

Computational Analysis and Learning for a Biologically Motivated Model of Boundary Detection

Iasonas Kokkinos^a Rachid Deriche^b Olivier Faugeras^b
Petros Maragos^a

^a*School of Electrical and Computer Engineering, National Technical University of Athens*

^b*Institut National de Recherche en Informatique et Automatique, Sophia-Antipolis, France*

Abstract

In this work we address the problem of boundary detection by combining ideas and approaches from biological and computational vision. Initially, we propose a simple and efficient architecture that is inspired from models of biological vision. Subsequently, we interpret and learn the system using computer vision techniques: First, we present analogies between the system components and computer vision techniques and interpret the network as minimizing a cost functional, thereby establishing a link with variational techniques. Second, based on Mean-Field Theory the equations describing the network behavior are interpreted statistically. Third, we build on this interpretation to develop an algorithm to learn the network weights from manually segmented natural images. Using a systematic evaluation on the Berkeley benchmark we show that when using the learned connection weights our network outperforms classical edge detection algorithms.

Key words: Computer Vision, Biological Vision, Neural Networks, Boundary Detection, Perceptual Grouping, Variational Techniques, Mean Field Theory, Boltzmann Machines, Learning

1 Introduction

Low and mid-level vision tasks have been the object of extensive research in both the computational and biological vision communities. This has resulted in efficient algorithms for problems like edge detection [1–4], image denoising [5,6] and segmentation [7,8] on the one hand and an increasingly detailed understanding of the functions of neurons in the first stages of the visual cortex on the other [9–11]. However, despite significant early works like e.g.

[12–14], research in these two areas proceeds along loosely coupled paths, leaving largely unexplored the potential of their fruitful interaction.

In this paper we present a thorough computational and statistical analysis of a biologically motivated system for vision tasks. Starting with the Boundary Contour System/Feature Contour System (BCS/FCS) architecture developed by S. Grossberg, E. Mingolla and collaborators through a series of publications [15–22], we have significantly simplified it and pursued the construction, analysis and learning of a biologically motivated and computationally competent model of boundary detection.

1.1 Contributions and Paper Outline

Our approach is synthetic in nature, putting together pieces from interrelated fields; even though some ideas used in this paper have been proposed in the past, they have not been combined in a cohesive manner. In synthesizing such a multi-faceted system our main contributions are:

- A minimal model comprising the functionalities of the BCS/FCS architecture: In Sec. 3 we propose our model that uses recurrent neural dynamics and integrates multi-scale contour, surface and saliency information in a simple architecture.
- Interpreting the network’s function using variational techniques: In Sec. 4 we identify the corresponding computer vision problems and analyze our model computationally by providing a Lyapunov function of the network. This describes the network dynamics as minimizing a variational criterion.
- A statistical interpretation of the network’s function and an algorithm for learning the network weights: In Sec. 5 we consider the network neurons as a field of interrelated random variables and use Mean Field Theory to interpret its function. In Sec 5.4 we present a learning algorithm for our network that reduces ad-hoc choices in network design and results in improved performance.
- The systematic evaluation of the network’s performance on an edge detection task using ground truth data for complex natural images: Experimental results and systematic evaluations are provided in Sec. 6, where significant performance improvements are demonstrated.

Some initial results of this work have been presented in [23], while an extensive presentation of our research can be found as a technical report in [24].

2 Previous Work

In this section, after some elementary background on the human visual system we present previous work related to neural architectures that inspired the proposed model; pls. cf. [24] for a more extensive background review.

2.1 Visual Cortex Neurons

Most biologically motivated architectures for vision problems adopt the bottom-up ordering laid out in [12]: a preliminary feature extraction stage takes place in the retina, LGN and area V1 followed by the introduction of increasing invariance to phase, location and pose in areas V1, V2, V4 and IT.

Simple cells in layer V1 are tuned to specific spatial frequencies and locations, and their receptive fields are commonly modelled in terms of two-dimensional Gabor filters [25]. Complex cells combine the outputs of simple filters in a phase-invariant response [26,27], which can be modelled as the magnitude of convolution with a complex Gabor filter. Refinements to these approximations take into account the *horizontal* interactions between cells in these areas [28,29].

In area V2 Von der Heydt and colleagues [30] discovered ‘context aware’ cells that are active in the absence of stimuli in their receptive field, provided their neighboring cells are active. The role assigned to such cells is grouping the responses of isolated cells in area V1 and strengthening the responses of mutually consistent neurons, and they are commonly viewed as the neural substrate of perceptual grouping.

Despite the similarities between cells in areas V1 and V2, there is evidence [31,32] that area V2 is more strongly involved than V1 in higher level processes, like surface formation, segmentation and grouping. Further, recent research e.g. [33] proposes that areas V1 and V2 continuously interact with higher level areas. In our work we consider only the interaction between contour linking and boundary detection typically assumed to be mediated by V1-V2 interaction.

2.2 Models of Neuron Dynamics

Models of increased sophistication and accuracy for the function of individual neurons have been developed during decades of research [34], but when modelling a large number of neurons computationally convenient approximations are used.

The *Mean Firing Rate* (MFR) [34] approximation summarizes the behavior of a cell in terms of its average rate of spike emission: this results in tractable models that account for a significant part of network dynamics, but discards timing information. The interactions of networks of neurons are described in terms of Ordinary Differential Equations -ODE(s). The most common ODE is that of the additive model, where the MFR U_i of neuron i is influenced by the MFRs U_j of neurons j in its neighborhood $\mathcal{N}(i)$ as:

$$\frac{dV_i}{dt} = -AV_i + \sum_{j \in \mathcal{N}(i)} W_{i,j}U_j, \quad U_i = g(V_i). \quad (1)$$

The *potential* V_i of neuron i is related to its mean firing rate U_i via a nonlinear function g that keeps the firing rate positive and saturates for high values of

V . The leakage term $-AV_i$ drives the neurons to a neutral state in the absence of external inputs. If neuron i influences the firing rates of its neighbors, this results in a recurrent architecture that has nontrivial dynamics.

A more elaborate model of interactions [35] replaces Eq. (1) with:

$$\frac{dV_i}{dt} = -AV_i + (V_{max} - V_i)E_i + (V_{min} - V_i)I_i \quad (2)$$

where $E_i = \sum_{j \in \mathcal{N}(i)} W_j^E U_j$, $I_i = \sum_{j \in \mathcal{N}(i)} W_j^I U_j$.

The steady state potential V_i thus remains bounded in $[V_{min}, V_{max}]$ and does not saturate for arbitrarily high net excitatory (E_i) and inhibitory (I_i) inputs. Further, in [28] an additional term was introduced in Eq. (2):

$$\frac{dV_i}{dt} = -AV_i + (V_{max} - V_i)E_i + (V_{min} - V_i)I_i - V_i G_{sh}, \quad (3)$$

where G_{sh} is again a sum over the activity of neighboring neurons. The speciality of this last term is that the inhibition it causes to neuron i is only evident if neuron i is active, and it is therefore termed *shunting inhibition*. Actually, by solving for the steady state potential of neuron i we see that effect of G_{sh} is divisive rather than subtractive, hence it is commonly considered to be the substrate for *divisive normalization*.

Divisive normalization has successfully been used in [28,36] to explain the non-linear behavior of neuron ensembles and the suppression of responses by stimuli presented outside the classical receptive field. The latter phenomenon, known as *surround suppression* [37–39] has been modelled in [40] explicitly in terms of a division with a pooled sum of neighboring neurons, and was analyzed in [41,42] as a mechanism to reduce the response of typical edge detection operators to textured areas.

2.3 Neural Architectures for Low- and Mid- level vision problems

Issues concerning model complexity, accuracy and plausibility emerge when building models for whole areas of the visual cortex and their interactions. In the following we present three influential models of increasing compactness, followed by a brief discussion motivating the introduction of our model.

2.3.1 The BCS-FCS model

The FACADE (Form And Colour and DEpth) theory of vision proposed by S. Grossberg and his collaborators in a series of papers [20,21,15,17,22,43] deals with almost the whole of low- and mid-level vision, starting from edge detection and ending at motion segmentation and binocular vision.

Our focus has been on the Boundary Contour System (BCS) and Feature Contour System (FCS) subparts of this model [44], which are related to the tasks of boundary detection and image segmentation respectively.

In its full-blown version [22,20] the BCS consists of 6 processing stages, each comprising 4-8 orientations and 3 scales: *Stage I* models On-Off/Off-On cells in LGN. Their inputs are obtained from isotropic Gaussian filters with different

widths; e.g. On-Off cells combine an excitatory input from a small Gaussian with an inhibitory input from a broad Gaussian to detect bright blobs. *Stage II* models simple cells in V1, by convolving the outputs of Stage I cells with elongated Gaussians to extract orientation-sensitive responses. Two directions along each orientation are considered, since neurons cannot have negative outputs and nonlinearities are used to avoid saturation. *Stage III* captures the behavior of complex cells by pooling the responses of simple cells at opposite directions. This yields a single, invariant response signaling variation along an orientation.

In *Stages IV-VI* boundary formation takes place: Stages IV and V perform nonmaximum suppression, along location and orientation respectively, aided both by bottom-up information and a top-down saliency signal computed by Stage VI. The latter is obtained by pooling the responses of Stage V outputs, and is fed back to Stage IV yielding a recurrent process that detects and enhances smooth boundaries.

The Feature Contour System (FCS) combines input from On/Off-Off/On cells with the boundaries provided by the BCS.

Continuous surface percepts are formed by diffusing the surface neuron activities via a system of coupled ODEs. The coupling between neighboring neurons is loosened at locations where the BCS signals an edge, thereby achieving the anisotropic diffusion of the image content.

Many ideas in this model are intuitively appealing, while the plethora of psychophysical phenomena it explains in a unified manner [21] support its plausibility. Still, its high complexity renders it unmanageable and this led us to develop the simpler architecture presented in the following.

2.3.2 Bayesian model of texture segmentation in the Cortex [45]

The model of [45] is strongly influenced by ideas developed in the computer vision community related to variational image segmentation [7,46] and provides a mapping between computational and biological processing modules. It relies on a region-based diffusion process that smooths the responses of simple and complex cells. The introduction of a line process blocks diffusion at feature discontinuity points, yielding feature channels separated in smooth compartments. To avoid trivial oversegmentations of the image, a cost on the line process is introduced in the energy functional, guaranteeing line elements are introduced only where necessary. A physiological mechanism potentially underlying the minimization of the functional using an annealing technique is proposed and a neural circuitry implementation of the model is suggested.

This model can be seen as a mathematically sound version of the BCS/FCS model, allowing for a variational/Bayesian treatment. Still, it lacks both an edge detection and a saliency computation module, accounting only for region-based boundary detection. However, the abundance of biological evidence on boundary processes, and computational evidence of the merit of combining bottom-up edge detection with region-based cues [4] hint that edge detection is useful for any model.

2.3.3 Segmentation without Classification model [47]

In this model a minimal architecture is proposed to address the question whether separate feature and boundary streams are necessary. Using a model where all V1 cells interact with their neighbors in the same manner, the ability to perform texture segmentation without a separate feature processing stage is demonstrated. This contradicts typical computational [3,46,48] and biological [21,45] models which assume texture processing takes place in a separate stream that cooperates with boundary processing.

The compactness of this model is appealing, but the complexity of the role assigned to V1 cells seems daunting: feature extraction, boundary detection, nonmaximum suppression and saliency computation are assumed to be accomplished by the same cells. Even though evidence in favor [49] and against [50] such an architecture is not conclusive, for practical reasons we adopt a less involved approach at the stage of model building.

2.4 Discussion and motivation for the new model

In our understanding there is no ‘winner’ among the above and other models, e.g. [51,52]. Psychophysical phenomena do provide insight on the functionality of the computational modules, while the architecture of the visual cortex can constrain the possible models that explain the phenomena. However, the partiality of this knowledge leaves space for numerous models: as a first example, illusory contour formation has been modelled in [47] as being accomplished in V1 by horizontal interactions, in [51] as a gating feedback from V2, and in [44,21] as an additive feedback from V2. A second example concerns surface processes related to brightness perception [19,21] and texture segmentation [45]: these are commonly modelled with a surface stream complementary to the boundary stream, but in [47] this was explicitly omitted from the model, resulting again in plausible results. Psychophysical evidence in favor of separate region and boundary streams [50] is not conclusive, especially for texture-related channels, where the same simple and complex cells respond both when lying on a region boundary and in the interior of a textured region [31].

The bottom line is that few models come with actual physiological arguments in favor of their structure, while the focus is primarily on reproducing psychophysical phenomena. In our initial research efforts we had focused on the BCS/FCS model, primarily because it has been extended to account for almost the whole of low- and mid-level vision [21] and has been applied to natural images [22] giving promising results. In our implementation we faced practical problems with the large number of processing modules, design parameters and the lack of a mathematical analysis of its behavior. This led us to propose the simpler model presented in the following section. Our model uses a recurrent architecture comprising the main functionalities of the BCS/FCS architecture; however, the number of BCS processing stages is significantly reduced, while its recurrent dynamics facilitate its variational and statistical interpretation.

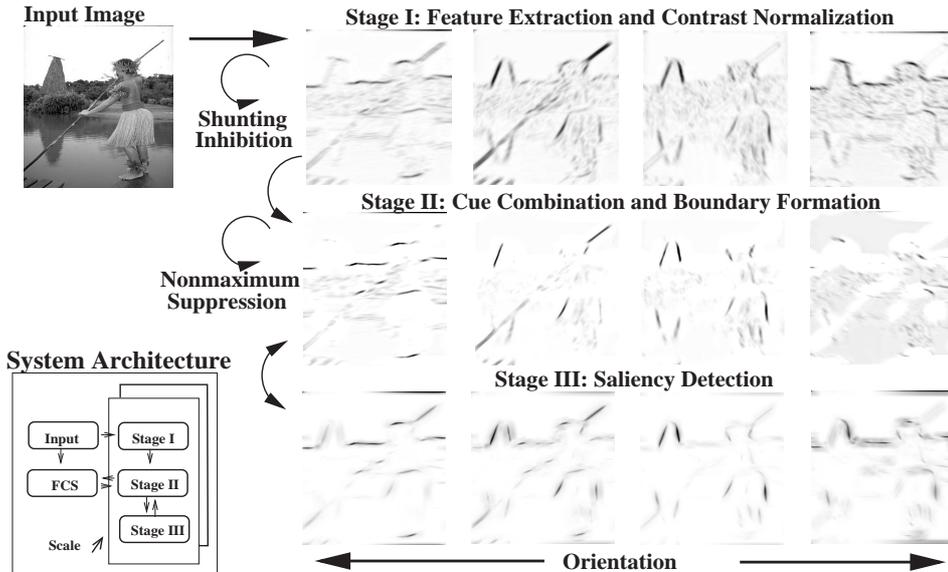


Fig. 1. Bottom left: Block Diagram of system module interactions. Right: Boundary computation across the three processing Stages.

3 A Recurrent Architecture for Boundary and Surface Processing

A visual roadmap of our system is shown in Fig. 1, where we show the neuron activations of the three processing stages used by our model at a single scale: feature extraction at multiple orientations takes place in Stage I and is followed by boundary formation in Stage II. There the bottom-up cues are recurrently combined with the top-down saliency signal computed from Stage III. Stages I and II follow recurrent dynamics, determined by their lateral interactions. Below we describe in more detail the function performed by each processing stage. Since in Section 5 we derive an algorithm to learn the connection weights among neurons from ground truth data we omit the heuristic expressions used in our earlier work [23,24].

3.1 Stage I: Feature Extraction, Divisive Normalization.

The first two stages (contrast and feature detection) of the BCS are merged into one, using Gabor filters for feature extraction; the optimality properties related to 2-D Gabor filters and their accurate modelling of simple cell receptive fields [25] suggest they constitute an adequate approximation to the feedforward input to simple cells.

We use the Gabor filterbank described in [45] since its parameters are chosen to comply with measurements of simple cell receptive field and its even-symmetric filters have zero DC components, which is necessary for modelling simple cells. The feedforward input to a single cell at location i, j with orientation preference θ , scale σ with an even-symmetric receptive field is given by

$$I_e^{\theta,\sigma}(i, j) = \Re(\Psi^{\theta,\sigma} * I)(i, j). \quad (4)$$

$\Psi^{\theta,\sigma}$ is a complex Gabor filter with scale σ and orientational preference θ , $*$ denotes convolution, I is the input image and \Re is the real part of a complex number. For odd-symmetric cells we use the imaginary part, $\Im: I_o^{\theta,\sigma}(i, j) = \Im(\Psi^{\theta,\sigma} * I)(i, j)$.

Shunting inhibition is used to account for surround suppression and contrast normalization as in [28], with a shunting term that changes dynamically based on the activations of neighboring cells. The equation driving the potential V_i of neuron i writes:

$$\frac{dV_i}{dt} = -AV_i + (V_{max} - V_i)I_i - V_i \sum_{j \in \mathcal{N}(i)} W_{i,j}U_j, \quad (5)$$

$$V_i^\infty = \frac{V_{max}I_i}{A + I_i + \sum_{j \in \mathcal{N}(i)} W_{i,j}U_j} \quad (6)$$

where $U_i = \max(V_i, 0)$ is the MFR of neuron j , $\mathcal{N}(i)$ the neighborhood of neuron i and $W_{i,j}$ denotes the strength of the interaction between i and j . Starting from Eq. (5), a neuron at location k, l and with orientation θ is indexed using the one-dimensional index i for simplicity.

The output sent to Stage II is obtained by combining the outputs of odd- and even- symmetric cells in a phase-invariant term as in [26,53,3]:

$$O = \sqrt{U_o^2 + U_e^2} \quad (7)$$

where U_e, U_o are the steady-state responses of even- and odd- symmetric Stage I neurons at the same location, orientation and scale. The merits of combining even and odd-symmetric filter as in Eq. (7) have been established in [3] while such a scheme has been used to model the behavior of complex cells in [26] and the perception of boundaries in [53]; biological evidence supporting it is reviewed in [27].

Since neurons can have only positive firing rates, one should separate positive and negative cell responses and perform the normalization in parallel for each part. In practice we observed that initially estimating $I = \sqrt{\Re(\Psi^{\theta,\sigma} * I)^2 + \Im(\Psi^{\theta,\sigma} * I)^2}$, and subsequently performing the normalization process on this single channel yields hardly discernible results.

3.2 Stage II: Edge Thinning, Contour Formation.

The function of this stage lies at the heart of the boundary detection process: in this stage the combination of bottom-up (Stage I), top-down (Stage III), region-based (FCS) and multi-scale information takes place, while simultaneously performing edge thinning.

Edge sharpening in space and orientation is simultaneously accomplished via horizontal inhibitory connections. Thereby the activities of the most active neurons are enhanced to the detriment of their less active neighbors.

Saliency information and *coarse-scale* boundary detection results are provided from other modules, enhancing the activity of neurons lying on perceptually

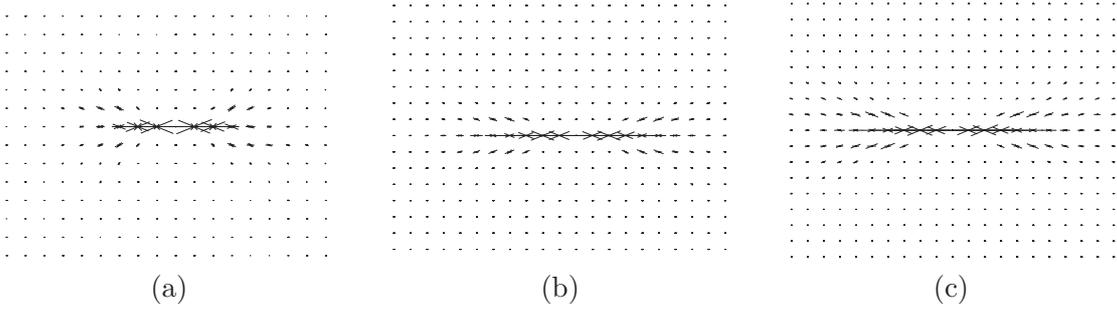


Fig. 2. The lobes derived from the Elastica model (defined in Eq. (12)) of curves for parameters (a) $\lambda = 10, \sigma = 0.4$, (b) $\lambda = 10, \sigma = 0.2$, (c) $\lambda = 20, \sigma = 0.2$.

strong borders, or borders appearing at multiple scales.

The *surface process*, S , influences the evolution by favoring the formation of edges at locations of feature discontinuities. These are indicated by the magnitude of the directional derivative of S perpendicular to orientation θ , $|\nabla S_{\theta^\perp}|$.

Combining all these cues, the evolution of neuron i potential writes:

$$\frac{dV_i}{dt} = -AV_i + (V_{max} - V_i)I_i + (V_{min} - V_i) \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j \quad (8)$$

$$I_i = [c_1 O + c_2 C + c_3 T + c_4 |\nabla S_{\theta^\perp}|] .$$

The excitatory input I_i is formed by combining T , the top-down signal provided from Stage III, C , the coarser scale term arriving from an homologous system functioning at a larger scale, O , the oriented-energy feedforward term in Eq. (7) and the FCS-derived signal. The sigmoidal function

$$U_i = g(V_i) = \frac{1}{1 + \exp(-(\frac{V_i}{\beta} - \gamma))} \quad (9)$$

relates the potential of a neuron to its MFR, with β determining the steepness of the input-output relation and γ acting like a bias term.

3.3 Stage III: Saliency Computation

At this stage salient contours are detected by pooling information from the neighborhood of each neuron according to the connection pattern shown in Fig. 2; as will be explained later, the interconnection pattern shown in Fig. 2 depends on two design parameters λ, σ corresponding to the scale and spread of the lobes, respectively.

For each direction of arrival (left/right in the figure), a weighted sum is computed and the two sums are combined to guarantee that evidence in favor of a contour exists on both sides.

For this we use the evolution equation:

$$\frac{dV_i}{dt} = -AV_i + \left(\sum_{j \in N^+(i)} W_{i,j}^+ U_j \right) \left(\sum_{j \in N^-(i)} W_{i,j}^- U_j \right) \quad (10)$$

where N^+, N^- are the subsets of neighbors lying on the two sides of i , W^+, W^-

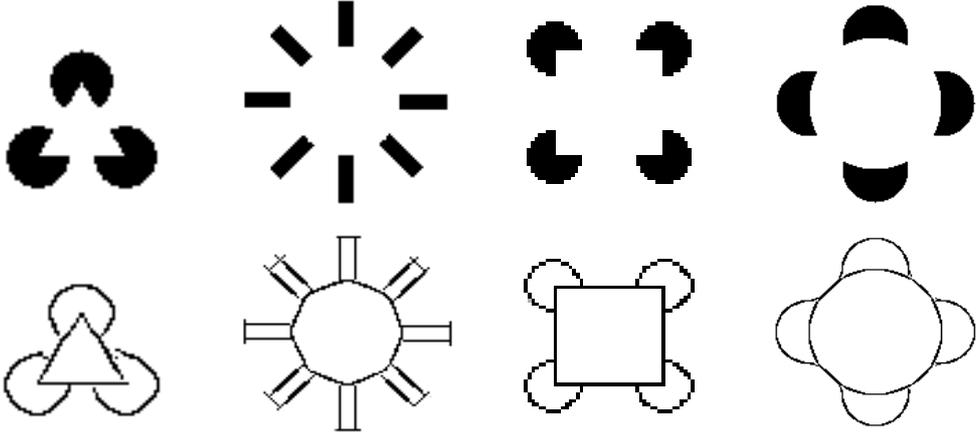


Fig. 3. The top row figures give rise to the illusory boundaries of the bottom row. These are detected in Stage II, thanks to the saliency signal from Stage III.

are the corresponding connection weights and U_j is the MFR of neuron j , derived from V_j via rectification. We used multiplication in Eq. (10) assuming the neural circuitry proposed in [54,51] can approximate it sufficiently well. The contours detected at this stage are fed back to Stage II, resulting in a cooperative process that favors smooth and long boundaries. Illusory contours can thereby be perceived, as shown in Fig. 3.

3.4 FCS: Anisotropic Image Smoothing

Two practical differences of our model’s FCS from the original are that, first, the intensity values of the image are used instead of the On/Off- Off/On cell outputs and second, we consider BCS neurons are located between FCS neurons as in [55]. The evolution of an FCS neuron potential S writes:

$$\frac{d}{dt}S = \sum_{\theta} \nabla_{\theta^{\perp}} S (1 - U_{\theta}), \quad (11)$$

where U_{θ} is the MFR of the Stage II neuron having orientation θ and lying between the neighbors used to estimate $\nabla_{\theta^{\perp}} S$. As in [6] the value of S at neuron i is always the subtracted quantity in $\nabla_{\theta^{\perp}} S$, i.e. for a discrete grid $\nabla_{\theta^{\perp}} = S' - S_i$ where S' is the closest neighbor to i along orientation θ^{\perp} .

In our model surface formation interacts with boundary detection, keeping the boundaries close to maxima of brightness gradients. On synthetic images we observed that this avoids the occasional shifting of edges due to higher level (Stage III) cues and the breaking up of corners eventually caused by orientational competition. This interaction can also be useful if region-based cues drive boundary detection, e.g. for texture segmentation [45].

3.5 Edge Detection Results

Some results of our system on real-world images are shown in Fig. 4 where its performance is compared to Canny edge detection [1] with approximately

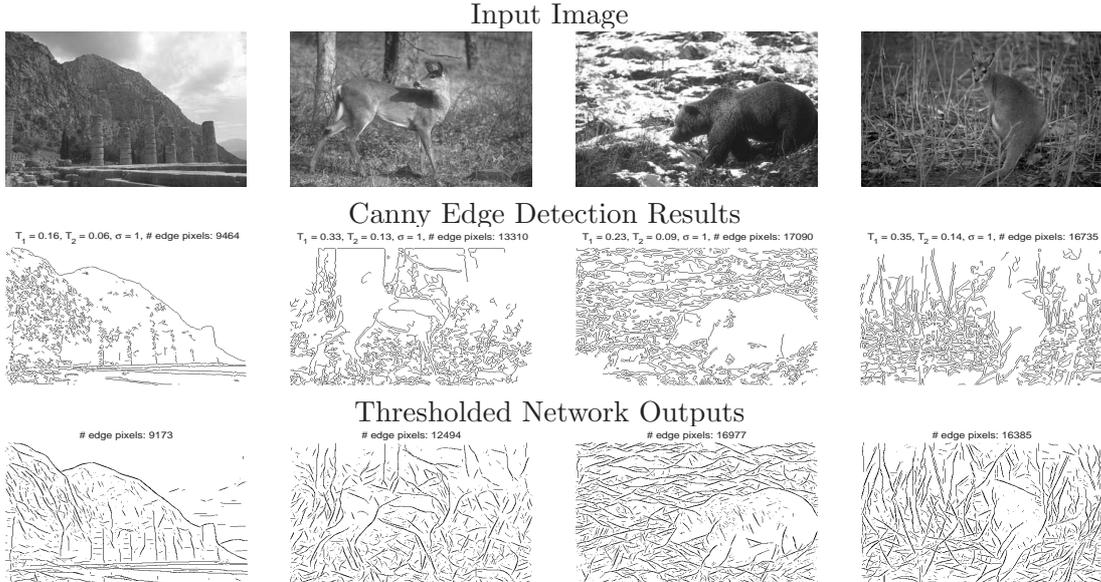


Fig. 4. Comparison between Canny edge detection and the results of our network. The thresholds for both methods have been set to return the same number of edge elements. Canny edge detection yields a large number of edges on textured areas, while our network focuses on salient borders.

the same number of pixels. Our system favors smooth and long contours, occasionally even hallucinating them as in the image of the deer: the edge starting from its leg is unified with the one due its head’s shadow. In general, however, it is a bias that pays off, compared to plain Canny edge detection. Specifically the latter suffers at coarse scales from poor edge localization while at finer scales edges are detected at textured regions. Our model avoids these problems due to both its multi-scale architecture and its surround suppression mechanism. The benchmark results presented in Sec. 6 validate that the manually designed model initially presented in [23] systematically outperforms fine scale Canny edge detection, but for a coarser scale the improvement of our model is only evident once learning is introduced.

4 Interpreting the Model in Computer Vision Terms

Most of our network functions correspond to common computer vision techniques, which help analyze the network’s function mathematically. We start with its saliency estimation module and continue with analyzing the function of the network as a whole.

4.1 Perceptual Grouping and the Elastica Prior on Shapes:

The enhancement of perceptually salient borders has been extensively studied in computer vision during the last decades; early work includes the hysteresis-thresholding technique of [1] and the penalty terms on line endings used in [56,5] which favor edge groupings. In the probabilistic formulations used in

[57,58] edge saliency is propagated among processing nodes, while in [59] a variational criterion including the squared integral of the curvature is used for edge linking.

In work on contour completion that is more closely related to ours, like the tensor voting method of [60], the relaxation labelling process of [61] and the work of [52] a similar pattern of edgel interactions to that of Fig. 2 is used to identify salient locations. Actually, it is shown in [58] that this pattern, initially introduced by S. Grossberg in [15], is intrinsically related to variational approaches to contour completion [59,62]. According to the *Elastica* model of [62] a curve Γ is the path of a particle driven by a diffusion process in the continuous (x, y, θ) space where x, y are the particle's position and θ is the orientation of the tangent to its trajectory at that point. The motion of the particle is described by the stochastic differential equation:

$$\frac{d}{dt}x = \cos(\theta) \quad \frac{d}{dt}y = \sin(\theta) \quad \frac{d}{dt}\theta = \kappa(t), \quad \kappa(t) \sim \mathcal{N}(0, \sigma^2) \quad (12)$$

where the curvature, κ , of the associated curve is a Gaussian white noise process. The length of the curves is constrained by stopping the diffusion process at a time instant drawn from an exponential distribution with parameter λ .

The probability $P(\Gamma)$ of a curve is then related to the *Elastica* functional:

$$E(\Gamma) = \int_{\Gamma} (\alpha k^2 + \beta) ds \quad (13)$$

The smoothness of the curves is determined by the parameter α and β is related to their expected length. In [62] it is shown that according to Eq. (12)

$$P(\Gamma) \propto e^{-\int_{\Gamma} (\alpha k^2 + \beta ds)}, \quad \beta = \lambda, \quad \alpha = \frac{1}{2\sigma^2} \quad (14)$$

Minimizers of the *Elastica* energy are thus modes of $P(\Gamma)$.

The link between this model and the shape of the lobes was established in [58]: based on Eq. (14) the posterior probability of a curve passing through a point $X_0 = (x_0, y_0, \theta_0)$ was expressed as a product of weighted sums:

$$P(X_0) = \left(\sum_{i \in r(X_0)} g(X_0, X_i) P(X_i) \right) \left(\sum_{j \in l(X_0)} g(X_0, X_j) P(X_j) \right).$$

Here g is a six-dimensional tensor on $(x, y, \theta, x', y', \theta')$ and equals the probability that a particle will pass through (x', y', θ') given that at time $t = 0$ it is located at x, y, θ . $r(X_0)$ and $l(X_0)$ stand for the neighborhoods on the two sides of point X_0 and $P(X_i)$ is the probability of a curve starting from X_i at $t = 0$. Using Monte Carlo simulation to approximate the tensor g results in the patterns shown in Fig. 2.

The feedback term is thus related to the posterior probability of a contour passing through a point conditioned on its surroundings. This justifies the specific shape of the lobes while simplifying their design since only two easily interpretable parameters, λ and σ , are involved. This results in a drastic reduction in the number of parameters required, compared e.g. to the expressions used in [22,51].

4.2 A Variational Perspective

The BCS/FCS architecture parallels the use of line and surface processes by computer vision researchers [5,55,45,56]; the general approach is to introduce a line process and diffuse the image intensity (termed ‘surface process’) in the whole image except for areas where a line process element is active. Discontinuities of the surface process determine where a line process element can be afforded and vice versa.

The variational approach phrases the task as the minimization of a functional, as e.g. that of Mumford and Shah [7]:

$$\mathcal{E}(S, U) = \lambda \int_D (S - I)^2 + \mu \int_{D-U} |\nabla S|^2 + \nu |U|. \quad (15)$$

In Eq. (15) I is the observed image intensity, S is the smooth reconstructed surface, U is a line process, $|U|$ is the length of the line process and D is the image domain. Finally, μ, ν are weighting factors determining the relative contribution of each term.

Variational problems in computer vision can be solved using recurrent neural networks [13,45,63,55] based on their link with Lyapunov functions [64,35,65,66]. A Lyapunov function is a positive definite function of the system’s state that constantly decreases as the system evolves; one can then view the function of the network as driving its state towards a minimum of this function.

Our goal is to see how our network can be analyzed in variational terms; based on [35] one can directly derive a Lyapunov function for the recurrent network described in the previous section but the integrals become involved and do not help intuition. We therefore consider the simplified version of Eq. (8):

$$\frac{dV_i}{dt} = -AV_i + CI_i - D \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j \quad (16)$$

where instead of the synaptic interaction among neurons we use the common sum-of-inputs model.

4.2.1 Boundary Processing Module

Initially we treat the excitatory input I as constant, in which case a Lyapunov function of the network is (see Appendix A):

$$\sum_i \left[\underbrace{a \int_{1/2}^{U_i} g^{-1}(u) du - cI_i U_i}_{\text{P(U): Penalty for values of U}} + \frac{d}{2} \underbrace{\sum_{i,j \in \mathcal{N}(i)} U_i W_{i,j} U_j}_{\text{C(U): cost of configurations of U}} \right] \quad (17)$$

For the sigmoidal function used in Eq. (9) we have

$$P(U) = [U \ln(U) + (1 - U) \ln(1 - U)] / \beta + \gamma U - \gamma^2. \quad (18)$$

This consists of a negative entropy term $[U \ln(U) + (1 - U) \ln(1 - U)]$, punishing binary responses and the term γU that punishes high responses. The first term is due to using a sigmoid transfer function in Eq. (9) and the second is due to shifting it by γ to the right. The factor β is related to the slope of the sigmoid function: if a rapidly increasing sigmoid is used, i.e. β is large, the

penalty on binary responses becomes less important and vice versa. The term $-IU$ lowers the cost of a high U , facilitating the emergence of a boundary at locations where I is large.

The weight $W_{i,j}$ expresses the interaction among neurons $i = (x_i, y_i, \theta_i)$ and $j = (x_j, y_j, \theta_j)$; taking into account the natural requirement of translation and rotation invariance and assuming symmetric connections among neurons of the same stage, a more elaborate notation would be $W_{i,j} = W^{\theta_i, \phi_j}(x_i - x_j, y_i - y_j)$. A special case we examined in [23,24] expresses the connection weights $W^{\theta, \theta}(x, y)$ among neurons of the same orientation θ as the difference between an isotropic filter G_s and an even-symmetric filter elongated along dimension θ , G_s^θ ; for different orientations, $W^{\theta, \phi}(i, j)$ is expressed in terms of an isotropic filter, G_d . For each of these filters there is another filter, g , such that $g * g = G$, and it can be shown that the g filters corresponding to elongated G filters are elongated along the same orientation.

Using these kernels the configuration cost writes [24]:

$$C(U) = \sum_{i,j,\theta} [g_{is} * U_\theta]^2 - \sum_{i,j,\theta} [g_\theta * U_\theta]^2 + 1/2 \sum_{i,j,(\theta,\phi):\theta \neq \phi} [g_d * U_\theta] [g_d * U_\phi] \quad (19)$$

Here U_θ denotes the set of neurons of orientation θ and g_{is} is an isotropic kernel so $[g_{is} * U_\theta]^2$ punishes in general broad activation patterns, irrespective of whether the edge elements are collinear and consistent or simply scattered around. g_θ is an elongated filter with principal axis along the preferred orientation of neurons, θ . The $-[g_\theta * U_\theta]^2$ term is thus 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. The penalties on $[g_s * U_\theta]^2$ and $[g_d * U_\theta] \cdot [g_d * U_\phi]$ lead to the suppression of broad structures, while the reactive term $-[g_\theta * U_\theta]^2$ favors 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. The behavior of Winner-Take-All-type networks used for nonmaximum suppression in [55] can be seen as a special case of this setting.

4.2.2 Boundary-Surface Interaction

To put all pieces together we use the interaction with the surface process that evolves according to Eq. (11); as shown in Appendix A, a Lyapunov function is given by:

$$\mathcal{E} = \sum_i c_4 \underbrace{(1-U)|\nabla_\theta S|^2}_{\text{Line - Surface interaction}} - U \underbrace{[c_1 O + c_2 T + c_3 C]}_{\text{External inputs}} + \underbrace{a \int_{1/2}^U g^{-1}(u) du + \frac{d}{2} C(U)}_{\text{Cost for line process}} \quad (20)$$

where the constants $c_{1,\dots,4}$ have absorbed c of Eq. (16). This function has a lower bound, since the neuron outputs in the subtracted terms cannot become larger than 1, so adding this lower bound to \mathcal{E} makes it positive. By differentiating with respect to S and U we get the ODE in equations (11) and (16) respectively.

4.2.3 Relation to other variational models

Having derived an expression for the energy minimized by our network we can place it in the context of existing variational methods. First, we can see our functional as a more complex version of that introduced in [56], where a simple penalty term is used to enforce nonmaximum suppression and contour continuity to the anisotropic diffusion-derived line process. We also mention the similarity with [67], where the evolutions of the line and surface process are coupled, resulting in a conceptually similar model. Finally, adding a data fidelity term to the surface evolution process would result in a more sophisticated version of line process-based algorithms for the minimization of the Mumford-Shah functional.

Our system however does not simply boil down to trying to find an approximate minimum of the Mumford-Shah functional. Edge detection is more sophisticated 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 information used related to boundary detection is the variation of the reconstructed surface process in the 4-pixel neighborhood of each location. This ignores the architecture of the visual system, which has filters integrating evidence over large regions to facilitate bottom-up edge detection.

5 Statistical Analysis and Network Learning

The use of variational criteria similar to those presented in the previous section is widespread in computer vision; yet little work has been done on *learning* the variational criteria from ground truth data. This research direction is receiving increasing interest, cf. e.g [68,69] and in this section we present a probabilistic treatment of our network’s function that allows us to derive a learning algorithm for the connection weights.

Considering the outputs of Stage II as posterior probabilities of boundaries conditioned on the input image, we phrase our goal as the minimization of a probabilistic distance measure between the network outputs and ground-truth edge probabilities. Specifically, we obtain ideal edge detector outputs from human generated segmentations from the Berkeley Segmentation benchmark [70] and target the modification of the network connections towards the minimization of the Kullback Leibler divergence between the network outputs and the ground truth data.

Since our network is similar architecturally to the Boltzmann Machine (BM), we present in Sec. 5.1 the basic notions and previous work on learning algorithms for the BM. In Sec. 5.2 we introduce the Mean Field Approximation (MFA) and relate it to the function of our network. In Sec. 5.4 we derive the learning algorithm for the network weights.

5.1 The Boltzmann Machine

The Boltzmann Machine [71,72,66,73,74] is a probabilistic network of symmetrically connected binary units, which are separated in visible and hidden units, X and Y respectively. An energy function of the form

$$E(X, Y) = - \left(\frac{1}{2} Y^T V Y + \frac{1}{2} X^T W X + \frac{1}{2} Y^T J X \right) \quad (21)$$

is typically used for BMs, where V, W, J are symmetric matrixes that determine the intra/inter module interactions. A Boltzmann-Gibbs probability distribution of the network's state can be defined in terms of $E(X, Y)$ as:

$$P_{BM}(X, Y) = \frac{1}{Z} \exp(-E(X, Y)), \quad (22)$$

where $Z = \sum_{X, Y} \exp(-E(X, Y))$ is the partition function. This equation can also be written in a factorized form as:

$$P_{BM}(X, Y) = \prod_{i, j \in \mathcal{E}_{X, Y}} \Phi_{i, j}(X_i, Y_j) \prod_{i, j \in \mathcal{E}_{X, X}} \Psi_{i, j}(X_i, X_j) \prod_{i, j \in \mathcal{E}_{Y, Y}} \Xi_{i, j}(Y_i, Y_j) \quad (23)$$

where $\mathcal{E}_{X, Y}, \mathcal{E}_{X, X}, \mathcal{E}_{Y, Y}$ are the pairs of interacting units while Φ, Ψ, Ξ are determined by Eq.s (21) and (22).

The BM defines a probability distribution on its observable nodes, Y as $P_{BM}(Y) = \sum_X P_{BM}(Y, X)$. Parameter estimation for a BM is phrased as the minimization of the Kullback-Leibler (KL) divergence between P_{BM} and the 'environmental' [72] distribution $P(Y)$ of the observable units:

$$KL(P(Y)|P_{BM}(Y)) = \sum_Y P(Y) \ln \frac{P(Y)}{P_{BM}(Y)} \quad (24)$$

From the two terms formed by breaking the fraction within \ln , only the term $-\sum_Y P(Y) \ln P_{BM}(Y)$ is actually of interest, since the other term, $\sum_Y P(Y) \ln(P(Y))$, is independent of the BM and equals minus the entropy of Y . Training a BM by minimizing Eq. (24) changes the network weights in Eq. (21) to bring $P_{BM}(Y)$ closer to the observed $P(Y)$, since $KL(P(Y)|P_{BM}(Y))$ reaches its minimum -zero- when $P_{BM}(Y) = P(Y)$.

The original training algorithm for BMs [72] updates the weights according to

$$\Delta W_{i, j} \propto \langle v_i v_j \rangle^+ - \langle v_i v_j \rangle^- \quad (25)$$

where $\langle v_i v_j \rangle^+$ is the correlation of nodes v_i, v_j when the observable nodes are fixed to their observed values and $\langle v_i v_j \rangle^-$ when the network is running free. Estimating these means is typically performed via Monte Carlo techniques which entail stochastic simulation of the network, and replacing the expectations with a sum over the samples. This is computationally demanding and therefore impractical for a network as large as the one we use here; the need for less accurate, yet more efficient inference algorithms [74,73] emerges, and we now show that this is performed by the evolution equations of Sec. 4.

5.2 Mean Field Approximation

A popular alternative to Monte Carlo estimation is the variational approach; pls. cf. [75] for an excellent introduction to the subject on which this section is based. If the distributions Q to be inferred are assumed to belong to a specific family \mathcal{Q} , inference is phrased as the search for the distribution $Q \in \mathcal{Q}$ maximizing a criterion $J(Q)$ that favors the proximity of Q to the target distribution. A common inference problem concerns estimating the posterior distribution $P(X|Y)$ of a set of random variables X given a set of observations, Y . A suitable criterion to maximize for this purpose is:

$$J(Q) = \ln P(Y) - KL(Q(X)|P(X|Y)) \quad (26)$$

The subtracted term is always positive and equals zero for $Q(X) = P(X|Y)$, whereupon $J(Q)$ attains its maximum. If $P(X|Y)$ does not belong to \mathcal{Q} , maximizing $J(Q)$ will choose the member of \mathcal{Q} which is closest to $P(X|Y)$.

5.2.1 MFA for the Boltzmann Machine

Using the identities $P(Y)P(X|Y) = P(X, Y)$ and $\sum_X Q(X) = 1$, we can write Eq. (26) as:

$$J(Q) = \sum_X Q(X) \ln P(Y) - Q(X) \ln \frac{Q(X)}{P(X|Y)} = \sum_X Q(X) \ln P(X, Y) - Q(X) \ln Q(X)$$

If $P(X, Y)$ has the form of Eq. (23), the above expression writes:

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

All terms of Eq. (23) involving only pairwise potentials among observable units have been absorbed in the constant c , while $S = -\sum_X Q(X) \ln Q(X)$ is the entropy of the variational distribution.

Considering that \mathcal{Q} belongs to a specific family of distributions can further simplify this criterion. Specifically, we use the Mean Field Approximation (MFA) [75,66,55] according to which the joint distribution $P(X|Y)$ over $X = \{X_1, \dots, X_N\}$ is approximated in terms of individual distributions Q_i by considering the X_i as independent:

$$P(X|Y) \simeq Q(X) = \prod_i Q_i(X_i) \quad (27)$$

We can thus write for $J(Q)$

$$J(Q) = \sum_{i,j \in \mathcal{E}_{X,X}} \sum_{X_i, X_j} Q_i(X_i) Q_j(X_j) \ln \Psi_{i,j}(X_i, X_j) + \sum_i \sum_{X_i} Q_i(X_i) \ln \Phi_i(X_i) + \sum_i S_i \quad (28)$$

where we have used \mathcal{E}_X instead of $\mathcal{E}_{(X,X)}$, have dropped the c constant and have absorbed all the observed-hidden node interactions for node i into its *observation potential* Φ_i :

$$\sum_{j \in \mathcal{N}(i)} \sum_{X_i} Q_i(X_i) \ln \Phi_{i,j}(X_i, Y_j) = \sum_{X_i} Q_i(X_i) \ln \Phi_i(X_i). \quad (29)$$

Finally, the term $\sum_i S_i$ results from the assumption of independent random variables, which breaks the entropy S of distribution $Q(X)$ into the sum of entropies of the individual distributions, $S_i = \sum_{X_i} Q_i(X_i) \ln Q_i(X_i)$.

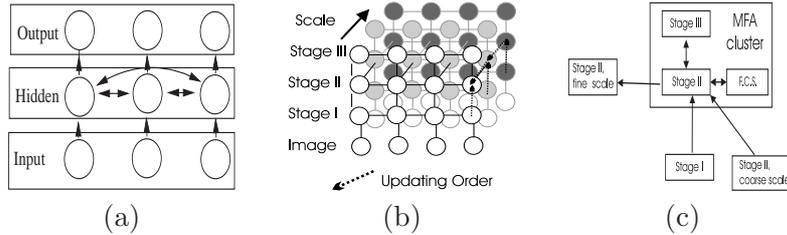


Fig. 5. (a) Boltzmann Machine learning: The input and outputs of the system are considered given training, and the task is to learn the hidden node connection weights. During testing, the output is computed by the network. (b) Coarse-to-fine flow of information. (c) MFA approximation over clusters of neurons.

5.2.2 Network Evolution equations as MFA

If the distributions $Q_j(\cdot), j \in \mathcal{N}(i)$ are kept fixed it is straightforward to estimate the individual distribution Q_i maximizing $J(Q)$. For binary variables estimating Q_i amounts to estimating the probability that node i is active, $Q_i(1) \equiv Q_i(X_i = 1) = 1 - Q_i(X_i = 0)$. Setting the derivative of $J(Q)$ with respect to $Q_i(1)$ equal to zero we have:

$$\ln \frac{Q_i(1)}{1 - Q_i(1)} = \ln \frac{\Phi_i(1)}{\Phi_i(0)} + \sum_{j \in \mathcal{N}(i)} \sum_{X_j} Q_j(X_j) \ln \frac{\Psi_{i,j}(X_j, 1)}{\Psi_{i,j}(X_j, 0)}.$$

For the BM, where 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 simplifies to:

$$\ln \frac{Q_i(1)}{1 - Q_i(1)} = \sum_{j \in \mathcal{N}(i)} Q_j(1)C_{i,j} + C_i. \quad (30)$$

This gives us the following expression for the $Q_i(1)$ that maximizes the variational criterion of Eq. (28):

$$Q_i(1) = \frac{1}{1 + \exp(-\sum_{j \in \mathcal{N}(i)} C_{i,j}Q_j(1) - C_i)}. \quad (31)$$

We know observe that the ODEs in Eq. (16) for Stage II lead to steady-states of the form Eq. (31) as can be seen by setting the time derivative to zero and using Eq. (9) to express the mean firing rate of the neurons. This means that solving the system of ODE's determining the activity of Stage II neurons amounts to performing the Mean Field Approximation on a Boltzmann Machine.

Formally, the Mean Firing Rates $U_i = g(V_i), i = 1, \dots, N$ provide the expected values $Q_i(1)$ of the binary variables X_i that bring the distribution $Q(X) = \prod_i Q(X_i)$ closest in KL distance to $P(X|Y)$. Intuitively, we interpret X_i as indicating whether neuron i is active; this constitutes the link between stochastic neural networks and variational-probabilistic approaches to vision [63,55,66].

5.3 Adapting BM learning to our network

To make the link with the BM we consider first our network function as a Boltzmann machine, shown in Fig. 5(a), and consider the training and testing

scenarios. The output of our system is delivered by Stage II neurons at the finest scale, and all other neurons are considered hidden. During training the input and output are given as the image and its boundary map. In the terminology of the previous section, during testing 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 training case the observables are both the image intensities and the states of the neurons at Stage II, which are clumped to the manual edge detection results.

The goal of training is to maximize the likelihood of the observed edge and image pairs contained in the training set. This requires integrating out the activities of non-observable neurons and can be approximated via the MFA. Before proceeding to the learning rules, we discuss some technical issues.

First, 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 minimized by the evolution equations derived from [35] cannot be written in the form used for the BM. However, we still consider the outputs of that stage as the means of binary random variables, signalling the existence or absence of an edge.

Second, 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, considering only the fine-scale Stage II neuron outputs as clamped. To optimize the behavior of each module separately we consider instead the desired distributions for each stage known in advance, following the coarse-to-fine updating scheme of Fig. 5(b). 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. An exception is that we update the values of Stage III and FCS nodes in parallel with Stage II, which amounts to performing MFA over the cluster of nodes as shown in Fig. 5(c).

Further, concerning the desired outputs of Stage I neurons used for training, we note that the objective in Stage I is to elicit in a contrast-invariant manner areas of high edge probability, while avoiding responses on textured regions, leaving the thinning of broad maps as a separate task for the following stage. We therefore use Gaussian filtering to smooth the ground truth edge probability maps and use them as ideal outputs. The Gaussian filter variance is half that of the Gabor filters used for feature extraction at the corresponding scale. Similarly, for the coarser scales of Stage II, we derive the desired outputs by smoothing the ground truth maps with a Gaussian having half the spread of that used for Scale I, thereby enforcing edge thinning.

Finally, some constraints are hard-wired in network training: first, connection weights are forced to be greater or equal to zero, thereby constraining horizontal interconnections to be only of the inhibitory form. Excitatory long range

interactions are mediated only from the feedback term (Stage III) thereby deconvoluting the role of each stage. Second, the updates to the connection weights among neurons are appropriately averaged to guarantee rotation invariance in the connection pattern.

5.4 Estimating the Network Weights

Having described the statistical interpretation of our network’s function and established the analogies with Boltzmann Machines we can now present the learning rules for estimating the network weights.

The training data made available in [70] are of the form $Y_{obs} = \{Y_1, \dots, Y_n\}$ 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:

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

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 [73,74] 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

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

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:

$$L' = \frac{1}{N} \sum_{n=1}^N \sum_i \sum_{b=\{0,1\}} P_{i,n}(b) \ln(Q_{i,n}(b)) \quad (32)$$

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 Eq. (5) as $Q_i(1)$ in Eq. (32):

$$Q_i(1) = \frac{I_i}{A + I_i + \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j}. \quad (33)$$

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.

The expression for the KL divergence derived in Appendix B yields the following update rule for the connection weight among neurons i and j :

$$W_{i,j} \leftarrow W_{i,j} + \alpha \left[\sum_i \frac{P_i(0) U_j}{A + \sum_k W_{i,k} U_k} - \frac{U_j}{A + I_i + \sum_k W_{i,k} U_k} \right].$$

For Stage II in Appendix B we derive the following lower bound $\Delta L(\Delta C)$ on the increase in the criterion L caused by a change ΔC in C :

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

where $\Sigma Q_i = \sum_{j \in \mathcal{N}(i)} Q_j(1)$ and $C_j = W_{i,j}$. The optimal change ΔC can be found by maximizing the lower bound via gradient ascent *separately* for each ΔC_j :

$$\frac{\partial \Delta C_j}{\partial t} = \frac{\partial \Delta L'}{\partial \Delta C_j} \quad (34)$$

and then using the optimal ΔC in the update rule $C \leftarrow C + \Delta C$.

6 Experimental Results

6.1 Comparing the Learned to the Manual Network Performance

We have learned our network’s weights using 50 images from the training set of the Berkeley segmentation benchmark [70,76], where natural images are provided together with manual segmentations. In Fig. 6 we compare the learned interconnection weights with those estimated using the ad-hoc choices in [23,24]; it is clear that ‘intuition’ quickly misled us to non-optimal connection patterns. For example, comparing the connections weights at Stage II, 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 that a pattern similar to that initially conjectured for Stage II has been learned for Stage I. Further, in Fig. 7 we demonstrate the effect of the training algorithm using one image that belongs to the training set. We observe that the network avoids taking sharp decisions and gives fewer false alarms at highly textured areas.

To quantitatively assess whether learning improves the performance of our network and to see where it stands compared to other works, we have used

Manual								
Learned								
	Stage I				Stage II			

Fig. 6. Difference between the manually set and the learned connection weights. The top row shows the manually set connection weights, with a bright value indicating a strong connection between a neuron located at the center of the figure and a neuron at that location. The bottom row shows the learned values for these. For each stage, the first two columns correspond to Horizontal-to-Horizontal and Vertical-to-Vertical connections at coarse scale and the following two to the same connections at a fine scale.

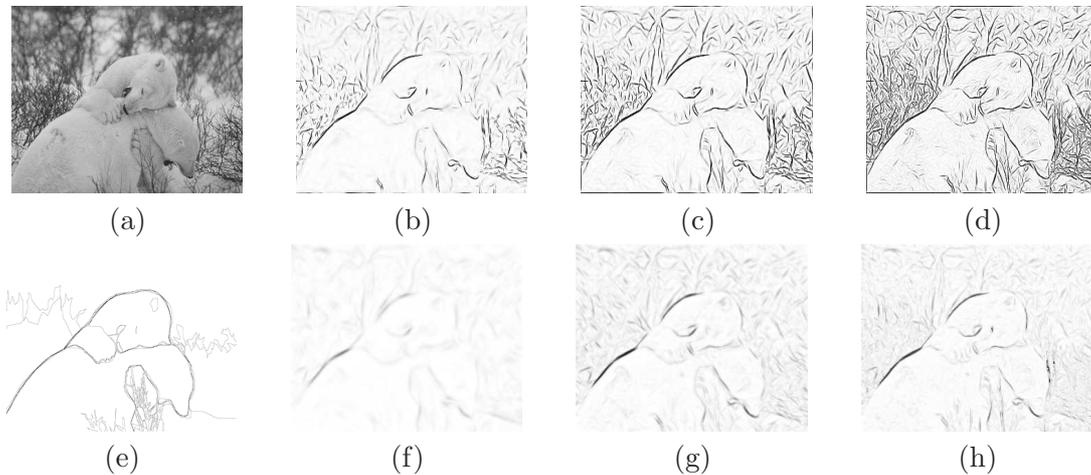


Fig. 7. Learning edge detection: (a) Input Image, (b)-(d) Probability of edge using manually set weights, at increasing scales. (e) Probability of edge based on Human Segmentations (f)-(h) Same as (b)-(d) using learned weights.

the Berkeley Segmentation benchmark ground truth.

6.2 Boundary Detection Benchmarking Results

The quantities used to describe a detector’s performance are its Precision, P and its Recall, R . Recall equals the ratio of correctly detected to actual edges and Precision the ratio of correctly detected to detected edges. 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. By modifying the detector’s threshold and plotting the values these quantities take we obtain a Precision-Recall curve based on which we can compare two detectors. When two Precision-Recall curves intersect, a useful measure for summarizing the performance of

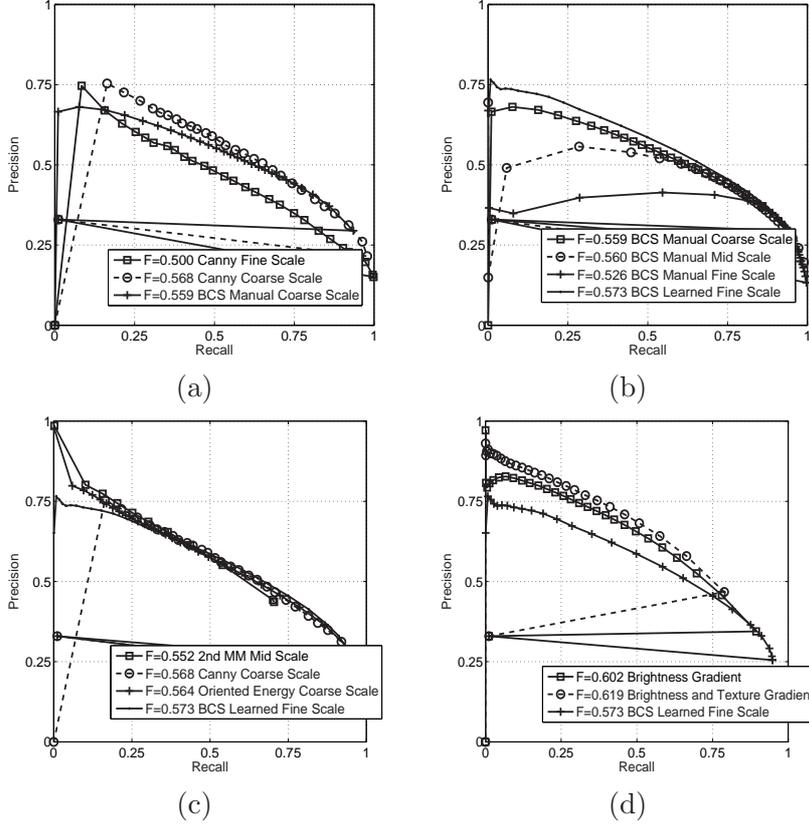


Fig. 8. Comparative results for edge detection: (a) Manually trained BCS results vs. Canny edge detection at two different scales. (b) BCS system using learned weights vs. BCS systems with manually set weights at multiple scales. (c) Comparison to Canny, Oriented Energy and Second Moment Matrix methods. (d) Comparison to Brightness/Texture Gradient. *Pls. cf. [77] for further results.*

the detector is its F -measure defined as:

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

where α is a weighting factor set to 0.5; larger values are more desirable and two detectors can be compared using their F measure.

To save space, we present here only the PR curves from the whole dataset, and provide results on individual images in the website [77]. In Fig. 8 we plot the 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 [4]. Shown in the legend is the maximal F -measure of each detector, valued from zero to one. The Precision and Recall values are obtained by aggregating the false positives and true matches of the boundary detector at each threshold value *over the whole set of images*.

From Fig. 8(a) we realize that the claim made in Sec. 3.5 that the original system outperforms Canny edge detection is partially true, and holds when using a small Gaussian function. For larger scales Canny’s method outperforms

the original system in the high precision regime.

As we show however in Fig. 8(b) learning the network weights systematically improves performance, and the learned system outperforms all manually-determined BCS systems. In Fig. 8(c) we observe that our system achieves a higher F-measure compared to the Canny, Oriented Energy and the Second Moment Matrix methods at their optimal scales. We attribute this to the incorporation of multi-scale cues with region-based, saliency information and learning the connection weights. Still, as shown in Fig. 8(d) our detector outperforms the Brightness or Brightness Texture Gradient methods of [4] only in the high recall regime, i.e. when edge ‘hallucination’ is useful.

In [78,4] the features and classifiers that practically proved to offer the best performance are used, using e.g. χ^2 tests and decision trees to classify a pixel as an edge. Such approaches do result in improved performance, but deviate from biological vision towards pattern recognition.

The Oriented Energy used as input to our system is shown by the comparison Fig. 8(c) to be outperformed by our system outputs. We therefore expect that if the more elaborate cues utilized e.g. in [4,79–81] were introduced in our system its ability to learn its recurrent connection weights from ground truth data would allow our system to surpass the performance of [4,81]; this is a direction that we intend to pursue in future research. In specific, in the Tensor Discriminant Analysis framework of [79,80] and the Boosting-Based approach of [81] the feature combination used for classification is *learned* discriminatively, which can further reduce human intervention in network design. Before concluding, we note that the improvement in performance has been largely due to the removal of textured edges via surround suppression and the enhancement of low-contrast edges via the saliency signal. The use of such mechanisms in conjunction with edge detection has previously been pursued in computer vision, cf. e.g. [42,61,82], and its potential has been demonstrated in these works. Still, what has been missing is a principled way to combine such cues and the bottom-up information provided by the filtering operators commonly used for edge detection. Our work fills this gap, and hopefully will help strengthen research in this direction.

7 Conclusion

In this paper we proposed a simple and efficient model for low- and mid- level vision tasks which compares favorably to classical computer vision algorithms, relying on biologically motivated mechanisms.

Apart from putting together the parts of the presented system, the 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. System performance has been demonstrated on the Berkeley segmentation benchmark, showing that our system outperforms standard edge detection algorithms, clearly validating the merits of a learning-based approach.

Concerning future extensions of this work, the main direction we consider is extending the current approach to other low- and mid- level computer vision tasks. Learning has dramatically reduced the human intervention required and we believe it will prove useful to both the biologically-motivated vision community and people working in classical low- and mid- level computer vision problems, like motion, stereo and image restoration. The cross-fertilization of learning and computer vision can weed out heuristics and ad-hoc choices, resulting in improved performance and better understanding of vision problems.

Acknowledgements

This research was supported by the European project IMAVIS - HPMT-CT-2000-00040 within the framework of the Marie Curie Fellowships - Training Sites programme and the Greek Ministry of Education Research Program HRAKLEITOS.

A Derivation of the Lyapunov Function

We validate that a system of neurons U, S following (16) and (11) decreases Eq. (20):

$$\mathcal{E} = \sum_i c_4(1-U)|\nabla_{\theta^\perp} S|^2 - U[c_1O + c_2T + c_3C] + a \int_{1/2}^U g^{-1}(u)du + \frac{d}{2}C(U)$$

We have defined $\nabla_{\theta^\perp} S = S' - S_i$, where S' is the surface process element closest to i along the line passing through i with angle θ^\perp . Therefore, using Eq. (11) we have:

$$\frac{\partial \mathcal{E}}{\partial S_i} = -2c_4 \sum_{\theta} (1-U_i) \nabla_{\theta^\perp} S^\theta, \quad \frac{dS_i}{dt} = \sum_{\theta} (1-U_i) \nabla_{\theta^\perp} S^\theta$$

Since $c_4 > 0$, for all i we have $\frac{\partial \mathcal{E}}{\partial S_i} \frac{dS_i}{dt} < 0$ so the evolution of S decreases \mathcal{E} . Concerning the line process, writing $cI = [c_1O + c_2T + c_3C + c_4|\nabla_{\theta^\perp} S|^2]$ we have

$$\frac{\partial \mathcal{E}}{\partial U_i} = aV - cI + d \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j \quad \text{since } \frac{\partial C(U)}{\partial U_i} \stackrel{W_{i,j}=W_{j,i}}{=} 2 \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j$$

By Eq. (9) and Eq. (16) we have for the MFR of neuron i

$$\frac{dU_i}{dt} = \frac{dg(V)}{dV} \frac{dV_i}{dt} = \alpha \left[-aV + cI - d \sum_{j \in \mathcal{N}(i)} W_{i,j} U_j \right]$$

where α is a positive quantity, given that g is an increasing function. We thereby conclude again that $\frac{\partial \mathcal{E}}{\partial U_i}$ and $\frac{dU_i}{dt}$ are of opposite sign for all i , so the evolution of the line process leads to the decrease of \mathcal{E} , as well.

B Derivation of the learning rules

For Stage I the steady state values of Eq. (5) are used in place of $Q_i(1)$ in Eq. (32); writing for simplicity $W_{i,j}U_j = C_jU_j$ we have:

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

For Stage II, we adapt to our case the Improved Iterative Scaling Algorithm [83]. Using the steady-state value of Eq. (16) as $Q_i(1)$ we can write:

$$L(C) = \sum_i P_i(1) \ln \frac{1}{1 + \exp(-\sum_{j \in \mathcal{N}(i)} C_j Q_j(1))} + P_i(0) \ln \frac{\exp(-\sum_{j \in \mathcal{N}(i)} C_j Q_j(1))}{1 + \exp(-\sum_{j \in \mathcal{N}(i)} C_j Q_j(1))} \quad (\text{B.1})$$

We now derive the lower bound on the change $\Delta L(\Delta C)$ in L caused by a change by ΔC in the connection weights. We have:

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

For the first equation we use the expression of $L(C)$ in Eq. (B.1), for the second basic properties of the \ln function and in the last step the expressions for $Q_i(1)$ $Q_i(0)$ in Eq. (B.1). A lower bound of J is derived as in [83], first by applying the inequality $-\ln(a) \geq 1 - a$ to the last term of J , which gives

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

The second bound is based on Jensen's inequality, $\exp(\sum p_i q_i) \leq \sum p_i \exp(q_i)$, if $\sum p_i = 1, p_i > 0$. We introduce $\Sigma Q_i = \sum_{j \in \mathcal{N}(i)} Q_j(1)$ and write:

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

We can now obtain the lower bound $\Delta L'$ used in Eq. (34):

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

References

- [1] J. Canny, A Computational Approach to Edge Detection, IEEE Trans. PAMI 8 (6) (1986) 679–698.
- [2] R. Deriche, Using Canny's Criteria to Derive a Recursively Implemented

- Optimal Edge Detector, *Int.'l Journal of Computer Vision* 1 (2) (1987) 167–187.
- [3] P. Perona, J. Malik, Detecting and Localizing Edges Composed of Steps, Peaks and Roofs, in: *Int.'l Conf. on Computer Vision*, 1990, pp. 52–57.
- [4] D. Martin, C. Fowlkes, J. Malik, Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues, *IEEE Trans. PAMI* 26 (5) (2004) 530–549.
- [5] S. Geman, D. Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Treatment of Images, *IEEE Trans. PAMI* 66 (6) (1984) 721–741.
- [6] P. Perona, J. Malik, Scale-Space and Edge Detection Using Anisotropic Diffusion, *IEEE Trans. PAMI* 12 (7) (1990) 629–639.
- [7] D. Mumford, J. Shah, Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems, *Communications on Pure and Applied Mathematics* 42 (5) (1989) 577–685.
- [8] J.-M. Morel, S. Solimini, *Variational Methods in Image Segmentation*, Birkhauser, Boston, 1995.
- [9] D. H. Hubel, T. N. Wiesel, Receptive Fields, Binocular Interaction, and Functional Architecture in the Cat's Visual Cortex, *J. Physiol. (London)* 160 (1962) 106–154.
- [10] D. H. Hubel, T. N. Wiesel, Receptive Fields and Functional Architecture of Monkey Striate Cortex, *J. Physiol.* 195 (1968) 215–243.
- [11] B. Wandell, *Foundations of Vision*, Sinauer Associates, 1995.
- [12] D. Marr, *Vision*, W.H. Freeman, 1982.
- [13] C. Koch, J. Marroquin, A. Yuille, Analog 'Neuronal' Networks in Early Vision, *Proc. Nat'l Academy of Sciences of USA* 83 (751) (1986) 4263–4267.
- [14] D. Mumford, *Neuronal Architectures for Pattern Theoretic Problems*, in: *Large Scale Theories of the Cortex*, MIT Press, 1994.
- [15] S. Grossberg, E. Mingolla, Neural Dynamics of Perceptual Grouping: Textures, Boundaries, and Emergent Segmentations, *Perception and Psychophysics* 38 (1985) 141–171.
- [16] S. Grossberg, E. Mingolla, Neural Dynamics of Form Perception: Boundary Completion, Illusory Figures, and Neon Color Spreading, *Psychological Review* 92 (1985) 173–211.
- [17] S. Grossberg, E. Mingolla, Neural Dynamics of Surface Perception: Boundary Webs, Illuminants, and Shape from Shading, *Computer Vision, Graphics, and Image Processing* 37 (1987) 116–165.
- [18] S. Grossberg, Cortical Dynamics of Three-Dimensional Form, Color, and Brightness Perception, I: Monocular Theory, *Perception and Psychophysics* 41 (1987) 87–116.
- [19] S. Grossberg, 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) 241–277.
- [20] S. Grossberg, *Neural Networks and Natural Intelligence*, MIT Press, 1988.

- [21] S. Grossberg, 3-D Vision and Figure-Ground Separation by Visual cortex, *Perception and Psychophysics* 55 (1994) 48–121.
- [22] S. Grossberg, E. Mingolla, J. Williamson, Synthetic Aperture Radar Processing by a Multiple Scale Neural System for Boundary and Surface Representation, *Neural Networks* 8 (7-8) (1995) 1005–1028.
- [23] I. Kokkinos, R. Deriche, P. Maragos, O. Faugeras, A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks, in: *ECCV*, 2004.
- [24] I. Kokkinos, R. Deriche, T. Papadopoulou, O. Faugeras, P. Maragos, Towards Bridging the Gap Between Computational and Biological Segmentation, Tech. rep., INRIA (2006).
- [25] J. Daugman, Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters, *J. Opt. Soc. Amer. (A)* 2 (7) (1985) 160–169.
- [26] E. Adelson, J. Bergen, Spatiotemporal Energy Models for the Perception of Motion, *J. Opt. Soc. Amer.* 2 (2) (1985) 284–299.
- [27] P. Dayan, L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, 2001.
- [28] M. Carandini, D. Heeger, Summation and Division by Neurons in Visual Cortex, *Science* 264 (1994) 1333–1336.
- [29] M. Carandini, D. Heeger, J. A. Movshon, Linearity and Normalization of Simple Cells of the Macaque Primary Visual Cortex, *Journal of Neuroscience* 17 (1997) 8621–8644.
- [30] R. Heydt, E. Peterhans, G. Baumgarthner, Illusory Contours and Cortical Neuron Responses, *Science* 224 (4654) (1984) 1260–1262.
- [31] D. Marcus, D. C. V. Essen, Scene Segmentation and Attention in Primate Cortical Areas V1 and V2, *J. Neurophysiology* 88 (2002) 2648–2658.
- [32] J. S. Bakin, K. Nakayama, C. D. Gilbert, Visual Responses in Monkey Areas V1 and V2 to Three-Dimensional Surface Configurations, *J. of Neuroscience* 20 (21) (2000) 8188–8198.
- [33] T. S. Lee, Computations in the Early Visual Cortex, *J. Physiology* 97 (203) (2003) 121–139.
- [34] C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 1999.
- [35] M. Cohen, S. Grossberg, Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks, *IEEE Trans. Systems, Man, and Cybernetics* 13 (5) (1983) 815–826.
- [36] E. P. Simoncelli, D. J. Heeger, A Model of Neuronal Responses in Visual Area MT, *Vision Research* 38 (5) (1998) 743–761.
- [37] J. J. Knierim, D. C. V. Essen, Neuronal Responses to Static Texture Patterns in Area V1 of the Alert Macaque Monkey, *Journal of Neurophysiology* 67 (4) (1992) 961–980.

- [38] H.-C. Nothdurft, J. L. Gallant, D. C. V. Essen, Response modulation by texture surround in primate area V1: Correlates of ‘popout’ under anesthesia, *Visual Neuroscience* 16 (1999) 15–34.
- [39] H. E. Jones, K. L. Grieve, W. Wang, A. M. Sillito, Surround suppression in primate v1, *Journal of Neurophysiology* 86 (2001) 2011–2028.
- [40] O. Schwartz, E. P. Simoncelli., Natural signal statistics and sensory gain control., *Nature Neuroscience* 4.
- [41] N. Petkov, M. A. Westenberg, Suppression of contour perception by band-limited noise and its relation to non-classical receptive field inhibition, *Biological Cybernetics* 88 (2003) 236–246.
- [42] C. Grigorescu, N. Petkov, M. A. Westenberg, Contour and boundary detection improved by surround suppression of texture edges, *Image and Vision Computing* 22 (8) (2004) 609–622.
- [43] W. Ross, Grossberg, S., E. Mingolla, Visual Cortical Mechanisms of Perceptual Grouping: Interacting Layers, Networks, Columns and Maps, *Neural Networks* 13 (6) (2000) 571–588.
- [44] S. Grossberg, *The Adaptive Brain*, Vol. 1, North-Holland, 1987.
- [45] T. S. Lee, A Bayesian Framework for Understanding Texture Segmentation in the Primary Visual Cortex, *Vision Research* 35 (18) (1995) 2643–2657.
- [46] T. S. Lee, D. Mumford, A. Yuille, Texture Segmentation by Minimizing Vector Valued Energy Functionals, in: *ECCV*, 1992.
- [47] Z. Li, Visual Segmentation by Contextual Influences via Intracortical Interactions in Primary Visual Cortex, *Network: Computation in Neural Systems* 10 (2) (1999) 187–212.
- [48] S. Zhu, T. Lee, A. Yuille, Region Competition: Unifying Snakes, Region Growing and Bayes/MDL for Multiband Image Segmentation, in: *Int.’l Conf. on Computer Vision*, 1995.
- [49] D. Groszof, R. Shapley, M. Hawken, Macaque V1 Neurons can Signal Illusory Contours, *Nature* 365 (6446) (1993) 550–552.
- [50] D. C. Ramachandran, V. S. Ramachandran, Psychophysical Evidence for Boundary and Surface Systems in Human Vision, *Vision Research* 38 (1) (1998) 71–77.
- [51] H. Neumann, W. Sepp, Recurrent V1-V2 Interaction in Early Visual Boundary Processing, *Biological Cybernetics* 91 (1999) 425–444.
- [52] F. Heitger, R. von der Heydt, A Computational Model Of Neural Contour Processing: Figure-Ground Segregation And Illusory Contours, in: *Int.’l Conf. on Computer Vision*, 1993, pp. 32–40.
- [53] C. Morrone, D. Burr, Feature Detection in Human Vision: a Phase-Dependent Energy Model, *Proceedings of the Royal Society of London B* 235 (1988) 221–245.
- [54] L. Pessoa, E. Mingolla, H. Neumann, A Contrast and Luminance-driven Multiscale Network Model of Brightness Perception, *Vision Research* 35 (15) (1995) 2201–2223.

- [55] A. Yuille, D. Geiger, A Common Framework for Image Segmentation, *Int.'l Journal of Computer Vision* 6 (1991) 227–243.
- [56] M. Black, G. Sapiro, D. Marrimont, D. Heeger, Robust Anisotropic Diffusion, *IEEE Trans. Im. Proc.* 7 (3) (1998) 421–432.
- [57] X. Ren, J. Malik, A Probabilistic Multi-scale Model for Contour Completion Based on Image Statistics, in: *ECCV*, Vol. 1, 2002, pp. 312–327.
- [58] L. Williams, D. Jacobs, Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Saliency, *Neural Computation* 9 (4) (1997) 837–858.
- [59] A. Sha'ashua, S. Ullman, Structural Saliency: the Detection of Globally Salient Structures Using a Locally Connected Network, in: *Int.'l Conf. on Computer Vision*, 1988, pp. 321–327.
- [60] G. Guy, G. Medioni, Inferring Global Perceptual Contours from Local Features, *Int.'l Journal of Computer Vision* 20 (1) (1996) 113–133.
- [61] P. Parent, S. Zucker, Trace Inference, Curvature Consistency, And Curve Detection, *IEEE Trans. PAMI* 11 (8) (1989) 823–839.
- [62] D. Mumford, *Elastica and Computer Vision*, in: B. J. (Ed.), *Algebraic Geometry and its applications*, Springer Verlag, 1993, pp. 507–518.
- [63] A. Yuille, Energy Functions for Early Vision and Analog Networks, Tech. Rep. 987, MIT A.I. Lab, <ftp://publications.ai.mit.edu/ai-publications/500-999/AIM-987.ps> (1987).
- [64] J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat.l Academy of Sciences of USA* 79 (1982) 2554–2558.
- [65] J. Hopfield, Neurons with Graded Response have Collective Computational Properties like those of Two-State Neurons, *Proc. Nat.l Academy of Sciences of USA* 81 (1984) 3088 – 3092.
- [66] J. Hertz, A. Krogh, R. G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley, 1989.
- [67] M. Proesmans, E. Pauwels, L. v. Gool, Coupled Geometry Driven Diffusion Equations for Low-Level Vision, in: *Geometry Driven Diffusion in Computer Vision*, Kluwer Academic, 1994.
- [68] W. Freeman, E. Pasztor, O. T. Carmichael, Learning Low-Level Vision, *Int.'l Journal of Computer Vision* 40 (1) (2000) 25–47.
- [69] S. Roth, M. J. Black, Fields of Experts: A Framework for Learning Image Priors, in: *IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
- [70] D. Martin, C. Fowlkes, D. Tal, J. Malik, A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, in: *Int.'l Conf. on Computer Vision*, 2001.
- [71] D. Ackely, G. Hinton, T. Sejnowski, A Learning Algorithm for Boltzmann Machines, *Cognitive Science* 9 (1985) 147–169.
- [72] G. Hinton, T. Sejnowski, Learning and Relearning in Boltzmann Machines, in: Rumelhart, McClelland (Eds.), *Parallel Distributed Processing*, Vol. 1, MIT Press, 1986, Ch. 7.

- [73] H. J. Kappen, Deterministic Learning Rules for Boltzmann Machines, *Neural Networks* 8 (1995) 537–548.
- [74] M. Welling, G. E. Hinton, A New Learning Algorithm for Mean Field Boltzmann Machines, Tech. Rep. 2001-002, Gatsby Computational Neuroscience Unit (2001).
- [75] T. Jaakkola, Tutorial on Variational Approximation Methods, in: *Advanced Mean Field Methods: Theory and Practice*, MIT Press, 2000.
- [76] The Berkeley Segmentation Dataset and Benchmark, "<http://www.cs.berkeley.edu/projects/vision/grouping/segbench/>".
- [77] Berkeley benchmark results, <http://www.stat.ucla.edu/~jkkokkin/html/algorithms.html>.
- [78] S. Konishi, A. L. Yuille, J. M. Coughlan, S. C. Zhu, Statistical Edge Detection: Learning and Evaluating Edge Cues, *IEEE Trans. PAMI* 25 (1) (2003) 57–74.
- [79] D. Tao, X. Li, X. Wu, S. Maybank, General Tensor Discriminant Analysis and Gabor Features for Gait Recognition, *IEEE Trans. PAMI* 29 (10) (2007) 1700–1715.
- [80] D. Tao, X. Li, X. Wu, W. Hu, S. Maybank, Supervised Tensor Learning, *Knowledge and Information Systems* 13 (1) (2007) 1–42.
- [81] P. Dollar, Z. Tu, S. Belongie, Supervised learning of edges and object boundaries, in: *IEEE Conf. Computer Vision and Pattern Recognition*, 2006.
- [82] J. August, S. W. Zucker, Sketches with curvature: The curve indicator random field and markov processes, *IEEE Trans. PAMI* 25 (4) (2003) 387–400.
- [83] A. Berger and S. D. Pietra and V. D. Pietra, A Maximum Entropy Approach to Natural Language Processing, *Computational Linguistics* 22 (1) (1996) 39–71.