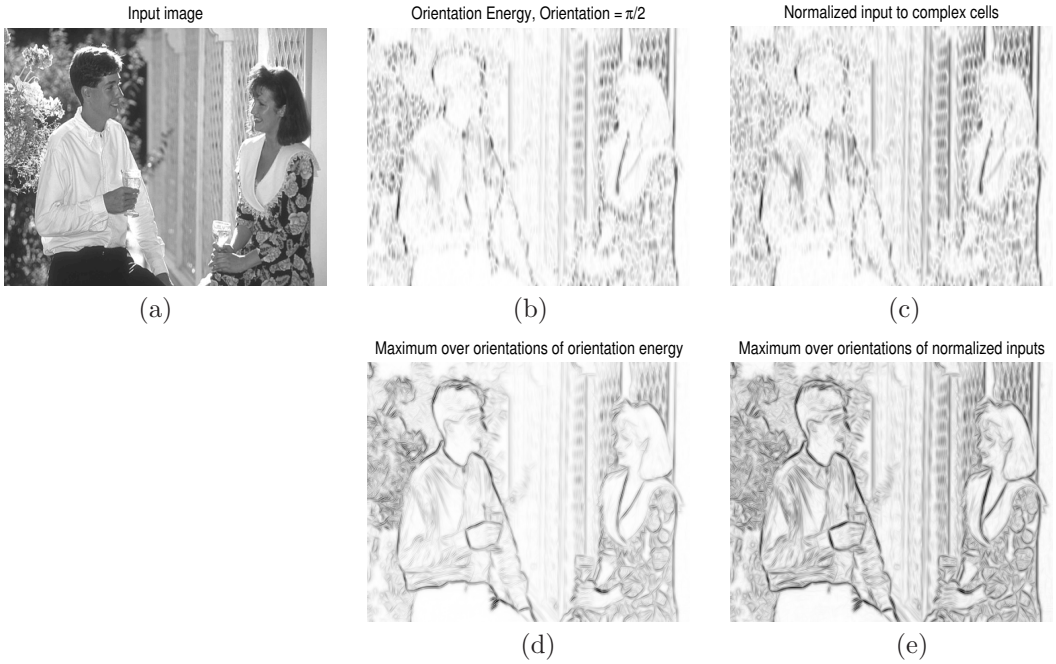


Figure 30: (a)Couple figure, taken from the database described in [71] (b) Orientation Energy and (c)Orientation Energy after Normalization for the vertical Direction (d)Maximum over directions of Orientation Energy (e) Maximum over directions of Orientation Energy after normalization

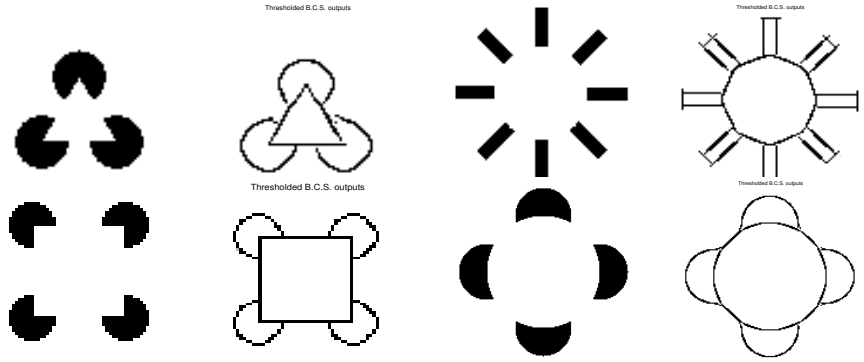
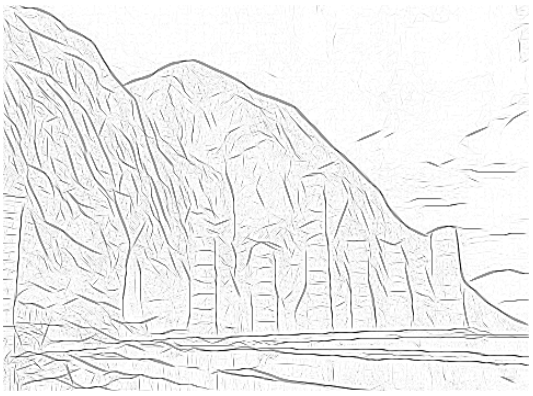


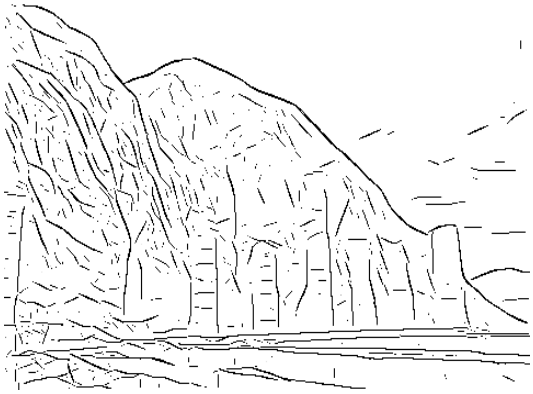
Figure 31: Illusory contour detection

Input image

B.C.S. outputs, finest scale



edge pixels: 9173



T_1 = 0.16, T_2 = 0.06, σ = 1, # edge pixels: 9464

F.C.S. results, finest scale

Perona–Malik diffusion λ =.1, 100 steps, k = 0.05

Input image

B.C.S. outputs, finest scale

edge pixels: 13445

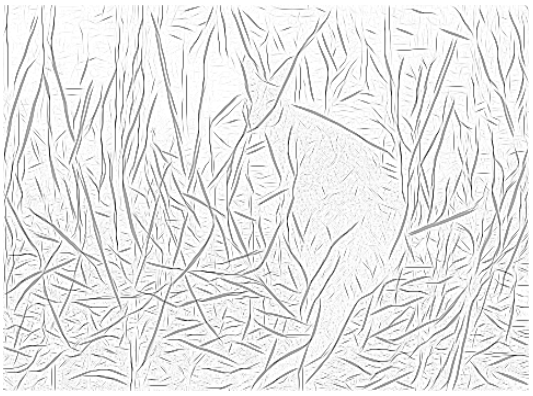
$T_1 = 0.16$, $T_2 = 0.06$, $\sigma = 1$, # edge pixels: 13858

F.C.S. results, finest scale

Perona–Malik diffusion $\lambda = .1$, 100 steps, k = 0.05

Input image

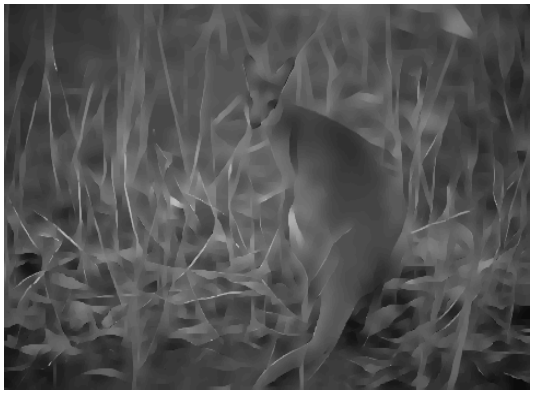
B.C.S. outputs, finest scale



edge pixels: 16385

T_1 = 0.35, T_2 = 0.14, σ = 1, # edge pixels: 16735

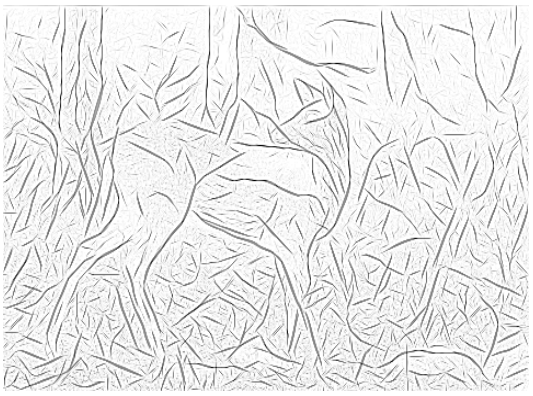
F.C.S. results, finest scale



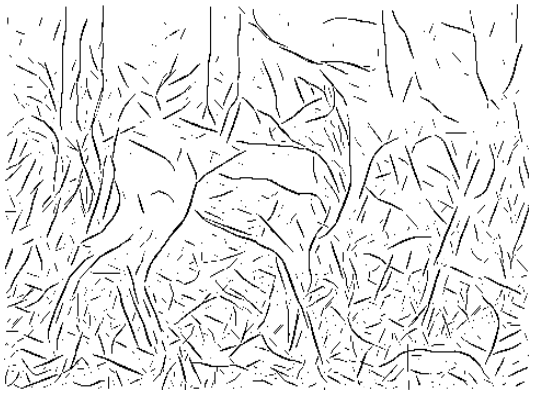
Perona–Malik diffusion λ =.1, 100 steps, k = 0.05

Input image

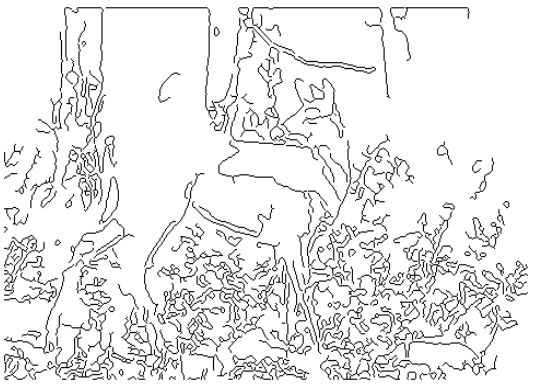
B.C.S. outputs, finest scale



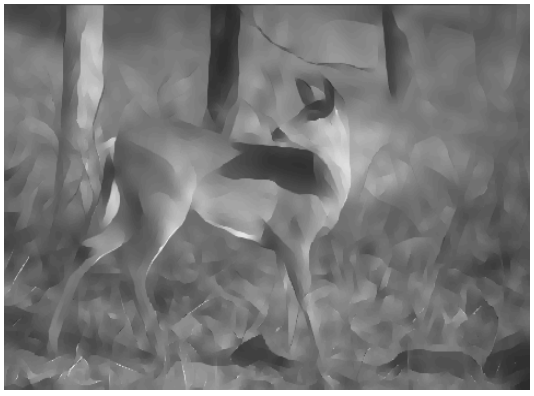
edge pixels: 12494



$T_1 = 0.33$, $T_2 = 0.13$, $\sigma = 1$, # edge pixels: 13310



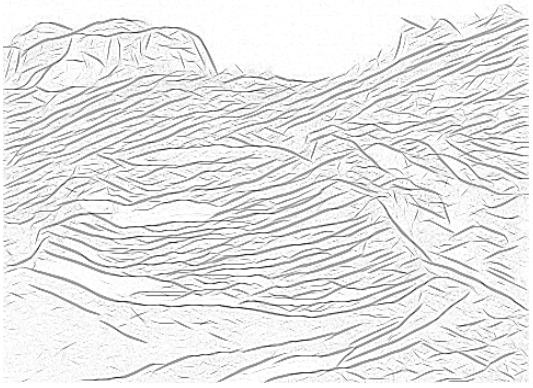
F.C.S. results, finest scale



Perona–Malik diffusion $\lambda = .1$, 100 steps, k = 0.05

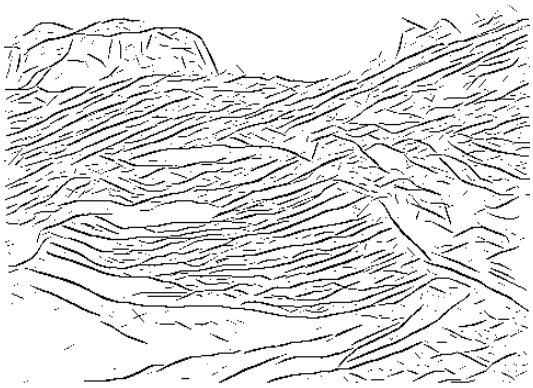
Input image

B.C.S. outputs, finest scale



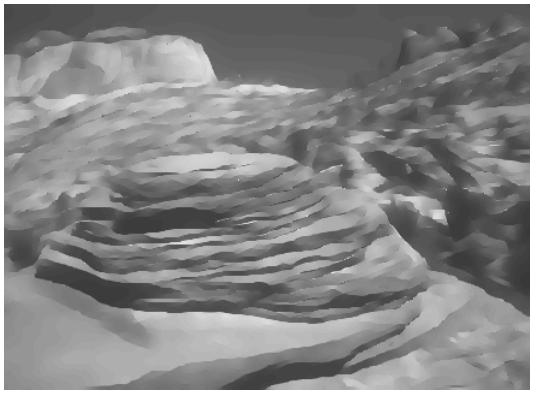
edge pixels: 18459

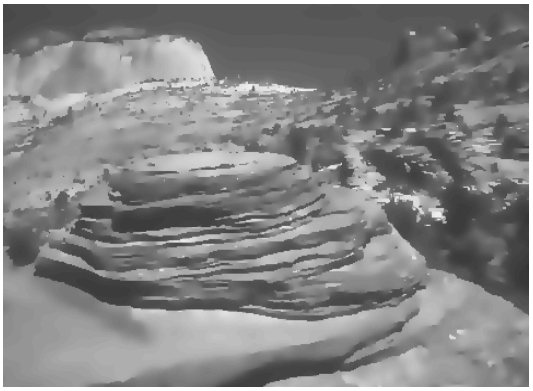
$T_1 = 0.25$, $T_2 = 0.10$, $\sigma = 1$, # edge pixels: 18868



F.C.S. results, finest scale

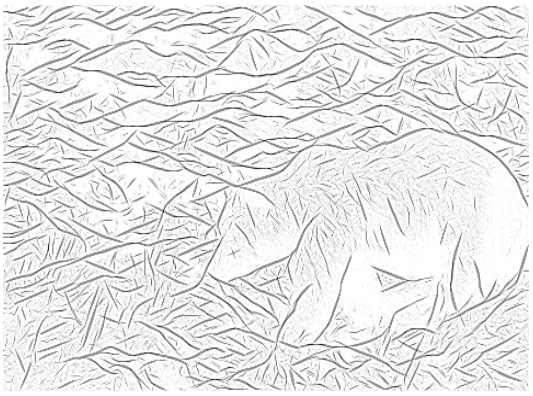
Perona–Malik diffusion $\lambda = .1$, 100 steps, k = 0.05



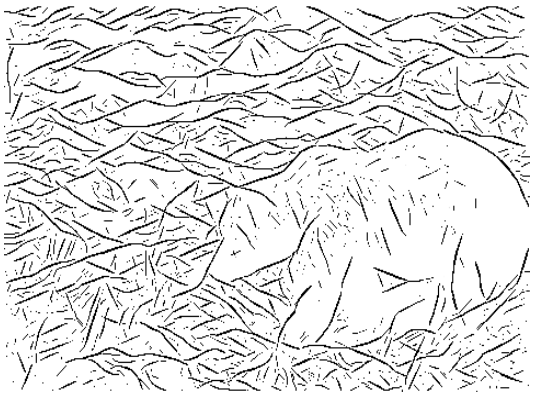


Input image

B.C.S. outputs, finest scale



edge pixels: 16977



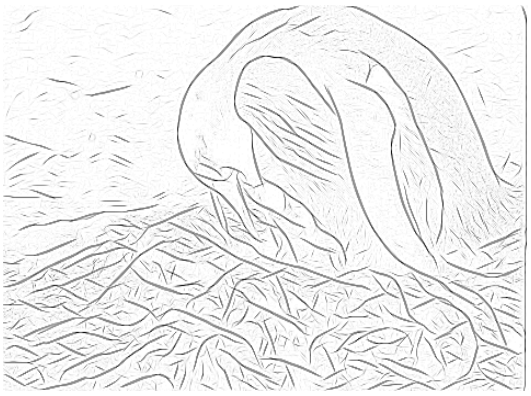
T_1 = 0.23, T_2 = 0.09, σ = 1, # edge pixels: 17090

F.C.S. results, finest scale

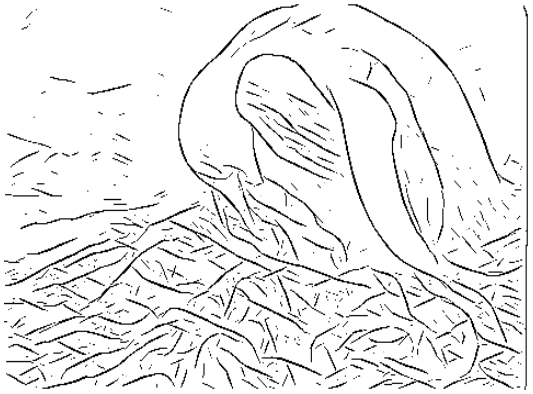
Perona–Malik diffusion λ =.1, 100 steps, k = 0.05

Input image

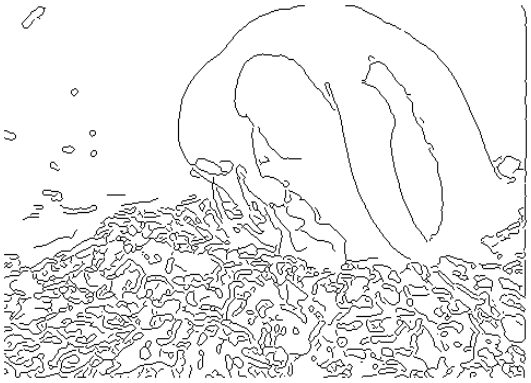
B.C.S. outputs, finest scale



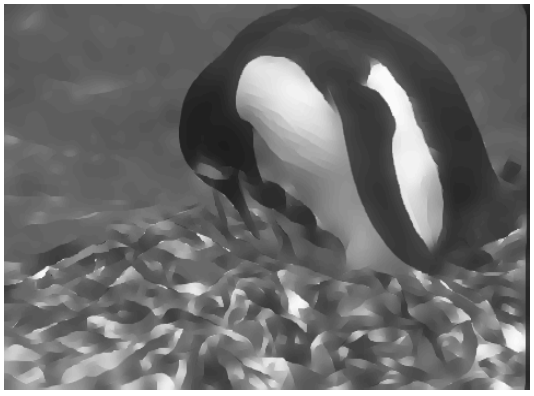
edge pixels: 11398



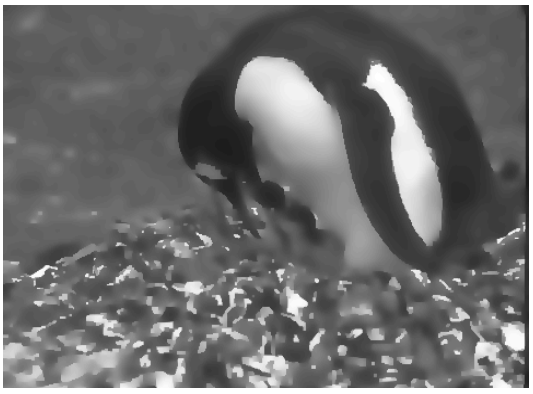
$T_1 = 0.20$, $T_2 = 0.08$, $\sigma = 1$, # edge pixels: 11671



F.C.S. results, finest scale



Perona–Malik diffusion $\lambda = .1$, 100 steps, k = 0.05



4.6 Interpreting the Model in Computer Vision Terms

Many of the functions used in this network, as well as the original B.C.S/F.C.S. model correspond to some commonly used techniques in computer vision; we start with the B.C.S.-related functions of the network and continue with the function of the network as a whole.

4.6.1 Nonmaximum Suppression

Nonmaximum suppression is commonly used for the purpose of deriving a clean and coherent edge map from fuzzy inputs like the outputs of filtering with spatially extended filters. A common technique for nonmaximum suppression is to take the local maximum of the filter responses in the gradient direction and set the others to zero, like in [11, 18, 78]; in [9, 24] a penalty term punishes spatial configurations of broad edges, while in [82] fitting the surface with a parabola and using the curvature and distance to the peak was proposed.

Keeping in accord with biological plausibility, our system in (48) implements nonmaximum suppression by an analog winner-take-all type network [93, 92] that suppresses the activations of less active neurons, allowing the stronger ones to stand out. Specifically, acting complementary to the excitatory term I, the inhibitory term $\sum_{\theta'} \sum_{k,l} W^{\theta,\theta'}_{[i,j]}[k,l] U^{\theta'}[k,l]$ punishes configurations where many neurons are active in the same neighborhood, leading to the suppression of weaker edge responses. The neuron which has the strongest excitatory term, I, than all its neighbors 'survives' and gives a well localized, both in space and orientation response.

4.6.2 Perceptual Grouping and the Elastica Prior on Curves

For the goal of enhancing perceptually salient contours various techniques have been developed in the computer vision community; a similar pattern of pixel interactions as that shown in fig. 28 is used in the voting technique of [42], while in [76] edge grouping is performed using a relaxation labelling process, where the compatibility pattern among the labels used in the relaxation process is highly reminiscent of the shape of the lobes used in S. Grossberg's model. In the probabilistic formulations used in [82, 89] edge saliency is propagated among processing nodes, while in [86] a criterion including the squared integral of the curvature is used for edge linking. The popular and simple hysteresis thresholding technique used in [11] can be seen as performing some sort of edge grouping, as well as the penalty term for line endings used in [9, 24].

In our model the contour grouping process is cooperating with the contour formation process, driving the latter to salient contours, while avoiding using initial hard decisions from an edge detector output. The shape of the lobes used to perform perceptual grouping, introduced by S.Grossberg in [36] is now popular among researchers in both computer and biological vision [20, 42, 68, 75, 76]; this is natural since their shape, which favors low curvature contours, that occur frequently in our visual environment, enforces a reasonable prior on the contour formation process.

In [89] a link between the shape of the lobes used for contour linking and the Elastica model of curves [73] was clarified which offers a better-founded mathematical interpretation of the bipole fields used. In [73] D. Mumford presented a prior model of curves in computer vision, which assumes that the curvature of a curve can be modeled as a white noise process. In particular, a curve Γ is supposed to be the path of a particle driven by a diffusion process in (x, y, θ) space, where x, y are the particle's position and θ is its orientation (the direction of the tangent to its trajectory at that point). The motion of the particle is described by the stochastic differential equation:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}x &= \cos(\theta) \\ \frac{\mathrm{d}}{\mathrm{d}t}y &= \sin(\theta) \\ \frac{\mathrm{d}}{\mathrm{d}t}\theta &= \kappa \sim n(0, \sigma^2; t) \end{aligned} \tag{52}$$

where κ, the curvature of the associated curve, is a white noise process. An exponential distribution with parameter λ is used as prior on the length of edges, so there is a time t after which the process gets trapped in a 'death' state; see also [6] for an elaborated model based on the above diffusion process.

Using this stochastic model of curves, the probability of a curve Γ can be related to an energy functional proposed by Euler and introduced in computer vision by B. Horn [55], namely the *Elastica* energy functional, that is given by

$$E(\Gamma) = \int_\Gamma (\alpha k^2 + \beta)\mathrm{d}s \tag{53}$$

Actually, it was shown in [73] that under the model (52)

$$P(\Gamma) \sim e^{-\int_\Gamma (\alpha k^2 + \beta \mathrm{d}s)}, \quad \beta = \lambda, \quad \alpha = \frac{1}{2\sigma^2} \tag{54}$$

Therefore one can see minimizers of the Elastica energy (53) as modes of the distribution (54) on particle paths Γ.

An interesting point is how we can apply this prior when we have some disconnected edge elements around a point (i,j), and we want to find the posterior probability $P(i, j)$ of a curve passing through it with tangent direction at that point equal to θ. This was dealt with in [89] and is reviewed below. A quantity used in [89] is the six dimensional tensor g on $(x, y, \theta, x', y', \theta')$ which helps express the probability that a curve will pass through a point (x, y, θ), given that its probability density function at time $t = 0$ is $p(x, y, \theta; 0)$. The probability of the particle path passing through x, y, θ at time t is given by

$$p(x, y, \theta; t) = \iiint_{x', y', \theta'} g(x, y, \theta, x', y', \theta') p(x', y', \theta'; 0) e^{-t/\tau} \tag{55}$$

and the probability that a curve will pass through a point (x, y) given $p(x, y, \theta; 0)$ is given by

$$\begin{aligned} p(x,y,\theta) &= \iiiint_{x',y',\theta',t} g(x,y,\theta,x',y',\theta')p(x',y',\theta';0)e^{-t/\tau} \\ &= \iiint_{x',y',\theta'} g'(x,y,\theta,x',y',\theta')p(x',y',\theta';0) \\ \text{where} \quad g'(x,y,\theta) &\equiv \int_t g(x,y,\theta,x',y',\theta')e^{-t/\tau} \end{aligned}$$

We could think of the points of x, y, θ space D^S lying on one side, S, of a point (x_o, y_o, θ_o) as starting points of curves and the points D^E on the other side, E, as ending points of curves (or source and sink points, as in [89]). The probability of a curve passing through (x_0, y_0, θ_0) given that it starts from a point $D_k^S = (x_k, y_k, \theta_k)$ and ends at a point $D_l^E = (x_l, y_l, \theta_l)$ is equal, due to the Markov nature of the random walk (52), to the product of the probabilities of the curves doing the 'two halves' of the road:

$$\begin{aligned} P(x_0,y_0,\theta_0|D_k^S,D_l^E) &= P(x_0,y_0,\theta_0|D_k^S)P(x_0,y_0,\theta_0|D_l^E) \\ &= \iiint_{x,y,\theta} g'(x_0,y_0,\theta_0,x,y,\theta)\delta(D_k^S;0) \iiint_{x,y,\theta} g'(x_0,y_0,\theta_0,x,y,\theta)\delta(D_l^E,0) \\ &= g'(x_0,y_0,\theta_0,x_k,y_k,\theta_k)\cdot g'(x_0,y_0,\theta_0,x_l,y_l,\theta_l) \end{aligned}$$

In the formula above the fact that the curve starts from a point D_k^S is expressed using as initial distribution a Dirac function, centered at D_k^S at time 0. In case there are multiple starting and ending points, a summation is required to calculate the probability of a curve arriving at x_0, y_0, θ_0 for each of the two sides; the summed terms are subsequently multiplied to calculate the probability of a curve passing through (x_0, y_0, θ_0) given its surroundings:

$$P(x_0,y_0,\theta_0) = \sum_k g'(x_0,y_0,\theta_0,x_k,y_k,\theta_k)\cdot \sum_l g'(x_0,y_0,\theta_0,x_l,y_l,\theta_l) \tag{56}$$

Using Monte Carlo simulation to approximate the tensor g' in [89] resulted in shapes that are similar to the bipole fields introduced by S. Grossberg (see fig. 28). Given that we used the product of these lobes in order to model the high-level feedback term, one could say that this is somehow related to the posterior probability of a contour passing through a point, conditioned on its surroundings, using the prior model of Elastica on curves. This would be mathematically sound if the summation in (56) was only over line-endings, as is done in [89]. What is done in the B.C.S. model and in the related models of [42, 76, 68] is that the whole surrounding of each point is used in order to calculate its posterior probability; however, the Markov assumption of the nature of the curves dictates that given the closest point of a curve potentially passing through a point, the rest of the curve should be indifferent.

This link with the Elastica model of curves helps intuitively understand on what grounds the specific shape of the lobes is justified, while it greatly simplifies designing the lobes,

since only two easily interpretable parameters are involved, namely λ and σ. It would be interesting to see what gain in performance can be achieved using line-ending detectors as the inputs to the Stage III' grouping process and how these could be combined with the outputs of Stage II' cells.

4.6.3 A Variational Perspective

The B.C.S./F.C.S. architecture clearly parallels the usage of line and surface processes by computer vision researchers, see e.g. [24, 93, 66, 9]; the work in these references is related in one way or another with the minimization of the Mumford-Shah functional [74]:

$$E(U,l) = \lambda \int_D (U-I)^2 + \mu \int_{R-l} |\nabla U|^2 + \nu |l| \tag{57}$$

In all cases, the idea is to introduce a line process l, make it as sharp and fine as possible and diffuse the image intensity (commonly called 'surface process') in the whole image except for areas where a line process element is active. What discriminates the original B.C.S./F.C.S. model from the work referenced above is that the F.C.S. is purely passive relative to the B.C.S. and therefore one cannot find a similar functional to (57); minimizing (57) in the discrete setting implies that the discontinuities of the surface process determine where a line process element can be 'afforded' and vice versa, while in the original B.C.S./F.C.S. the F.C.S. has no influence on the B.C.S. For the model proposed in this section we can find a Lyapunov function including both the surface and the line process neurons which is being decreased as the system evolves.

The existence of Lyapunov functions for recurrent networks [14, 53] has been exploited previously [62, 66, 91, 93] to devise neural networks that can solve variational problems in computer vision; this link as well as the interpretation of the recurrent scheme used in probabilistic terms is revised in Appendix A. Our goal is in the reverse direction, that is, to see how the recurrent variation of the B.C.S./F.C.S. model we proposed in the previous section can be interpreted in variational terms, and how we can explain its behavior using the derived energy functional. Even though based on [14] one can write down a Lyapunov function for the recurrent network described in the previous section, the integrals become messy and do not help intuition; we shall therefore consider the simplified version of (48):

$$\frac{\mathrm{d}V^\theta}{\mathrm{d}t}[i,j] = -aV^\theta[i,j] + cI^\theta[i,j] - d\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi,\sigma}[k,l] \tag{58}$$

where instead of the synaptic interaction among neurons we use the common sum-of-inputs model. For simplicity of notation we drop the scale index, considering every scale separately and treat temporarily the excitatory input I as constant; a Lyapunov energy of the network is then (see Appendix B):

$$E = \sum_{i,j,\theta}[a \underbrace{\int_{1/2}^{U^\theta[i,j]} g^{-1}(u)\mathrm{d}u}_{\text{P(U): Penalty for values of U}} - cI^\theta[i,j]U^\theta[i,j]]$$

$$+ \quad d/2 \underbrace{\sum_{i,j,\theta} \left[U^{\theta}[i,j] \sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l] U^{\phi}[k,l] \right]}_{\text{C(U): cost of configurations of U}} \tag{59}$$

where one can show that

$$P(U) = [U \ln(U) + (1-U)\ln(1-U)]/\beta + 1/2U - 1/4$$

which consists of an entropy-like term $[U \ln(U)+(1-U)\ln(1-U)]$, punishing binary responses and a second term $1/2U$ that generally punishes high responses; the first term is due to using a sigmoid transfer function in (50) and the second is due to shifting it by $1/2$ to the right. The factor β is related to the slope of the sigmoid function used: if β is large, i.e. a rapidly increasing sigmoid is used, the penalty on binary responses is less important and vice versa. Further, the term $-IU$ lowers the cost of a high value of U, facilitating the emergence of an edge.

One can also show (Appendix B) that

$$C(U) = \sum_{i,j,\theta} [Z * U^{\theta}]^2 - b \sum_{i,j,\theta} [Z^{\theta} * U^{\theta}]^2 + 1/2 \sum_{i,j,\theta,\phi,\theta \neq \phi} \left[Z^{\phi,\theta} * U^{\theta} \right] \left[Z^{\theta,\phi} * U^{\phi} \right] \tag{60}$$

In the above relation, $Z, Z^{\theta}, Z^{\theta,\phi}$ are scaled by $\sqrt{1/2}$ copies of the kernels $G, G^{\theta}, G^{\theta,\phi}$ used to determine the connection weights among neurons in the previous section. Z is an isotropic Gaussian filter so that $[Z * U^{\theta}]^2$ can be thought of as a term that punishes in general broad edges, irrespective of whether the edge elements are collinear and consistent or simply scattered around. Z^{θ} is an elongated Gaussian with principal axis the preferred orientation of neurons, θ; this term is a negative potential -a 'reaction' term- that favors sharp and well aligned edge profiles. The third term accounts for orientational competition, punishing neurons responding to edges at different orientations that are active in the same neighborhoods. $C(U)$ thus consists of both suppressive terms, namely the penalties on $Z * U_{\theta}$ and $\left[Z^{\phi,\theta} * U^{\theta} \right] \cdot \left[Z^{\theta,\phi} * U^{\phi} \right]$ that lead to the suppression of local structures, and a 'reactive' term $-[Z^{\theta} * U^{\theta}]^2$ that acts in favor of the emergence of isolated edges. These terms act in a complementary way, resulting in a reaction-diffusion like behavior, so that crisp boundaries are favored contrary to fuzzy ones.

Putting all the pieces together requires using the interaction with the surface process, as well; if we use the diffusion equation (51) and assume T^{θ} constant, a Lyapunov function is given by:

$$E = \sum_{i,j,\theta} c_4 \underbrace{(1 - U^{\theta}(i,j)) |\nabla_{\theta \perp} S|^2}_{\text{Line - Surface interaction}} - U^{\theta} \underbrace{[c_1 E^{\theta} + c_2 T^{\theta} + c_3 U^{\theta}_{\sigma+1}]}_{External\ inputs} + \underbrace{a \int_{1/2}^{U} g^{-1}(u) \mathrm{d}u + d/2 C(U)}_{\text{Cost for line process}} . \tag{61}$$

In the expression above the positive constants $c_1, ..c_4$ have absorbed the positive constant c of (58). Expression (61) has a lower bound, since the neuron outputs in the subtracted

terms cannot become larger than 1, so adding this lower bound to E makes it positive. By differentiating w.r.t S and U^θ we get the evolution equations (51),(58) respectively. This functional can be seen as a more complex version of that introduced in [9], where a simple penalty term was used to enforce nonmaximum suppression and contour continuity to the anisotropic diffusion-derived line process. One should also mention the similarity with [81], where the evolution of the line process was coupled with the evolution of the surface process, resulting in a conceptually similar model.

If one added a data fidelity term to the surface evolution process, one would get something similar to the line-process based algorithms for the minimization of the Mumford-Shah functional term. One should note, however, that this system does not simply boil down to a system that tries to find a minimum of the Mumford-Shah functional. Edge detection is more sophisticated in our system, since bottom up, region based, saliency based and multi - scale information is used to derive the final edge map. Based on the Mumford -Shah functional the only edge-detection-like information used should be the variation of the reconstructed surface process in the 4-pixel neighborhood of each location. This is too local and ignores the architecture of the visual system, which has filters integrating evidence over large regions to facilitate bottom-up edge detection.

5 Learning the Model Parameters from Ground-Truth Data

For our first experiments connection weights among neurons have been determined using ad-hoc function expressions, that were chosen based on intuition and evaluated by trial and error. It is however dubious whether intuition alone suffices for determining which combination of weights is better than another, and it is therefore desirable to be able to determine these weights using some learning scheme.

For this purpose we use ground truth segmentations from the Berkeley Segmentation benchmark [71] as ideal edge detector outputs and modify the network connections so as to minimize the difference between the network outputs and the ground truth data. A probabilistic approach is followed, where the outputs of the network are viewed as posterior probabilities of edges given the input image and our goal is expressed as minimizing the Kullback Leibler divergence [15] between the network outputs and the ground-truth edge probabilities. The network structure is not modified, since the same processing stages are used and the same neurons are interconnected. We thereby introduce hard-wired prior knowledge about the structure of the network, but allow for increased flexibility by allowing the connection weights to vary.

Related prior work, like the recent approaches to learning edge detection [64, 70] employs feedforward architectures, and the learning process focuses on doing some optimal cue combination. This distinguishes our model which uses a recurrent architecture, that is both biologically plausible and potentially more powerful. Further, the recurrent models

proposed in [21, 82] differ from ours both at the architectural and at the learning algorithm level.

Since our network is similar architecturally with the Boltzmann Machine (BM), we briefly present the basic notions and learning algorithms for the BM and how these apply to our case, subsequently we apply the Mean Field Approximation to our network and finally present the learning algorithm for each stage.

5.1 The Boltzmann Machine

The Boltzmann Machine (BM) [3, 49, 47, 58, 59, 88] is a probabilistic neural network of symmetrically connected binary units (fig. 32), which are separated in visible and hidden units, X and Y respectively. The separation of units into these groups is not fixed, but

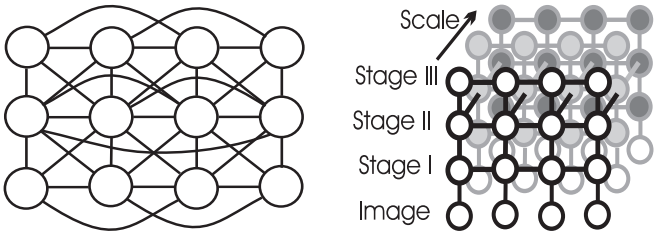


Figure 32: (a) Connection Pattern for a BM (b) Our system as a BM

depends on the available data.

An energy function of the form

$$E(X,Y) = -(\frac{1}{2}Y^T VY + \frac{1}{2}X^T WX + \frac{1}{2}Y^T JX) \tag{62}$$

is used for BMs, where V, W, J are symmetric matrixes that determine the intra/inter module interactions. Based on this energy function, a Boltzmann-Gibbs probability distribution of the network's state can be defined [3]:

$$P_{BM}(X,Y) = \frac{1}{Z}\exp(-E(X,Y)), \quad Z = \sum_{X,Y}\exp(-E(X,Y)), \tag{63}$$

where Z is a normalizing constant, called partition function in statistical physics. This equation can be rewritten in a factorized form as

$$P_{BM}(X,Y) = \prod_{i,j\in\mathcal{E}_{(X,Y)}}\phi(X_i,Y_j)\prod_{(i,j)\in\mathcal{E}_{(X,X)}}\psi(X_i,X_j)\prod_{(i,j)\in\mathcal{E}_{(Y,Y)}}\xi(Y_i,Y_j) \tag{64}$$

[3] for simplicity we assume the system temperature equals 1

where $\mathcal{E}_{\{(X,Y),(X,X),(Y,Y)\}}$ denote the pairs of interacting units and ϕ, ψ, ξ are determined by (62),(63). A notation related to $\mathcal{E}$ is $\mathcal{N}(j) = \{i : (i,j) \in \mathcal{E}\}$ which denotes the neighborhood of unit j.

In relation to our network we can consider first its basic function as an edge detection system and second its function during training. In the first case the observable nodes Y are the image intensities, and the 'hidden nodes' X are the edge neurons at the various stages and scales. In the second stage, the observables are both the image intensities and the states of some of the edge neurons, which are clumped to the manual edge detection results. In the following we shall provide a formal link based on which we view the previous section ODEs as implementing an inference algorithm in this setting. Based on this link we apply algorithms for learning the parameters of a BM to adapting the connection weights of our network.

Coming to the learning problem for the BM, this is posed as the minimization of the Kullback-Leibler (KL) divergence between the 'environmental' [49] distribution $P(Y)$ of the observable units and their distribution according to (63) $P_{BM}(Y) = \sum_X P_{BM}(Y,X)$:

$$\begin{aligned} KL(P(Y)|P_{BM}(Y)) &= \sum_Y P(Y) \log \frac{P(Y)}{P_{BM}(Y)} \\ &= \sum_Y P(Y) \log P(Y) - \sum_Y P(Y) \log P_{BM}(Y) \end{aligned}$$

In the above equation only the second term is of interest, since the first is independent of the BM and is equal to minus the entropy of Y. Learning can thus be seen as changing the network weights in (62) so as to construct a probability distribution function that closely follows the distribution of the observable network units.

The training algorithm for BMs [49] updates the weights according to $\Delta W_{i,j} \propto < v_i v_j >^+ - < v_i v_j >^-$ where $< v_i v_j >^+$ is the correlation of nodes v_i, v_j when the observable nodes are fixed to their observed values and $< v_i v_j >^-$ when the network is running free. Estimating these means demands performing stochastic simulation of the network which is computationally demanding and therefore impractical for a network as large as the one we use here. This calls for less accurate but more efficient inference algorithms [88, 59], which as we shall see is what the evolution equations described in the previous section perform.

5.2 Mean Field Approximation

The variational approach to approximate inference assumes the distributions to be inferred belong to a specific family $\mathcal{Q}$; inference is then posed as the search for the distribution $Q \in \mathcal{Q}$ which maximizes a specific criterion, $J(Q)$ (see [57] for a comprehensive tutorial). In our case we have a set of observable units Y and a set of hidden units X and we wish to estimate $P(X|Y)$; a suitable criterion to maximize for this purpose is:

$$J(Q) = \log P(Y) - KL(Q(X)|P(X|Y)) \tag{65}$$

The subtracted term is always positive and for $Q(X) = P(X|Y)$, $J(Q)$ equals $\log P(Y)$, which is the maximal value of $J(Q)$. Even though it is not guaranteed that $P(X|Y)$ can be expressed as a distribution in $\mathcal{Q}$, $J(Q)$ serves as a means to choosing a member of $\mathcal{Q}$ which is closest to $P(X|Y)$. Using the identities $P(Y)P(X|Y) = P(X,Y)$ and $\sum_X Q(X) = 1$, (65) can be written as:

$$\begin{aligned} J(Q) &= \sum_X Q(X) \log P(Y) - \sum_X Q(X) \log \frac{Q(X)}{P(X|Y)} \\ &= \sum_X Q(X) \log P(X,Y) - \sum_X Q(X) \log Q(X) \end{aligned} \tag{66}$$

In case $P(X,Y)$ has the form (64), the criterion simplifies to:

$$J(Q) = S + \sum_{i,j \in \mathcal{E}_{(X,X)}} \sum_{X_{i,j}} Q_{i,j}(X_{i,j}) \log \Psi_{i,j}(X_i, X_j) + \sum_{i,j \in \mathcal{E}_{(X,Y)}} \sum_{X_i} Q_i(X_i) \log \Phi_{i,j}(X_i, Y_j) + c \tag{67}$$

In the above equation, the interactions among observable units have been absorbed in the constant c and $S = -\sum_X Q(X) \log Q(X)$ equals the entropy of the variational distribution. In order to further simplify J we need to specify $\mathcal{Q}$, which is chosen according to the Mean Field Approximation (MFA) [57, 47]. According to the MFA the joint pdf $P(X|Y)$ over the set of random variables $X = \{X_1, \ldots, X_n\}$ can be well approximated in terms of pdfs $Q_i,\ i = 1, \ldots N$ defined over independent random variables:

$$P(X|Y) \simeq Q(X) = \prod_i Q_i(X_i) \tag{68}$$

In our case the Q_i are expressed by $Q_i(X_i = 0), Q_i(X_i = 1)$. $J(Q)$ becomes

$$J(Q) = S + \sum_{i,j \in \mathcal{E}_X} \sum_{X_i, X_j} Q_i(X_i) Q_j(X_j) \log \Psi_{i,j}(X_i, X_j) + \sum_i \sum_{X_i} Q_i(X_i) \log \Phi_i(X_i) \tag{69}$$

Above we use $\mathcal{E}_X$ instead of $\mathcal{E}_{(X,X)}$, we have dropped the c constant and have summarized all the observed-hidden node interactions for node i as:

$$\sum_{j \in \mathcal{N}_{(X,Y)}(i)} \sum_{X_i} Q_i(X_i) \log \Phi_{i,j}(X_i, Y_j) = \sum_{X_i} Q_i(X_i) \log \Phi_i(X_i) \tag{70}$$

If the values of $Q_j(X_j), j \in \mathcal{N}(i)$ are held fixed it is straightforward to estimate the values $Q_i(X_i = 0), Q_i(X_i = 1)$ that minimize $J(Q)$. Using $Q_i(1), \Phi_i(1)$ as shorthand notations for $Q_i(X_i = 1), \Phi_i(X_i = 1)$ and substituting $1 - Q_i(1)$ for $Q_i(X_i = 0)$ we can write the condition for an extremum of $J(Q)$ as:

$$\begin{aligned} \frac{\partial J(Q)}{\partial Q_i(1)} &= 0 \rightarrow \\ \log(Q_i(1)) - \log(1 - Q_i(1)) &= \log \Phi_i(1) - \log \Phi_i(0) + \\ &\quad \sum_{j \in \mathcal{N}(i)} \sum_{X_j} Q_j(X_j)(\log \Psi_{i,j}(1, X_j) - \log \Psi_{i,j}(0, X_j)) \end{aligned}$$

In case the potentials are of the form $\Psi_{i,j}(X_i, X_j) = \exp(C_{i,j}X_iX_j)$, $\Phi_i(X) = \exp(C_iX + b_i)$ the above equation is simplified as:

$$\log \frac{Q_i(1)}{1 - Q_i(1)} = \sum_{j \in \mathcal{N}(i)} Q_j(1)C_{i,j} + C_iQ_i(1) + b_i \rightarrow$$

$$Q_i(1) = \frac{1}{1 + \exp(-\sum_{j \in \mathcal{N}(i)} C_{i,j}Q_j(1) - C_iQ_i(1) - b_i)} \tag{71}$$

By iteratively updating the values of Q_i the criterion is steadily increased and the process converges to a local maximum of J.

Based on (71) we can relate the function of our network to that of a BM. Specifically, the ODEs for Stage II (58) lead to steady-states of the form (71) as can be seen by setting the time derivative to zero and using (50) to express the mean firing rate of the neurons. We can thus interpret these evolution equations as bringing the distribution $Q(X) = \prod_i Q_i(X_i)$ closer in the KL distance to $P(X|Y)$, where X are the units of Stage II and $Q_i(X_i = 1)$ is equal to $U_i = g(V_i)$.

For the first stage of our network it is not possible to derive such a straightforward interpretation of the ODEs in terms of mean field theory. The energy that is being minimized by the evolution equations cannot be written in the form used for the Boltzmann Machine, since there is a $\log(V)$ term in the final expression, using the Lyapunov Energy proposed by [14]. However we can still consider the outputs of that stage as the means of binary random variables, signalling the existence or absence of an edge. Even though this may not be derived using the MFA to Boltzmann machines, it is still used for all of the following whenever a mean of hidden variables is wanted.

Further, during training we only know the desired outputs of Stage II at the finest scale; we do not know the values that Stage I units or coarser-scale Stage II units should take. In principle one should iteratively apply the MFA to all stages and scales, going back and forth between updating the weights and the distributions of the hidden nodes and consider only the fine-scale Stage II neuron outputs as clamped. A heuristic we use instead of this, is to consider the desired distributions for each stage known in advance, which allows us to optimize the behavior of each module separately, along the coarse-to-fine flow of information in the network shown in fig. 33(a). This is also the updating scheme utilized during testing, so training the network is done consistently with testing it. Specifically, when updating the values of Stage II neurons we consider the processing modules of Stage I, and Stage II at coarser scales as fixed, whose nodes are treated as observables, contributing to the observation potentials $\Phi_i(X_i)$ for Stage II nodes. Still we update the values of Stage III and FCS nodes in parallel which can thus be seen as performing an MFA over clusters of nodes as shown in fig. 33(b), with inference for each cluster being accomplished by the original, small-scale MFA.

Concerning the desired outputs for Stage I neurons, we use Gaussian filtering to smooth the manually generated edge probability maps, where the Gaussian filter variance is half that of the Gabor filters used for feature extraction at the corresponding scale. The objective

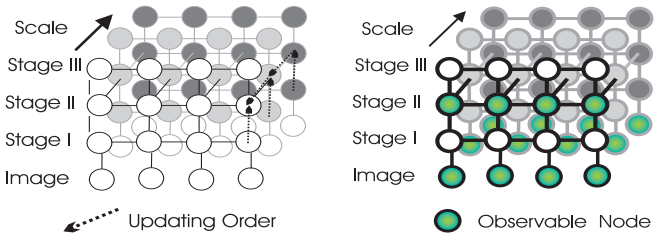


Figure 33: (a) MFA updating scheme: the MFA is used once for each stage, considering only the coarse-to-fine flow of information. (b) Observable nodes during training stage.

in Stage I is to elicit in a contrast-invariant manner areas of high edge probability, while avoiding responses on textured regions, so broad edge maps are left to the following stage for nonmaximum suppression. For Stage II the desired outputs are derived again by smoothing the manual probability maps, but at half the scale used for the corresponding map in Stage I, thereby enforcing an edge thinning behavior.

5.3 Estimating the Network Weights

Based on the interpretation of our network as a Boltzmann Machine we consider the following two cases: i) Learning phase, where the visible units are the pixel intensities and the outputs of Stage II, determined by the human segmentation-based edge probabilities -fig. 33(b), and ii) Testing phase, where the visible units are only the pixel intensities.

The training data made available in [71] are of the form $Y_{obs} = \{Y_1, \cdots, Y_n\}$ where each $Y_n = (I_n, E_n)$ consists of an image I_n and a corresponding manually determined binary edge map E_n. Training the network is interpreted as minimizing the KL divergence between the network distribution and the empirical distribution of observations:

$$KL(P(Y)|P_{BM}(Y)) = \sum_{Y} P(Y) \ln P(Y) - P(Y) \ln P_{BM}(Y)$$

Ignoring the entropy term $\sum_{Y \in Y_{obs}} P(Y) \ln P(Y)$ which is unaffected by learning, we focus on maximizing the empirical approximation to the second term:

$$\begin{aligned} L &= \sum_{n=1}^{N} P(I_n, E_n) \ln P_{BM}(I_n, E_n) \\ &= \sum_{n=1}^{N} P(I_n, E_n) \ln P_{BM}(E_n|I_n) P_{BM}(I_n) \end{aligned}$$

$$= \sum_{n=1}^{N} P(E_n|I_n)P(I_n)\ln P_{BM}(E_n|I_n) + \sum_{n=1}^{N} P(I_n,E_n)\ln P_{BM}(I_n)$$

where I_n, E_n is a pair of image-edge maps. Since the same image is segmented by more than one users, the environmental distribution $P(E_n|I_n)$ is approximated by considering all the edge maps provided for image I_n.

Optimizing the second summand would enable the network to correctly model the distribution of its inputs; as in [59, 88] this term is dropped as it is irrelevant to our case, since we are solely interested in the network outputs. Considering the training image-edge pairs are equally likely a priori, i.e. $P(I_n) = \frac{1}{N}$, we can write for the first summand

$$L' = \frac{1}{N}\sum_{n=1}^{N} P(E_n|I_n)\ln\left(P_{BM}(E_n|I_n)\right)$$

The quantities E, I used up to here correspond to whole images and not pixels, and therefore no node indexing has been used yet. However, since we are using MFA, we substitute the converged $Q(E_n) = \prod_i Q_i(E_{i,n})$ for $P_{BM}(E_n|I_n)$, which leads to the following simplification:

$$L' = \frac{1}{N}\sum_{n=1}^{N}\sum_{i}\sum_{b=\{0,1\}} P_{i,n}(b)\ln(Q_{i,n}(b)) \tag{72}$$

In the above relation $P_{i,n}(1)$ is estimated as the ratio of provided edge maps for image n for which node i belongs to a boundary. This time, all the terms in the criterion to be optimized are available: the expressions for $Q_{i,n}(b)$, as well as the pointwise defined probabilities $P_{i,n}(b)$.

For Stage I, we use the steady state values of (44) instead of $Q_i(1)$ in (72); the criterion to be maximized for a single image then becomes:

$$\begin{aligned}
L(C) &= \sum_i P_i(1)\log\frac{I_i}{a+I_i+\sum_j C_jU_j} + P_i(0)\log\frac{a+\sum_j C_jU_j}{a+I_i+\sum_j C_jU_j} \\
&= \sum_i P_i(0)\log\left(a+\sum_j C_jU_j\right) - (P_i(1)+P_i(0))\log\left(a+I_i+\sum_j C_jU_j\right) + P_i(1)\log(I_i) \\
&= \sum_i P_i(0)\log\left(a+\sum_j C_jU_j\right) - \log\left(a+I_i+\sum_j C_jU_j\right) + c \rightarrow \\
\frac{\partial L(C)}{\partial C_j} &= \sum_i P_i(0)\frac{U_j}{a+\sum_j C_jU_j} - \frac{U_j}{a+I_i+\sum_j C_jU_j}
\end{aligned} \tag{73}$$

which gives us a simple rule to update the values of C. For the sake of simplicity we use C_j instead of C_{i-j} inside the summations. In the previous expression and the following ones we drop the observation index n, considering a single image I at a time, which amounts to a stochastic gradient descent algorithm; we opted for this due to the large number of image edge-pairs that renders a batch learning algorithm impractical.

For Stage II we can derive a lower bound on the increase in the criterion L caused by a change in C and straightforwardly maximize this bound, along the lines of the Improved Iterative Scaling Algorithm [2, 8]. Using the steady-state value of (58) in place of $Q_i(1)$ we can write [4]:

$$L(C) = \sum_i P_i(1) \log \frac{1}{1+\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))} + P_i(0) \log \frac{\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))}{1+\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))} \tag{74}$$

$$\begin{aligned}
\Delta L(\Delta C) &= L(C+\Delta C) - L(C) \\
&= \sum_i P_i(1) \log \frac{1+\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))}{1+\exp(-\sum_{j\in\mathcal{N}(i)} (C_j+\Delta C_j) Q_j(1))} \\
&\quad + P_i(0) \log \frac{\exp(-\sum_{j\in\mathcal{N}(i)} (C_j+\Delta C_j) Q_j(1))}{\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))} \frac{1+\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))}{1+\exp(-\sum_{j\in\mathcal{N}(i)} (C_j+\Delta C_j) Q_j(1))} \\
&= -\sum_i P_i(0) \sum_j \Delta C_j Q_j(1) - (P_i(0)+P_i(1)) \log \frac{1+\exp(-\sum_{j\in\mathcal{N}(i)} (C_j+\Delta C_j) Q_j(1))}{1+\exp(-\sum_{j\in\mathcal{N}(i)} C_j Q_j(1))} \\
&= -\sum_i P_i(0) \sum_j \Delta C_j Q_j(1) - \log \left[Q_i(1) + Q_i(0) \exp(-\sum_{j\in\mathcal{N}(i)} \Delta C_j Q_j(1)) \right]
\end{aligned}$$

The first step uses the expression of $L(C)$ in (74), the second uses basic properties of the log function and the last step uses the expressions for $Q_i(1)$ $Q_i(0)$ that are used in (74). Along the steps described in [2, 8] we now derive a lower bound of J: The first lower bound is derived by applying the inequality $-\log(a) \geq 1-a$ to the last term of J, which gives

$$\Delta L \geq -\sum_i P_i(0) (\sum_j D C_j Q_j(1)) + 1 - Q_i(1) - Q_i(0) \exp(-\sum_j \Delta C_j Q_j(1))$$

The second bound is based on Jensen's inequality, according to which if $\sum p_i = 1$ we have $\exp(\sum p_i q_i) \leq \sum p_i \exp(q_i)$. We introduce $\Sigma Q_i = \sum_{j\in\mathcal{N}(i)} Q_j(1)$ and write:

$$\exp(\sum_j \Delta C_j Q_j(1)) = \exp(-\Sigma Q_i \sum_j \frac{Q_j(1)}{\Sigma Q_i} \Delta C_j) \leq \sum_j \frac{Q_j(1)}{\Sigma Q_i} \exp(-\Sigma Q_i \sum_j \Delta C_j)$$

[4] For simplicity we omit the observation potential terms appearing in (71); the following carry over directly to that case

We can now write

$$\Delta L \geq -\sum_i P_i(0)(\sum_j \Delta C_j Q_j(1)) + 1 - Q_i(1) - Q_i(0) \sum_j \frac{Q_j(1)}{\Sigma Q_i} \exp(-\Sigma Q_i \Delta C_j)$$

Thanks to the second lower bound, the partial derivatives of the rhs term, $\Delta L'$ w.r.t. each component of ΔC are decoupled:

$$\frac{\partial \Delta L'}{\partial \Delta C_j} = -\sum_i P_i(0) Q_j(1) + Q_i(0) Q_j(1) \exp(-\Sigma Q_i \Delta C_j) \tag{75}$$

In order to maximize the lower bound on ΔL we then perform gradient ascent:

$$\frac{\partial \Delta C_j}{\partial t} = \frac{\partial \Delta L'}{\partial \Delta C_j} \tag{76}$$

5.4 Learning Procedure

We have trained the detector using a subset (20 images) of the Berkeley segmentation training set, since the learning process proved to be time-consuming (~ 10 hours per picture). Using this small training set it was possible to learn interesting interconnection patterns and improve the detector's performance on a larger variety of images. Some bias we have hard-wired in the training procedure consists in:

- Clipping the connection weights to be greater or equal to zero; thereby the constraint of positive connection weights is satisfied.
- Averaging the connection weights, so that translation and rotation invariance of the interconnection pattern is guaranteed.
- Forcing the weights to be decreasing functions of the distance between the nodes; this was achieved by multiplying the increment for each weight by a factor proportional to $\exp(-d(i,j)/2\sigma^2)$, where $d(i,j)$ is the distance between nodes i and j and σ the network scale at which learning takes place.

In fig. 34 we compare the learned interconnection weights with those estimated using the formulae of section 4; it is obvious that intuition easily misled us to non-optimal connection patterns. For example when comparing the connections weights at Stage II we see that the conjectured absence of inhibition between neurons lying on the same line has been replaced by the contrary pattern; this can be credited to the feedback signal from Stage III, which leads to binary decisions and necessitates an inhibition signal to give rise to softer responses. We also observe a pattern similar to that set manually for Stage II (row 3 (b)) as the learned connection pattern for Stage I (row 2(b)).

In fig. 35 we demonstrate the effect of the training algorithm using one image that belongs to the training set and another that is in the test set. In general we observe that the network avoids taking sharp decisions and gives fewer false alarms at highly textured areas.

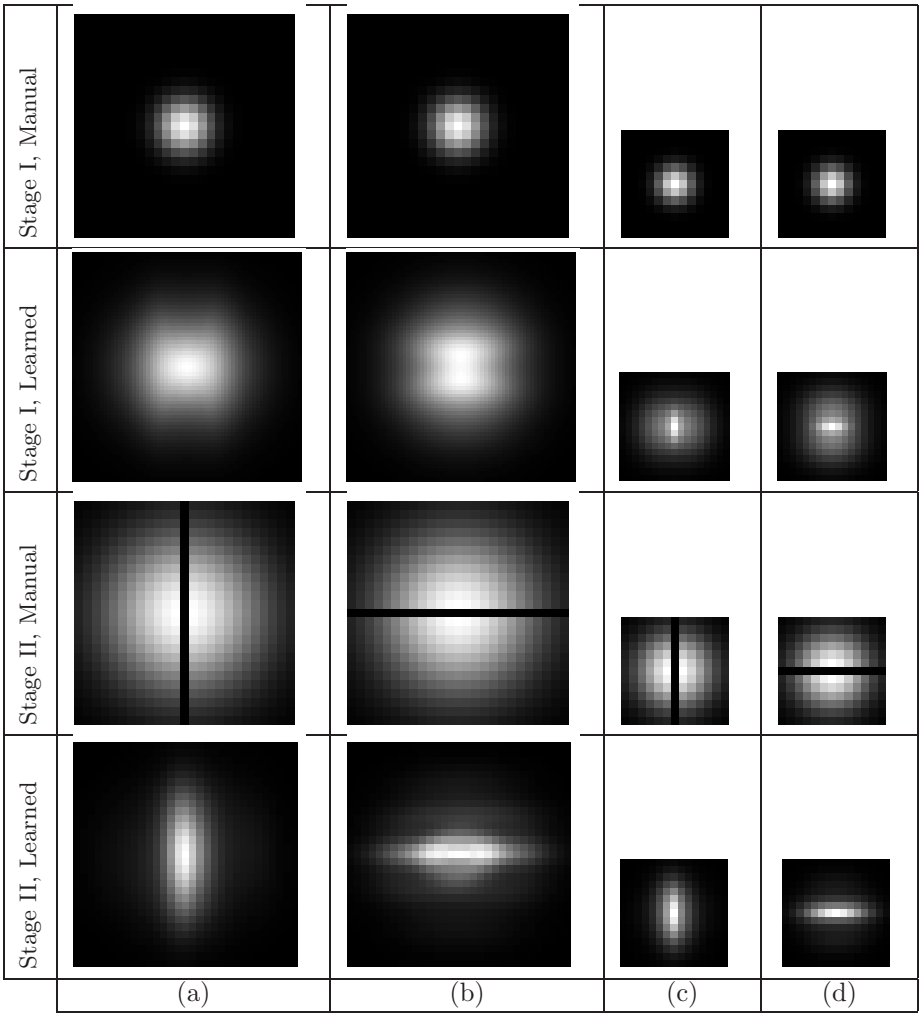
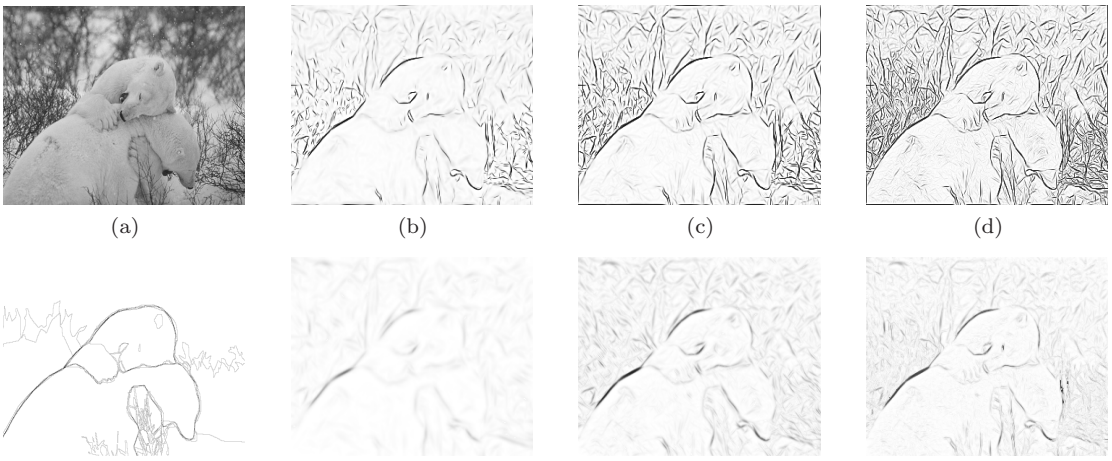


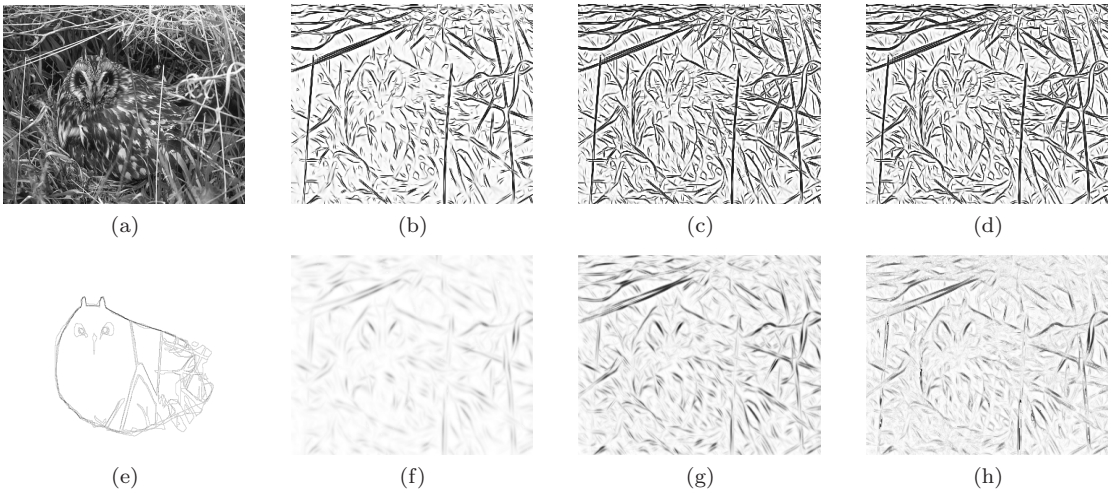
Figure 34: Difference between the manually set and the learned connection weights. In the even rows the manually set connection weights are shown, with a high gray value at a location indicating a strong connection between a neuron located at the center of the figure and a neuron at that location. In the odd rows the learned values for these weights are shown. The columns (a),(b) correspond to Horizontal-to-Horizontal and Vertical-to-Vertical connections at coarse scale while the columns (c)-(b) to the same connections at a fine scale.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 35: Learning edge detection- I: (a) Input Image, (b)-(d) Probability of Edge using manually set weights, at decreasing scales (e) Human Segmentations (f)-(h) Same as (b)-(d) using learned weights. This image belongs to the used training set.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 36: Learning edge detection-II: (a) Input Image, (b)-(d) Probability of Edge using manually set weights, at decreasing scales (e) Human Segmentations (f)-(h) Same as (b)-(d) using learned weights. This image is not included in the training set.

5.5 Benchmarking Results

By visual inspection one can qualitatively assess the performance of two different edge detection algorithms, however this is rarely a sufficient criterion to choose one algorithm over

another. An important step in the systematic evaluation of edge detection and segmentation algorithms has been the presentation of the Berkeley Segmentation benchmark [71, 1, 69] where natural images have been made available together with manually determined segmentations; data, code and details about benchmarking can be found in [1, 69].

The quantities used to describe a detector's performance are its Precision, P and its Recall, R. Recall equals the ratio of correctly detected edge pixels to the total number of edge pixels determined by the human segmentations and Precision equals the ratio of correctly detected edge pixels to the number of detected pixels by the detector. These two quantities are inversely related, and by modifying the detector's threshold they take a set of values, which is plotted in a Precision-Recall curve. Ideally a detector should have precision and recall equal to 1 (it should find all edge pixels and none of its detections would be false) so we can compare two detectors based on how close their Precision-Recall curves reach the ideal. When two Precision-Recall curves cannot be directly compared, for example when they intersect, a useful measure for summarizing the performance of the detector is its F-measure defined as:

$$F = \frac{1}{(\alpha)P^{-1} + (1-\alpha)R^{-1}} \tag{77}$$

α is a weighting factor which is typically set to 0.5. The curve's maximum F-measure can be used to summarize the detector's performance: larger values are more desirable and two detectors can be compared using their F measure.

In fig. 37 we show Precision-Recall curves for the edge-detectors implemented in our work, some well established edge detectors, as well as more recent edge detection algorithms from the work in [70]. Shown in the legend is the maximal F-measure of each detector, valued from zero to one, along with the coordinates of the location of the maximum.

The first conclusion drawn from fig. 37(a) is that learning the network weights improves the system's performance compared to that attained using manually set weights: none of the manually-determined BCS systems outperforms the learned system. From fig. 37(b) we realize that the claim made in the previous section, namely that the original system outperforms Canny edge detection is partially true, and holds only for the results obtained from Canny's method when using a small Gaussian function. For larger scales Canny's method outperforms the original system. In fig. 37(c) we observe however that when using the learned weights our system has a higher F-measure compared to the Canny, Oriented Energy and the Second Moment Matrix methods, for which the optimal scale is chosen. This may be attributed to the incorporation of multi-scale cues with region-based, saliency information and learning the connection weights in our system. Still, as shown in fig. 37(d) our detector does not perform quite as well as the Brightness or Brightness Texture Gradient methods of [70]. These methods utilize features and classifiers that practically proved to offer the best performance. For example, instead of the typically used output of a linear filter, a χ^2 test is used to assess the homogeneity of the observations on the two sides of a potential edge location. Even though this results in a significantly improved performance, it deviates from biological vision towards pattern recognition, leading us astray from our initial goal.

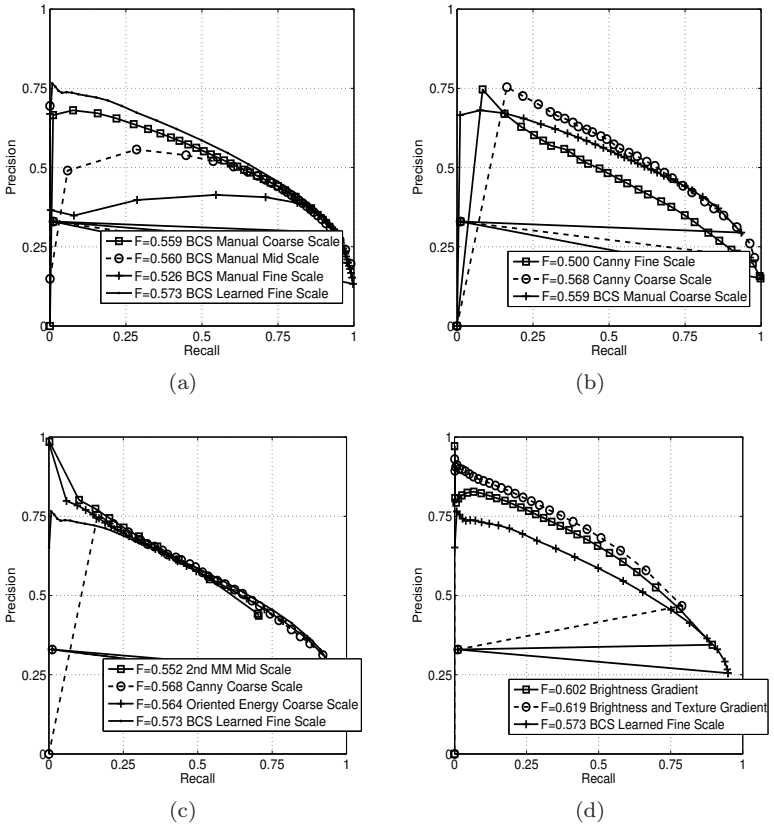


Figure 37: Comparative results for edge detection: (a) BCS system using learned weights vs. BCS systems with manually set weights at multiple scales. The learning algorithm results in a systematic improvement in performance. (b) Manually trained BCS results vs. Canny edge detection at two different scales. The improvement in performance observed in the previous section holds only for the fine-scale results. (c) Canny, Oriented Energy and Second Moment Matrix edge detection compared to the learned system results. Our system outperforms these algorithms, which constitute the state-of-the-art of the previous decades. (d) Brightness/Texture Gradient detection compared to our system. Our system's performance is inferior to these systems, which however lack biological plausibility.

6 Discussion

This report is a first step in the direction of integrating models from biological and computational vision; we have tried to understand and analyze a biological vision model from a

computer vision perspective, and examine whether some gain in performance compared to classical computer vision algorithms can be obtained using a biologically motivated model.

Apart from studying and simplifying the original BCS model the two major contributions of this work have been the analysis of the system in variational and statistical terms and the learning algorithm for the network weights. Contrary to relying on ad-hoc choices for the variational criterion terms involving line processes and filter responses, this approach minimizes the amount of human intervention in the construction of the network. The use of variational criteria for computer vision tasks is widespread, yet little work has been done on *learning* the terms in the criteria from ground truth data. Our approach can be seen as a step in this direction.

The research in this report could be extended in various directions; for example the experimental results in [70] demonstrate the significant role texture gradients and color play in detecting edges in natural images. A way to introduce texture-based and color based cues in our system could be the one described in [66], where a diffusion process on a Gabor-filterbank feature set is used to determine texture edges. This would introduce additional region-based terms in the evolution equations and again training data could be used to estimate how they should contribute to the evolution equations.

Our future research goals are focused on understanding at a deeper level the computations performed by the visual system, and establishing a firmer link between biological and computational vision systems. The FACADE theory of vision seems to be an interesting starting point for future research in this direction, since it has been proposed as a comprehensive model of biological vision; therefore, integrating the model presented here with the whole FACADE theory is one of our next goals. By critically and cautiously studying the FACADE model, we believe there is significant insight to be gained and probably some techniques to be developed that have not been yet proposed by computer vision researchers but are used by our visual system.

A Recurrent Neural Networks, Energy Minimization and Statistical Physics

A breakthrough in the analysis of recurrent networks was the introduction of a Lyapunov function in [52]. Soon after the introduction of the Lyapunov function for the analysis of recurrent neural networks in [54] the reverse procedure was studied, that is, designing neural networks for the problem of energy minimization. In case the energy function can be written in the form (17) the network then acts as if it were performing descent on the energy functional with a varying step (see also [91] for generalizations). Many problems in computer vision can be formulated in terms of minimizing energy functionals, so this point of view is of intrinsic interest to us. The use of Hopfield networks for variational problems and computer vision problems was explored in the 80's [54, 62, 91, 93]; in this Appendix we shall briefly review the most important aspects of this work, focusing on computer vision applications. For simplicity, we shall use one-dimensional data, so as to avoid tedious summations etc- the arguments carry over easily to the two dimensional case.

A.1 Recurrent networks and energy minimization techniques

First consider the regularization technique for ill-posed problems in computer vision, where the solution to the problem of the reconstruction of an image $\mathbf{X}$ from the observed data $\mathbf{F}$ is defined as

$$\mathbf{X}_{reg} = \text{argmin}_{\mathbf{X}} E_{reg}(\mathbf{X}), \quad E_{reg}(\mathbf{X}) = |\mathbf{X} - \mathbf{F}|^2 + \lambda |D\mathbf{X}|^2 \tag{78}$$

The first term in the energy functional E_{reg} of (78) is known as the *data fidelity* term, punishing the distance using the norm $|\cdot|$ between $\mathbf{X}$ and $\mathbf{F}$ and D is an operator detecting whatever property of X is undesirable, e.g. non-smoothness; λ determines the importance of our prior knowledge, which is encoded as a penalty on the outputs of the operator D. In the discrete setting, the operator D is expressed as a matrix $\mathbf{D}$ with elements $D_{i,j}$ which can be interpreted as determining the influence of the value of $\mathbf{X}$ at i to the value of $\mathbf{X}$ at j; if we use the Euclidean distance between the vectors $\mathbf{X}$ and $\mathbf{F}$ in equation (78) the energy functional E_{reg} can be written as

$$\begin{aligned} E_{reg}(\mathbf{X}) &= (\mathbf{X} - \mathbf{F})^T (\mathbf{X} - \mathbf{F}) + \lambda \mathbf{X}^T \mathbf{D}^T \mathbf{D} \mathbf{X} \\ &= \mathbf{X}^T (\lambda \mathbf{D}^T \mathbf{D} + \mathbf{Id}) \mathbf{X} - 2\mathbf{X}^T \mathbf{F} + \mathbf{F}^T \mathbf{F} \\ &= \frac{1}{2} \sum_{i,j=1}^{N} T_{i,j} X[i] X[j] + \sum_{i=1}^{N} I_i X[i] + c, \\ &\quad \text{where} \quad \mathbf{T} = 2(\lambda \mathbf{D}^T \mathbf{D} + \mathbf{Id}), \quad \mathbf{I} = -2\mathbf{F}, c = \mathbf{F}^T \mathbf{F} \end{aligned} \tag{79}$$

which can be identified (up to the constant c) with the Lyapunov function $E_{H.N.}$ of a Hopfield neural network (16) with linear g:

$$\begin{aligned} E_{H.N.}(\mathbf{X}) &= -\frac{1}{2}\sum_{i,j=1}^{N} w_{i,j}X[i]X[j] + \sum_{i=1}^{N}\frac{1}{R}\int_0^{g^{-1}(X[i])} g^{-1}(u)\mathrm{d}u + \sum_{i=1}^{N} I_i X[i] \\ &\overset{g(x)=x}{=} -\frac{1}{2}\sum_{i,j=1}^{N} w_{i,j}X[i]X[j] + \sum_{i=1}^{N}\frac{X[i]^2}{2R} + \sum_{i=1}^{N} I_i X[i] \\ &= \frac{1}{2}\sum_{i,j=1}^{N} T_{i,j}X[i]X[j] + \sum_{i=1}^{N} I_i X[i], \\ &= E_{reg} \quad \text{using} \quad w_{i,j} = -T_{i,j} + \delta_{i,j}\frac{1}{R}, \quad I_i = F_i \end{aligned} \tag{80}$$

Based on (79) T is a positive definite symmetric matrix, hence the W matrix involved in the expression of the corresponding $E_{H.N.}$ is symmetric as well by (80); hence we can interpret $E_{H.N.}$ as a Lyapunov function. Summing up, a continuous Hopfield network with linear units, connection weights $w_{i,j}$ and steady currents I_i as in (80) converges to a state that is the minimizer of the energy functional (78).

A more complicated case arises when the energy functional that is minimized is not convex; this is the case when a *line process* $\mathbf{L}$ is introduced, that acts complementary to the *surface process* $\mathbf{X}$, see e.g. [24, 10, 93]. When a line process element $L[i]$ between two surface process elements is active, their difference will not be punished (if their differences are encoded in $\mathbf{D}$). In order to avoid over-segmenting the image, line process elements should be inactivated, unless the image gradient is high enough to allow the introduction of a line process element. A common cost functional that expresses this is:

$$E(\mathbf{X},\mathbf{L}) = c_r\sum_{i=1}^{N}(X[i]-X[i+1])^2(1-L[i]) + c_f\sum_{i=1}^{N}(X[i]-F_i))^2 + c_l\sum_{i=1}^{N}L[i] \tag{81}$$

where c_r is the weight assigned to the regularization term punishing image variations, c_f is the weight assigned to the data fidelity term and c_l is for the line process term. In the two dimensional case the equations are similar, apart from the introduction of penalty functions for the spatial configurations of the line process elements [24, 62]. $L[i]$ in the above equation is a binary variable; thus, if $c_l L[i] > c_r(X[i]-X[i+1])^2$, i.e. if it is more economical to introduce a discontinuity in $\mathbf{X}$ at point i than to interpolate $\mathbf{X}$ there, then the $L[i]$ that minimizes E will be 1; otherwise it will be 0. The introduction of the binary line process results in a combinatorial optimization problem with many local minima, that has been dealt with using either simulated annealing techniques [24] or deterministic algorithms, like Graduated Nonconvexity [10] and Mean Field Annealing -MFA [23]. The technique that Hopfield and Tank proposed for the minimization of energy functions with neural networks [54] is closest in spirit to MFA and in [93] it was shown that MFA is actually implementable

by Hopfield networks, for appropriate parameter choices. What was suggested in [54] was to replace the binary valued variables $L[i]$ with continuous variables, and then let the system gradually evolve to states with binary states $L[i]$; that is, in equation (81), the line-process neuron should make initially softer 'decisions' (i.e. have a lower firing rate), allowing the system to avoid local minima and as time evolves take its decisions in a crisper way.

The desire to have non-binary line process can be expressed by the introduction of a term that punishes binary decisions in the energy function; a choice that fits the following analysis is the negative entropy-like term $c_g \sum_{i=1}^{N} L[i]\ln(L[i]) + (1-L[i])\ln(1-L[i])$, which results in:

$$\begin{aligned} E(\mathbf{X}, L) &= c_r \sum_{i=1}^{N} (X[i] - X[i+1])^2 (1 - L[i]) + c_f \sum_{i=1}^{N} (X[i] - f[i])^2 + c_l \sum_{i=1}^{N} L[i] \\ &+ c_g \sum_{i=1}^{N} L[i]\ln(L[i]) + (1-L[i])\ln(1-L[i]) \end{aligned} \tag{82}$$

c_g is a new constant, that determines the cost of having binary L_i values. By using initially a high value for c_g, binary decisions are avoided and gradually, by decreasing c_g a crisp, binary line process will emerge, which will be (hopefully) a global minimum of $E(\mathbf{X}, \mathbf{L})$; this is a *deterministic annealing* procedure that can give fast and high-quality results. The entropy-like term introduced is convenient, because, as we saw in 2.2.3, we have:

$$c_g \sum_{i=1}^{N} L(i)\ln(L[i]) + (1-L[i])\ln(1-L[i]) = \sum_{i=1}^{N} \int_{1/2}^{L[i]} g^{-1}(x)\mathrm{d}x \tag{83}$$

$$\text{where} \quad g(x) = \frac{1}{1+\exp(-(1/c_g)x)} \tag{84}$$

This means that the novel term introduced to avoid binary decisions is the same term that arose in the Lyapunov function of continuous Hopfield networks (17) as an effect of using a continuous g. This is reasonable intuitively, if we assume $L(i)$ are the neuron outputs of a Hopfield network with transfer function g; using a smooth g (i.e. a low $1/c_g$) is like demanding a strong input to get a binary output, so that the larger c_g, the harder it will be to get a binary response and vice versa. The deterministic annealing process of decreasing c_g is equivalent to letting the neuron take sharper responses, and as $c_g \to 0$ the neuron becomes a threshold unit giving binary responses. Based on the above, a computational scheme to minimize the energy function (82) is to use a Hopfield network having a Lyapunov function similar to the cost function (82), as described in [62]; the architecture proposed there includes:

(a) One network with linear neurons corresponding to the surface process $\mathbf{X}$, like the one that was used in (78). The part of the energy functional (82) that influences this network is

$$E_L(\mathbf{X}) = c_r \sum_{i=1}^{N} (X[i] - X[i+1])^2 (1 - L[i]) + c_f \sum_{i=1}^{N} (X[i] - F_i)^2 \tag{85}$$

The inner state (potential) of the neurons of this network is equal to their outputs, $\mathbf{X}$.
(b) One network with sigmoid neurons, corresponding to the line process $L(i)$, that is influenced by the energy term:

$$\begin{aligned} E_X(\mathbf{L}) = & c_r \sum_{i=1}^{N} (X[i] - X[i+1])^2 (1 - L[i]) + c_l \sum_{i=1}^{N} L[i] + \\ & + c_g \sum_{i=1}^{N} L[i] \ln(L[i]) + (1 - L[i]) \ln(1 - L[i]) \\ = & c_r \sum_{i=1}^{N} (X[i] - X[i+1])^2 (1 - L[i]) + c_l \sum_{i=1}^{N} L[i] + \sum_{i=1}^{N} \int_{1/2}^{L[i]} g^{-1}(u) \mathrm{d}u \end{aligned} \quad (86)$$

The inner state (potential) of the neurons of this network is V, and its outputs are given by $L[i] = g(V[i])$.

The minimization of (82) for a fixed c_g can be accomplished by updating the states of each network, considering the outputs of the other network fixed:

$$\frac{\mathrm{d}X[i]}{\mathrm{d}t} = -\frac{\mathrm{d}E_L(\mathbf{X})}{\mathrm{d}X[i]} = -2c_r(X[i] - X[i+1])(1 - L[i]) - 2c_f(X[i] - F_i) \quad (87)$$

$$\frac{\mathrm{d}V[i]}{\mathrm{d}t} = -\frac{\mathrm{d}E_X(\mathbf{L})}{\mathrm{d}L[i]} = -V[i] - c_l + c_r(X[i] - X[i+1])^2 \quad (88)$$

The above equations are readily interpretable as continuous Hopfield network evolution equations. The connection strengths between nodes $X[i], X[i+1]$ are determined by the existence of an active $L[i]$ between them: in case the $L[i]$ is active, as seen in (87) $X[i+1]$ has small influence on the value of $X[i]$ and vice versa. This is similar with the way the B.C.S. influences the activations of the F.C.S cells. $(X[i] - X[i+1])^2$ determines the formation of an edge: in the absence of evidence in favor of an edge, the term $-c_l$ in (88) will drive the neuron to a near-zero response; otherwise if $c_r(X[i] - X[i+1])^2 > c_l$ the neuron will become active, with a steady state inner potential $V[i] = c_r(X[i] - X[i+1])^2 - c_l$.

For the line process there are no connections between line neurons in (87). This is simply because we did not introduce any more complex cost term for the configuration of the line process than $\sum_i L[i]$; see e.g. [62] for a 2-D energy functional with interconnections between line-process neurons, as well as the following Appendix.

A.2 Neural Networks and Statistical Physics

A link can be made between neural networks and statistical physics techniques that have been used in computer vision for the minimization of energy functionals, allowing their interpretation in terms of a well-studied field of physics and computer vision. This link has been explored in depth (see [47] for an excellent reference) and has been discussed for computer vision applications in [93, 23].

Assume we have a system with state $\mathbf{X}$ at temperature β and associate with every state of it an energy $E(\mathbf{X})$. The Gibbs' distribution of the probability of each state is given by

$$P_\beta(\mathbf{X}) = \frac{1}{Z}\exp(-E(\mathbf{X})/\beta) \tag{89}$$

where Z is a normalizing quantity, called the partition function. A common approach to the minimization of an energy functional based on statistical physics consists in constructing a system whose energy is $E(\mathbf{X})$ and letting the system evolve stochastically according to some transition rules among the states of the system; these rules are designed in such a way that the system on average will be in state $\mathbf{X}$ with probability $P(\mathbf{X})$ and so that eventually the system comes into equilibrium, which means that mean values, variations, etc. of functions depending on its state become time invariant. Letting the system evolve in this way for a fixed temperature β, we can obtain a Monte Carlo approximation to the Minimum Mean Squared Error Estimate (MMSE) $< \mathbf{X} >_\beta$ of $\mathbf{X}$, by averaging the observed states of the system,

$$< \mathbf{X} >_\beta = \sum_{\mathbf{X}} \mathbf{X} P_\beta(\mathbf{X}) \simeq \sum_{\mathbf{X}_{observed}} \mathbf{X} \tag{90}$$

By slowly decreasing β, the system eventually gets 'trapped' around the (assumed unique) maximum $\mathbf{X}_{max}$ of the distribution $P_0(\mathbf{X})$, which results in the limit in

$$< \mathbf{X} >_0 = \sum_{\mathbf{X}} \mathbf{X} P_0(\mathbf{X}) = \sum_{\mathbf{X}} \mathbf{X}\delta_{\mathbf{X}_{max}}(\mathbf{X}) = \mathbf{X}_{max}$$

where $\delta_{\mathbf{X}_{max}}(\mathbf{X})$ is a Dirac distribution centered around $\mathbf{X}_{max}$. $\mathbf{X}_{max}$ is the state of maximal probability and minimal energy, so for $\beta \to 0$ the MMSE estimator becomes equal to the maximum likelihood estimator, which corresponds to a minimum of the energy function. $< \mathbf{X} >_\beta$ is not necessarily a minimum of E for $\beta \neq 0$, while it may correspond to states that are not attainable by the system: for a binary variable, $< \mathbf{X} >$ may be 1/2 if 0 and 1 are of equal probability.

An alternative to letting the system evolve stochastically at each β in order to calculate $< \mathbf{X} >_\beta$ is to use an approximation in equation (90), where instead of summing over all the possible values of the state variables, we use the mean values of all elements of the vector $\mathbf{X}$ apart from the one whose mean value is calculated:

$$< X[i] >_{mf} = \sum_{X[i]} X[i] P(X[i]| < \mathbf{X}' >) \quad \mathbf{X}' = \{X[1], ..X[i-1], X[i+1], ..X[n]\} \tag{91}$$

Solving this system of n equations results in a worse but fast and deterministic approximation to the system's state MMSE, yielding a *mean-field approximation*. By gradually increasing the temperature β involved in $P(\mathbf{X})$ we can lead the mean-field approximation to an estimate that may be close to the minimum of $E(\mathbf{X})$.

Using the above point of view, we can see the evolution equations of the previous section in a different light than that of performing descent on the energy functional (82): at steady state, the values of the processes L, X satisfy the following system of equations

$$\begin{aligned} X[i] &= c_r/(c_r+c_f)\cdot(1-L[i])\cdot X[i+1]+c_f/(c_f+c_r)F_i, \\ V[i] &= -c_l+c_r(X[i]-X[i+1])^2, \quad L[i]=g(V[i]) \quad i=1,\ldots,N-1 \end{aligned}$$

which we can interpret as relating the mean values of $L(i)$ and $X(i)$ with each other in an equilibrium configuration.

In that sense we could interpret the neuron outputs as the mean-field approximation to the MMSE of the random fields $\mathbf{L},\mathbf{X}$ of neuron outputs, if we consider the neuron outputs as inter-related random variables.

B Derivation of the Lyapunov Function

In this appendix we shall prove that the energy function (61)

$$E=\sum_{i,j,\theta} c_4(1-U^\theta[i,j])|\nabla_{\theta^\perp}S|^2-U^\theta[c_1E^\theta+c_2T^\theta+c_3U^{\theta,\sigma+1}]+a\int_{1/2}^{U} g^{-1}(u)\mathrm{d}u+\frac{d}{2}C(U)$$

where

$$C(U)=\sum_{i,j,\theta}\left[U^\theta[i,j]\sum_{\phi,k,l}W^{\theta,\phi}_{[i,j]}[k,l]U^\phi[k,l]\right]$$

is being constantly decreased by a system of neurons U,S following the evolution equations (58), (51).

We have defined $\nabla_{\theta^\perp}S=S'-S[i,j]$, where S' is the surface process element lying closest to $[i,j]$ along the line passing through $[i,j]$ with angle $\theta^\perp$. We therefore have that

$$\frac{\mathrm{d}E}{\mathrm{d}S[i,j]} = -2c_4\sum_\theta(1-U^\theta[i,j])\nabla_{\theta^\perp}S^\theta$$

while by (51) we have that:

$$\frac{\mathrm{d}S[i,j]}{\mathrm{d}t} = \sum_\theta(1-(U^\theta[i,j]))\nabla_{\theta^\perp}S^\theta$$

Since $c_4>0$, $\frac{\mathrm{d}E}{\mathrm{d}S}\frac{\mathrm{d}S}{\mathrm{d}t}<0$, and the evolution of the surface process leads to a decrease of E.

Concerning the line process, we have

$$\frac{\mathrm{d}E}{\mathrm{d}U^\theta[i,j]} = aV-[c_1E^\theta+c_2T^\theta+c_3U^{\theta,\sigma+1}+c_4|\nabla_{\theta^\perp}S|^2]+\frac{d}{2}\frac{\mathrm{d}C(U)}{\mathrm{d}U^\theta[i,j]}$$

Using the expression above for $C(U)$ we have

$$\begin{aligned}\frac{\mathrm{d}C(U)}{\mathrm{d}U^{\theta}[i,j]} &= \left[\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l] + U^{\phi}[k,l]W^{\phi,\theta}_{[k,l]}[i,j]\right] \\ &= 2\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]\end{aligned}$$

provided that $W^{\phi,\theta}_{[k,l]}[i,j] = W^{\theta,\phi}_{[i,j]}[k,l]$ which holds for the neuron interconnection pattern proposed in our model. In (58) we wrote $cI^{\theta} = [c_1E^{\theta} + c_2T^{\theta} + c_3U^{\theta,\sigma+1} + c_4|\nabla_{\theta^{\perp}}S|^2]$, so summing up we have:

$$\frac{\mathrm{d}E}{\mathrm{d}U^{\theta}[i,j]} = aV - cI^{\theta} + d\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]$$

while by (58)

$$\frac{\mathrm{d}V^{\theta}}{\mathrm{d}t}[i,j] = -aV^{\theta} + cI^{\theta} - d\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]$$

Therefore E can only decrease as with t, given that

$$U = g(V) \Rightarrow \frac{\mathrm{d}U}{\mathrm{d}t} = g'(V)\frac{\mathrm{d}V}{\mathrm{d}t}$$

and therefore

$$\frac{\mathrm{d}E}{\mathrm{d}U^{\theta}[i,j]}\frac{\mathrm{d}U^{\theta}[i,j]}{\mathrm{d}t} = -g'(V^{\theta}[i,j])\left[-aV^{\theta}[i,j] + cI^{\theta}[i,j] - d\sum_{\phi}\sum_{k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi,\sigma}[k,l]\right]^2 \leq 0,$$

since $g(.)$ is a nondecreasing function, it implies that $g'(V^{\theta}[i,j]) > 0$. We conclude that the evolution of the system of U, S neurons leads to a steady decrease of (61), so this is a Lyapunov function of the system.

We shall subsequently prove that $C(U)$ can be written as:

$$C(U) = \sum_{i,j,\theta}[Z * U^{\theta}]^2 - b\sum_{i,j}(Z^{\theta} * U^{\theta})^2 + 1/2\sum_{i,j,\theta,\phi,\theta\neq\phi}\left[Z^{\phi,\theta} * U^{\theta}\right]\left[Z^{\theta,\phi} * U^{\phi}\right]$$

where $Z, Z^{\theta}, Z^{\theta,\phi}$ are kernels which are scaled copies of the kernels determining the interconnection weights among neurons of the same and similar orientations. We shall initially show how the terms for neurons responding to the same orientation can be derived, and subsequently show the same for neurons of different orientations.

For the connection weights among neurons responding to the same orientation, θ we have used the expression

$$\begin{aligned} W^{\theta,\theta}_{[i,j]}[k,l] &= G_{[i,j]}[k,l] - bG^{\theta}_{[i,j]}[k,l] \\ &= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma^2}\right) \\ &\quad -\frac{b}{2\pi\sigma_1\sigma_2}\exp\left(-\frac{(\cos(\theta)(i-k)+\sin(\theta)(j-l))^2}{2\sigma_1^2}-\frac{(\sin(\theta)(i-k)-\cos(\theta)(j-l))^2}{2\sigma_2^2}\right) \end{aligned} \tag{92}$$

We note that $G_{[i,j]}[k,l] = G[i-j,k-l]$, $G^{\theta}_{[i,j]}[k,l] = G^{\theta}[i-k,j-l]$, where G, G^{θ} are simply Gaussian kernels, and specifically G^{θ} is elongated with principal axis along θ. We use the kernels Z, Z^{θ}, with the property that $Z * Z = G$,i.e. $\sum_{[q,r]} Z(q,r)Z(k-q,m-r) = G[k,m]$ and similarly $Z^{\theta} * Z^{\theta} = G^{\theta}$. Z, Z^{θ} are simply scaled by $\sqrt{1/2}$ copies of the original Gaussian kernels, G, G^{θ}. Given that G/G^{θ} and subsequently Z/Z^{θ} are even symmetric, one can write

$$\sum_{q,r} Z(q,r)Z(k-q,m-r) \stackrel{q'=-q,r'=-q}{=} \sum_{q',r'} Z(-q',-r')Z(k+q',m+r') = \sum_{q',r'} Z(q',r')Z(k+q',m+r')$$

so we can use the last expression interchangeably with the usual expression for convolution.

We introduce the quantity, $E^{\theta}[i,j]$:

$$\begin{aligned} E^{\theta}[i,j] &= \left(\sum_{k,l} Z^{\theta}_{[i,j]}[k,l]U^{\theta}[k,l]\right)^2 \\ &= \left[\sum_{k,l} Z^{\theta}_{[i,j]}[k,l]U^{\theta}[k,l]\right]\left[\sum_{m,n} Z^{\theta}_{[i,j]}[m,n]U^{\theta}[m,n]\right] \\ &= \sum_{k,l} U^{\theta}[k,l]\left\{Z^{\theta}_{[i,j]}[k,l]\left[\sum_{m,n} Z^{\theta}_{[i,j]}[m,n]U^{\theta}[m,n]\right]\right\} \end{aligned}$$

Summing over $[i,j]$ we have

$$\begin{aligned} \sum_{i,j} E^{\theta} &= \sum_{i,j}\sum_{k,l} U^{\theta}[k,l]\left\{Z^{\theta}_{[i,j]}[k,l]\left[\sum_{m,n} Z^{\theta}_{[i,j]}[m,n]U^{\theta}[m,n]\right]\right\} \\ &= \sum_{k,l} U^{\theta}[k,l]\left\{\sum_{i,j} Z^{\theta}_{[i,j]}[k,l]\left[\sum_{m,n} Z^{\theta}_{[i,j]}[m,n]U^{\theta}[m,n]\right]\right\} \\ &= \sum_{k,l} U^{\theta}[k,l]\left\{\sum_{i,j} Z^{\theta}(i-k,j-l)\left[\sum_{m,n} Z^{\theta}(i-m,j-n)U^{\theta}[m,n]\right]\right\} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k,l} U^\theta[k,l] \left\{ \sum_{i,j} \sum_{m,n} \left[Z^\theta(i-k, j-l) Z^\theta(i-m, j-n) \right] U^\theta[m,n] \right\} \\
&= \sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} \left[\sum_{i,j} Z^\theta(i-k, j-l) Z^\theta(i-m, j-n) \right] U^\theta[m,n] \right\} \\
\overset{i-k=q, j-l=r}{=} &\sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} \left[\sum_{q,r} Z^\theta(q,r) Z^\theta(q+k-m, r+l-n) \right] U^\theta[m,n] \right\} \\
&= \sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} G^\theta(k-m, l-n) U^\theta[m,n] \right\}
\end{aligned}
$$

We have thus shown that

$$\sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} (G^\theta_{[k,l]}[m,n]) U^\theta[m,n] \right\} = \sum_{k,l} \left(Z^\theta * U^\theta \right)^2$$

The proof for the relation

$$\sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} (G_{[k,l]}[m,n]) U^\theta[m,n] \right\} = \sum_{k,l} \left(Z * U^\theta \right)^2$$

is identical, so it is omitted. Using these two relations one can write

$$\sum_{k,l} U^\theta[k,l] \left\{ \sum_{m,n} W^{\theta,\theta}_{[k,l]}[m,n] U^\theta[m,n] \right\} = \sum_{k,l} \left(Z * U^\theta \right)^2 - b \sum_{k,l} \left(Z^\theta * U^\theta \right)^2$$

which is the part of the energy term for neurons having the same orientation, θ.

Concerning the interactions among neurons responding to edges at different orientations, we had chosen connection weights equal to

$$W^{\theta,\phi}_{[i,j]}[k,l] = a(\theta,\phi) G_{[i,j]}[k,l]$$

where $a(\theta,\phi)$ determines how strong an influence a neuron with orientation θ has on a neuron with orientation ϕ; this term makes no difference in the proof, as long as it is symmetric and non negative. The proof is the same with the above, using the kernel

$$Z^{\theta,\phi}_{[i,j]}[k,l] = \sqrt{a(\theta,\phi)} G(x/\sqrt{2}, y/\sqrt{2})$$

for which $Z^{\theta,\phi} * Z^{\theta,\phi} = W^{\theta,\phi}$.

Defining $E^{\theta,\phi}(i,j)$ as

$$\begin{aligned} E^{\theta,\phi}(i,j) &= \left[\sum_{k,l} Z^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]\right]\left[\sum_{k,l} Z^{\phi,\theta}_{[i,j]}[k,l]U^{\theta}[k,l]\right] \\ &= \sum_{k,l} U^{\phi}[k,l]\left\{Z^{\theta,\phi}_{[i,j]}[k,l]\left[\sum_{m,n} Z^{\phi,\theta}_{[i,j]}[m,n]U^{\theta}[m,n]\right]\right\} \end{aligned}$$

and summing over i,j one gets (we skip identical steps)

$$\begin{aligned} \sum_{i,j} E^{\theta,\phi} &= \sum_{k,l} U^{\phi}[k,l]\left\{\sum_{i,j}\sum_{m,n}\left[Z^{\theta,\phi}(i-k,j-l)Z^{\phi,\theta}(i-m,j-n)\right]U^{\theta}[m,n]\right\} \quad (93) \\ &= \sum_{k,l} U^{\phi}[k,l]\left\{\sum_{m,n}\left[\sum_{i,j} Z^{\phi,\theta}(i-k,j-l)Z^{\phi,\theta}(i-m,j-n)\right]U^{\theta}[m,n]\right\} \quad (94) \\ &\overset{i-k=q,j-l=r}{=} \sum_{k,l} U^{\phi}[k,l]\left\{\sum_{m,n}\left[\sum_{q,r} Z^{\phi,\theta}(q,r)Z^{\phi,\theta}(q+k-m,r+l-n)\right]U^{\theta}[m,n]\right\} \\ &= \sum_{k,l} U^{\phi}[k,l]\left\{\sum_{m,n} W^{\phi,\theta}_{[k,l]}[m,n]U^{\theta}[m,n]\right\} \end{aligned}$$

The relation $Z^{\phi,\theta}(i-k,j-l) = Z^{\theta,\phi}(i-k,j-l)$ used for the transition from (93) to (94) above is not an obvious one, and holds because we have used radially symmetric Gaussian kernels; in case elongated Gaussian kernels were used, with principal orientations along θ, it can be seen that it no longer holds. Summing up,

$$\sum_{k,l} U^{\phi}[k,l]\left\{\sum_{[m,n]} W^{\phi,\theta}_{[k,l]}[m,n]U^{\theta}[m,n]\right\} = \sum_{i,j}\left[\sum_{[k,l]} Z^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]\right]\left[\sum_{[k,l]} Z^{\phi,\theta}_{[i,j]}[k,l]U^{\theta}[k,l]\right]$$

The previous results show that we can write:

$$C(U) = \sum_{i,j,\theta}\left[Z * U^{\theta}\right]^2 - b\sum_{i,j,\theta}\left[Z^{\theta} * U^{\theta}\right]^2 + 1/2\sum_{i,j,\theta,\phi,\theta\neq\phi}\left[Z^{\phi,\theta} * U^{\theta}\right]\left[Z^{\theta,\phi} * U^{\phi}\right]$$

C Implementation Details

C.1 Original B.C.S./F.C.S. system

Our experiments with the original B.C.S./F.C.S. were mostly based on the paper [39] since it is there where most details are given about the exact structure of the B.C.S. However, we

faced quite a lot of problems when trying to replicate some of the experiments presented in other papers, so we finally chose to use a model as simple and at the same time as efficient as possible.

Specifically, the B.C.S. used for our experiments consisted of 3 processing scales, each of which had 6 processing stages, as described in section 3. In the remainder of this section, we shall use a 'scale index' s_i taking the values $1, 2, 3$, corresponding to the different processing scales used. In all of the equations, we used $b = d = 1$, so those quantities will not be mentioned further. For the B.C.S. system stages, we found the following parameter values gave a reasonable performance:

- On-off cells parameter values:
  $a = 0.2, \quad \sigma_{1,i} = 0.2, \quad \sigma_{2,i} = s_i, \quad i = 1 \dots 3.$
- Simple cells: 8 different orientations have been used, across which changes in On-Off cell activations are detected. Spatially offset elongated Gaussian filters were used, where the parameter values used for these are (as in eqn.(28)):
  $\sigma_{x,i} = 0.5 s_i, \quad \sigma_{y,i} = 1.5 s_i, \quad c_i = \sigma_{x,i}/2, \quad i = 1 \dots 3.$
- Stage IV Cells: The parameters used are:
  $\sigma_{1,i} = 0.2, \quad \sigma_{2,i} = s_i, \quad \alpha = 1, \quad i = 1 \dots 3.$
- Stage V Cells: The connection strengths among neurons with orientations θ, ϕ were set equal to $i^{\phi,\theta} = |\sin(\phi - \theta)|$ and $e^{\phi,\theta} = \delta(\phi, \theta)$, while α was set to 1.
- Stage VI Cells: the shapes of the lobes at the smallest scale has been derived by setting $\beta = 0.5$, $\mu = 15$, $\gamma = 10$, while at larger scales we used the same formula, but divided the coordinates of the neurons by s_i.

For the F.C.S. we used a time step $\mathrm{d}t = 0.1$ to solve system of O.D.E.s while the parameters $\delta = 1, \epsilon = 100, a = c = 0.001$ gave reasonable results.

C.2 Our model

Concerning our model, at Stage I', even and odd symmetric filters like those described in [66] were used. 3 scales and 8 different orientations were used. The orientations are determined by the period ω of the sinusoid used in the Gabor filter which was set to $\pi, \pi/2, \pi/4$ for the three scales. Even though one can compute analytically the value of a in (24) so that the D.C. component of the even symmetric filter is 0, calculating it numerically was necessary due to discretization problems. Normalization of the filter outputs was not possible by simply requiring that the norm of the filter equals one; discretization problems again required applying the filters to ideal stimuli (long step edges with the filters preferred orientations) and subsequently normalizing the filter responses based on these outputs.

A further subtlety concerns the fact that the line process neurons are located in between surface process neurons; careful treatment of the relations between odd and even symmetric filter outputs is therefore necessary. MATLAB code necessary to achieve the desired behavior

for various stimuli (bars, step edges etc.) is available on demand. For the sake of biological plausibility, we separated positive and negative simple cell responses, and performed the normalization process in parallel for each 'half', with the other half contributing to the shunting term; this was not denoted in the equations, for notational clarity.

For the connection strengths in Stage III', we used Gaussian filters with spreads equal to the larger spreads of the Gabor filters used at Stage II', while the weights c_1, c_2, c_3, c_4 have been determined empirically so as to strike a balance between the desire of being able to produce an edge in the absence of bottom-up input (E) and the desire of staying close to available bottom-up input. c_g, which was used to define the sigmoidal function, was set to the high value 7 to guarantee the all-or-nothing behavior of the network.

The lobes used for Stage III' were computed using Monte Carlo estimation of g' with 10^5 samples from the Elastica distribution on curves, with parameter values set to $\sigma = 0.2$ and $\lambda = 10, 20, 40$, for the three scales. Again, normalization of the outputs of this stage required using ideal stimuli for different orientations.

The F.C.S. diffusion process used the line process neurons detecting edges at (close to) vertical orientations θ to block the diffusion among neighboring horizontal cells. Their outputs were added, after multiplying them with $|\sin(\theta)|$. This way vertical cells contribute maximally to blocking the diffusion and horizontal ones do not. The same idea was used for the diffusion in the vertical direction.

References

[1] *The Berkeley Segmentation Dataset and Benchmark*. `http://www.cs.berkeley.edu/projects/vision/grouping/segbench/`.

[2] A. Berger and S. D. Pietra and V. D. Pietra, *A Maximum Entropy Approach to Natural Language Processing*, Computational Linguistics, 22 (1996), pp. 39–71.

[3] D. Ackely, G. Hinton, and T. Sejnowski, *A Learning Algorithm for Boltzmann Machines*, Cognitive Science, 9 (1985), pp. 147–169.

[4] E. Adelson and J. Bergen, *Spatiotemporal Energy Models for the Perception of Motion*, Journal of the Optical Society of America, 2 (1985), pp. 284–299.

[5] M. Arbib, ed., *The Handbook of Brain Theory and Neural Networks*, MIT Press, 1995.

[6] J. August and S. Zucker, *Generative Model of Curve Images with a Completely-Characterized Non-Gaussian Joint Distribution*, in Workshop on Statistical and Computational Theories of Vision at ICCV, 2001.

[7] R. Ben-Yishay, R. Bar-Or, and H. Sompolinsky, *Theory of Orientation Tuning in Visual Cortex*, Proc. Nat.l Academy of Sciences of USA, 92 (1995).

[8] A. Berger, *The Improved Iterative Scaling Algorithm: A Gentle Introduction*, 1997. `http://citeseer.ist.psu.edu/berger97improved.html`.

[9] M. Black, G. Sapiro, D. Marrimont, and D. Heeger, *Robust Anisotropic Diffusion*, IEEE Transactions on Image Processing, 7 (1998), pp. 421–432.

[10] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, 1987.

[11] J. Canny, *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986), pp. 679–698.

[12] M. Carandini and D. Heeger, *Summation and Division by Neurons in Visual Cortex*, Science, 264 (1994), pp. 1333–1336.

[13] M. Carandini, D. Heeger, and J. A. Movshon, *Linearity and Normalization of Simple Cells of the Macaque Primary Visual Cortex*, Journal of Neuroscience, 17 (1997), pp. 8621–8644.

[14] M. Cohen and S. Grossberg, *Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks*, IEEE Transactions on Systems, Man, and Cybernetics, 13 (1983), pp. 815–826.

[15] Cover and Thomas, *Elements of Information Theory*, Wiley, 1993.

[16] J. Daugman, *Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters*, Journal of the Optical Society of America, 2 (1985), pp. 160–169.

[17] P. Dayan and L. Abbott, *Theoretical Neuroscience, Computational and Mathematical Modeling of Neural Systems*, MIT Press, 2001.

[18] R. Deriche, *Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector*, Intl. Journal of Computer Vision, 1 (1987), pp. 167–187.

[19] ——, *Recursively Implementing the Gaussian and its Derivatives*, Tech. Rep. 1893, INRIA, Unite de Recherche Sophia-Antipolis, 1993.

[20] D. Field, A. Hayes, and R. Hess, *Contour Integration by the Human Visual System: Evidence for a Local 'Association Field'*, Vision Research, 33 (1993), pp. 173–193.

[21] W. Freeman, E. Pasztor, and O. T. Carmichael, *Learning Low-Level Vision*, Intl. Journal of Computer Vision, 40 (2000), pp. 25–47.

[22] W. T. Freeman and E. H. Adelson, *The Design and Use of Steerable Filters*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 891–906.

[23] D. Geiger and F. Girosi, *Parallel and Deterministic Algorithms from MRFs: Surface Reconstruction*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 401–412.

[24] S. Geman and D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Treatment of Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 66 (1984), pp. 721–741.

[25] C. Gray, *The Temporal Correlation Hypothesis of Visual Feature Integration: Still Alive and Well*, Neuron, 24 (1999), pp. 31–47.

[26] S. Grossberg, *The Quantized Geometry of Visual Space: The Coherent Computation of Depth, Form and Lightness*, The Behavioral and Brain Scinces, (1983), pp. 625–657.

[27] ——, *Cortical Dynamics of Three-Dimensional Form, Color, and Brightness Perception, II:Binocular Theory*, Perception and Psychophysics, 41 (1987), pp. 117–158.

[28] ——, *Cortical Dynamics of Three-Dimensional Form, Color, and Brightness Perception, I:Monocular Theory*, Perception and Psychophysics, 41 (1987), pp. 87–116.

[29] ——, *The Adaptive Brain I, Cognition, Learning, Reinforcement, and Rhythm*, North-Holland, 1987.

[30] ——, *The Adaptive Brain II, Vision, Speech, Language and Motor Control*, North-Holland, 1987.

[31] ——, *Neural Networks and Natural Intelligence*, MIT Press, 1988.

[32] ——, *Nonlinear Neural Networks: Principles, Mechanisms and Architectures*, Neural Networks, 1 (1988), pp. 17–61.

[33] ——, *3-D Vision and Figure-Ground Separation by Visual cortex*, Perception and Psychophysics, 55 (1994), pp. 48–121.

[34] S. Grossberg and N. McLoughlin, *Cortical Dynamics of Three-Dimensional Surface Perception: Binocular and Half-Occluded Scenic Images*, Neural Networks, 10 (1997), pp. 1583 –1605.

[35] S. Grossberg and E. Mingolla, *Neural Dynamics of Form Perception: Boundary Completion, Illusory Figures, and Neon Color Spreading*, Psychological Review, 92 (1985), pp. 173–211.

[36] ——, *Neural Dynamics of Perceptual Grouping: Textures, Boundaries, and Emergent Segmentations*, Perception and Psychophysics, 38 (1985), pp. 141–171.

[37] ——, *Neural Dynamics of Surface Perception: Boundary Webs, Illuminants, and Shape from Shading*, Computer Vision, Graphics, and Image Processing, 37 (1987), pp. 116–165.

[38] S. Grossberg, E. Mingolla, and L. Viswanathan, *Neural Dynamics of Motion Integration and Segmentation Within and Across Apertures*, Vision Research, 41 (2001), pp. 2521–2553.

[39] S. Grossberg, E. Mingolla, and J. Williamson, *Synthetic Aperture Radar Processing by a Multiple Scale Neural System for Boundary and Surface Representation*, Neural Networks, 8 (1995), pp. 1005–1028.

[40] S. Grossberg and G. Swaminathan, *A Laminar Cortical Model for 3D Perception of Slanted and Curved Surfaces and of 2D Images: Development, Attention and Bistability*, tech. rep., 2004. `http://www.cns.bu.edu/Profiles/Grossberg/GroSwa2004VR.pdf`.

[41] S. Grossberg and D. Todorovic, *Neural Dynamics of 1-D and 2-D Brightness Perception: A Unified Model of Classical and Recent Phenomena*, Perception and Psychophysics, 43 (1988), pp. 241–277.

[42] G. Guy and G. Medioni, *Inferring Global Perceptual Contours from Local Features*, Intl. Journal of Computer Vision, 20 (1996), pp. 113–133.

[43] H. K. Hartline and F. Ratliff, *Inhibitory Interactions in the Retina of Limulus*, in Physiology of Photoreceptor Organs, M. G. F. Fuortes, ed., Springer Verlag, 1972, pp. 381–447.

[44] D. Heeger, *Optical Flow Using Spatiotemporal Filters*, Intl. Journal of Computer Vision, 1 (1988), pp. 279–302.

[45] F. Heitger, L. Rosenthaler, R. von der Heydt, E. Peterhans, and O. Kubler, *Simulation of Neural Contour Mechanisms: from Simple to End-Stopped Cells*, Vision Research, 32 (1992), pp. 963–981.

[46] F. Heitger and R. von der Heydt, *A Computational Model Of Neural Contour Processing: Figure-Ground Segregation And Illusory Contours*, in Intl. Conf. on Computer Vision, 1993, pp. 32–40.

[47] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1989.

[48] R. Heydt, E. Peterhans, and G. Baumgarthner, *Illusory Contours and Cortical Neuron Responses*, Science, 224 (1984), pp. 1260–1262.

[49] G. Hinton and T. Sejnowski, *Learning and Relearning in Boltzmann Machines*, in Parallel Distributed Processing, Rumelhart and McClelland, eds., vol. 1, MIT Press, 1986, ch. 7.

[50] A. L. Hodgkin and A. F. Huxley, *A Quantitative Description of Ion Currents and its Applications to Conduction and Excitation in Nerve Membranes*, Journal of Physiology, (1952), pp. 500–544.

[51] G. R. Holt and C. Koch, *Shunting Inhibition Does Not Have a Divisive Effect on Firing Rates*, Neural Computation, 9 (1997), pp. 1001–1013.

[52] J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Nat.l Academy of Sciences of USA, 79 (1982), pp. 2554–2558.

[53] ——, *Neurons with Graded Response have Collective Computational Properties like those of Two-State Neurons*, Proc. Nat.l Academy of Sciences of USA, 81 (1984), pp. 3088 – 3092.

[54] J. J. Hopfield and D. W. Tank, *Neural Computation of Decisions in Optimization Problems*, Biological Cybernetics, 52 (1985).

[55] B. Horn, *The Curve of Least Energy*, Tech. Rep. 612, MIT A.I. Lab, 1981. ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-612.pdf.

[56] D. Hubel, *Eye, Brain, and Vision*, Scientific American Library, 1988.

[57] T. Jaakkola, *Tutorial on Variational Approximation Methods*, in Advanced Mean Field Methods: Theory and Practice, MIT Press, 2000.

[58] H. J. Kappen, *Using Boltzmann Machines for Probability Estimation: A General Framework for Neural Network Learning*, in Intl. Conf. on Artificial Neural Networks, 1993, pp. 521–526.

[59] ——, *Deterministic Learning Rules for Boltzmann Machines*, Neural Networks, 8 (1995), pp. 537–548.

[60] F. Kelly and S. Grossberg, *Neural Dynamics of 3-D Surface Perception: Figure-Ground Separation and Lightness Perception*, Perception & Psychophysics, 62 (2000), pp. 1596–1619.

[61] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 1999.

[62] C. Koch, J. Marroquin, and A. Yuille, *Analog Neuronal Networks in Early Vision*, Tech. Rep. 751, MIT A.I. Lab, 1985. ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-751.pdf.

[63] I. Kokkinos, R. Deriche, P. Maragos, and O. Faugeras, *A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks*, in European Conf. on Computer Vision, 2004, pp. 506–517.

[64] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, *Statistical Edge Detection: Learning and Evaluating Edge Cues*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25 (2003).

[65] P. Kovesi, *Invariant Measures of Image Features From Phase Information*, PhD thesis, Department of Psychology University of Western Australia, 1996.

[66] T. S. Lee, *A Bayesian Framework for Understanding Texture Segmentation in the Primary Visual Cortex*, Vision Research, 35 (1995), pp. 2643–2657.

[67] ——, *Image Representation Using 2D Gabor Wavelets*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18 (1996), pp. 959–971.

[68] Z. Li, *Visual Segmentation by Contextual Influences via Intracortical Interactions in Primary Visual Cortex*, Network: Computation in Neural Systems, 10 (1999), pp. 187–212.

[69] D. Martin, *An Empirical Approach to Grouping and Segmentation*, PhD thesis, University of California, Berkeley, 2004.

[70] D. Martin, C. Fowlkes, and J. Malik, *Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 530–549.

[71] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, in Intl. Conf. on Computer Vision, 2001, pp. 416–423.

[72] E. Mingolla, W. Ross, and S. Grossberg, *A Neural Network for Enhancing Boundaries and Surfaces in Synthetic Aperture Radar Images*, Neural Networks, 12 (1999), pp. 495–511.

[73] D. Mumford, *Elastica and Computer Vision*, in Algebraic Geometry and its applications, B. J., ed., Springer Verlag, 1993, pp. 507–518.

[74] D. Mumford and J. Shah, *Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems*, Communications on Pure and Applied Mathematics, 42 (1989), pp. 577–685.

[75] H. Neumann and W. Sepp, *Recurrent V1-V2 Interaction in Early Visual Boundary Processing*, Biological Cybernetics, 91 (1999), pp. 425–444.

[76] P. Parent and S. Zucker, *Trace Inference, Curvature Consistency, And Curve Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 823–839.

[77] P. Perona, *Deformable Kernels for Early Vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (1995).

[78] P. Perona and J. Malik, *Detecting and Localizing Edges Composed of Steps, Peaks and Roofs*, in Intl. Conf. on Computer Vision, 1990, pp. 52–57.

[79] ——, *Scale-Space and Edge Detection Using Anisotropic Diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (1990), pp. 629–639.

[80] L. Pessoa, E. Mingolla, and H. Neumann, *A Contrast and Luminance-driven Multiscale Network Model of Brightness Perception*, Vision Research, 35 (1995), pp. 2201–2223.

[81] M. Proesmans, E. Pauwels, and L. v. Gool, *Coupled Geometry Driven Diffusion Equations for Low-Level Vision*, in Geometry Driven Diffusion in Computer Vision, Kluwer Academic Publishers, 1994.

[82] X. Ren and J. Malik, *A Probabilistic Multi-scale Model for Contour Completion Based on Image Statistics*, in European Conf. on Computer Vision, vol. 1, 2002, pp. 312–327.

[83] W. Ross, Grossberg, S., and E. Mingolla, *Visual Cortical Mechanisms of Perceptual Grouping: Interacting Layers, Networks, Columns and Maps*, Neural Networks, 13 (2000), pp. 571–588.

[84] E. Salinias and T. J. Sejnowski, *Gain Modulation in the Central Nervous System: Where Behavior, Neurophysiology, and Computation Meet*, Neuroscientist, 7 (2001), pp. 430–440.

[85] S. Seung, *Course Web-Page: Introduction to Neural Networks*. `http://hebb.mit.edu/courses/9.641/lectures/index.html`.

[86] A. Sha'ashua and S. Ullman, *Structural Saliency: the Detection of Globally Salient Structures Using a Locally Connected Network*, in Intl. Conf. on Computer Vision, 1988, pp. 321–327.

[87] B. t. H. Romeny, *Geometry Driven Diffusion in Computer Vision*, Kluwer Academic Publishers, 1994.

[88] M. Welling and G. E. Hinton, *A New Learning Algorithm for Mean Field Boltzmann Mahines*, Tech. Rep. 2001-002, Gatsby Computational Neuroscience Unit, 2001.

[89] L. Williams and D. Jacobs, *Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience*, Neural Computation, 9 (1997), pp. 837–858.

[90] L. Williams, J. Zweck, T. Wang, and K. Thornber, *Computing Stochastic Completion Fields in Linear Time Using a Resolution Pyramid*, Computer Vision and Image Understanding, 76 (1999), pp. 289–297.

[91] A. Yuille, *Energy Functions for Early Vision and Analog Networks*, Tech. Rep. 987, MIT A.I. Lab, 1987. `ftp://publications.ai.mit.edu/ai-publications/500-999/AIM-987.ps`.

[92] ——, *Winner-Take-All Networks*, in The Handbook of Brain Theory and Neural Networks, M. Arbib, ed., MIT Press, 1995.

[93] A. Yuille and D. Geiger, *A Common Framework for Image Segmentation*, Intl. Journal of Computer Vision, 6 (1991), pp. 227–243.

[94] S. Zucker, C. David, A. Dobbins, and L. Iverson, *The Organization Of Curve Detection: Coarse Tangent Fields And Fine Spline Coverings*, in Intl. Conf. on Computer Vision, 1988, pp. 568–577.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399