Biological Cybernetics manuscript No.
(will be inserted by the editor)

A Rotation and Translation Invariant Discrete Saliency Network

Lance R. Williams[1], **John W. Zweck**[2]

[1] Dept. of Computer Science, Univ. of New Mexico, Albuquerque, NM 87110, USA
[2] Dept. of CS and EE, Univ. of Maryland, Baltimore County, MD 21250, USA

21 Oct 2002

Abstract We describe a neural network which enhances and completes salient closed contours in images. Our work is different from all previous work in three important ways. First, like the input provided to primary visual cortex (V1) by the lateral geniculate nucleus (LGN), the input to our computation is isotropic. That is, it is composed of spots not edges. Second, our network computes a well defined function of the input based on a distribution of closed contours characterized by a random process. Third, even though our computation is implemented in a discrete network, its output is invariant to continuous rotations and translations of the input image.

1 Introduction

Vision, whether that of man or machine, must solve certain fundamental information processing problems. One such problem is robustness to rotations and translations of the input image. Ideally, the output of a visual computation should vary continuously under continuous transformation of its input. We believe that understanding how the human visual system achieves this invariance will lead to significant improvements in the performance of machine vision systems. Conversely, studying the fundamental information processing problems common to both human and machine vision can lead to a deeper understanding of the structure and function of the human visual cortex.

There is a long history of research on neural networks inspired by the structure of visual cortex whose functions have been described as contour completion, saliency enhancement, orientation sharpening, or segmentation, *e.g.*, August (2000), Li (1998), Yen and Finkel (1996), Guy and Medioni (1996), Iverson (1993), Heitger and von der Heydt (1993), Parent and Zucker (1989), Sashua and Ullman (1988), and Grossberg and Mingolla (1985). In this paper, we describe a neural network which enhances and completes salient closed contours in images. Our work is different from all previous work in three important ways. First, like the input provided to primary visual cortex (V1) by the lateral geniculate nucleus (LGN), the input to our computation is isotropic. That is, the input is composed of spots not edges.[1] Second, our network computes a well defined function of the input based on a distribution of closed contours characterized by a random process. Third, even though our computation is implemented in a discrete network, its output is invariant to continuous rotations and translations of the input image.[2]

There are two important properties which a computation must possess if it is to be invariant to rotations and translations, *i.e.*, Euclidean invariant. First, the input, the output, and all intermediate representations must be Euclidean invariant. Second, all transformations of these representations must also be Euclidean invariant. Previous models are not Euclidean invariant, first and foremost, because their input representations are not Euclidean invariant. That is, not all rotations and translations of the input can be represented equally well. This problem is often skirted by choosing input patterns which match particular choices of sampling rate and phase. For example, Li (1998) used only six samples in orientation (including 0°) and Heitger and von der Heydt (1993) only twelve (including 0°, 60° and 120°). Li's first test pattern was a dashed line of orientation, 0°, while Heitger and von der Heydt (1993) used a Kanizsa Triangle with sides of 0°, 60°, and 120° orientation.[3] There is no reason to believe that the experimental results shown in these papers would be similar if the input images were rotated by as little as 5°. To our knowledge, no one has ever commented on this problem before.

[1] Like our model, Guy and Medioni (1996)'s can also be applied to input patterns consisting of spots.

[2] Guy and Medioni (1996) represent a distribution on $\mathbf{R}^2 \times S^1$ as an $N \times N$ array of 2×2 scatter matrices. Ignoring the issue of the biological plausibility of such an encoding scheme, we observe that although the representation of S^1 is clearly continuous, the rectangular grid which is used to sample $\mathbf{R}^2$ is not. It follows that their computation is not Euclidean invariant.

[3] Nor are we blameless in this regard. Williams and Jacobs (1997) used 36 discrete orientations including 0°, 60°, and 120°, and also show results on a conveniently oriented Kanizsa Triangle.

2 A continuum formulation of the saliency problem

The following section reviews the continuum formulation of the contour completion and saliency problem as described in Williams and Thornber (2001).

2.1 Shape distribution

Mumford (1994) observed that the probability distribution of object boundary shapes could be modeled by a *Fokker-Planck* equation of the following form,

$$\frac{\partial p}{\partial t} = -\cos\theta\frac{\partial p}{\partial x} - \sin\theta\frac{\partial p}{\partial y} + \frac{\sigma^2}{2}\frac{\partial^2 p}{\partial\theta^2} - \frac{1}{\tau}p, \quad (1)$$

where $p(\mathbf{x},\theta;t)$ is the probability that a particle is located at position, $\mathbf{x} = (x,y)$, and is moving in direction, θ, at time, t. This partial differential equation can be viewed as a set of independent *advection* equations in x and y (the first and second terms) coupled in the θ dimension by the *diffusion* equation (the third term). The advection equations translate probability mass in direction, θ, with unit speed, while the diffusion term implements a Brownian motion in direction, with *diffusion parameter*, σ. The combined effect of these three terms is that particles tend to travel in straight lines, but over time they drift to the left or right by an amount proportional to σ^2. Finally, the effect of the fourth term is that particles decay over time, with a half-life given by the decay constant, τ.

2.2 The propagators

The Green's function, $G(\mathbf{x},\theta;t_1 \,|\, \mathbf{u},\phi;t_0)$, gives the probability that a particle observed at position, $\mathbf{u}$, and direction, ϕ, at time, t_0, will later be observed at position, $\mathbf{x}$, and direction, θ, at time, t_1. It is the solution, $p(\mathbf{x},\theta;t_1)$, of the Fokker-Planck initial value problem with initial value, $p(\mathbf{x},\theta;t_0) = \delta(\mathbf{x}-\mathbf{u})\delta(\theta-\phi)$ where δ is the Dirac delta function. Unlike Williams and Thornber (2001), who assumed that the location of input edges could be specified with arbitrary precision, we explicitly model the limited spatial acuity of the edge detection process. We consider two edges to be indistinguishable if they are separated by a distance smaller than some scale, Δ. The Green's function is used to define two *propagators*. The long-time propagator:

$$P_0(\mathbf{x},\theta \mid \mathbf{u},\phi) = \int_0^\infty dt\, \chi(t)\, G(\mathbf{x},\theta;t \,|\, \mathbf{u},\phi;0) \quad (2)$$

gives the probability that $(\mathbf{x},\theta)$ and $(\mathbf{u},\phi)$ are distinguishable edges from the boundary of a single object. The short-time propagator:

$$P_1(\mathbf{x},\theta \mid \mathbf{u},\phi) = \int_0^\infty dt\, [1-\chi(t)]\, G(\mathbf{x},\theta;t \,|\, \mathbf{u},\phi;0) \quad (3)$$

gives the probability that $(\mathbf{x},\theta)$ and $(\mathbf{u},\phi)$ are from the boundary of a single object, but are indistinguishable. In both of these propagators, $\chi(.)$ is a cut-off function with $\chi(0) = 0$ and $\lim_{t\to\infty}\chi(t) = 1$:

$$\chi(t) \;=\; \tfrac{1}{2}\left[1 + \tfrac{2}{\pi}\,\mathrm{atan}\left(\mu\left[\tfrac{t}{\Delta} - \alpha\right]\right)\right]. \quad (4)$$

The cut-off function is characterized by three parameters, α, μ, and Δ. The parameter, Δ, is the scale of the edge detection process. Relative to this scale, the parameter, α, specifies the location of the cut-off and μ specifies its hardness.

2.3 Eigenfunctions

We use the long-time propagator, P_0, to define an integral linear operator, $Q(.)$, which combines three sources of information: (1) the probability that two edges belong to the same object; (2) the probability that the two edges are distinguishable; and (3) the probability that the two edges exist. It is defined as follows:

$$Q(\mathbf{x},\theta \,|\, \mathbf{u},\phi) = b(\mathbf{x})^{\frac{1}{2}} P_0(\mathbf{x},\theta \mid \mathbf{u},\phi) b(\mathbf{u})^{\frac{1}{2}} \quad (5)$$

where the *input bias function*, $b(\mathbf{x})$, gives the probability that an edge exists at $\mathbf{x}$. Note that, as defined, $b(\mathbf{x})$ depends only on the position, $\mathbf{x}$, and not the direction, θ, of the edge. This reflects the fact that the input to the computation is isotropic.[4]

As described in Williams and Thornber (2001), the right and left eigenfunctions, $s(.)$ and $\bar{s}(.)$, of $Q(.)$ with largest positive real eigenvalue, λ, play a central role in the computation of saliency:

$$\lambda s(\mathbf{x},\theta) \;=\; \int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{u}d\phi\; Q(\mathbf{x},\theta \,|\, \mathbf{u},\phi)s(\mathbf{u},\phi) \quad (6)$$

$$\lambda \bar{s}(\mathbf{x},\theta) \;=\; \int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{u}d\phi\; \bar{s}(\mathbf{u},\phi)Q(\mathbf{u},\phi \,|\, \mathbf{x},\theta). \quad (7)$$

Because $Q(.)$ is invariant under the transformation

$$Q(\mathbf{x},\theta \,|\, \mathbf{u},\phi) = Q(\mathbf{u},\phi+\pi \,|\, \mathbf{x},\theta+\pi), \quad (8)$$

which reverses the order and direction of its arguments, the right and left eigenfunctions are related by

$$\bar{s}(\mathbf{x},\theta) = s(\mathbf{x},\theta+\pi). \quad (9)$$

2.4 Stochastic completion field

The fact that the distribution of closed contours, or *stochastic completion field*, can be factored into a *source field* derived from $s(.)$ and a *sink field* derived from $\bar{s}(.)$ is discussed in detail in Williams and Thornber (2001). The stochastic completion field, $c(\mathbf{u},\phi)$, gives the probability that a closed contour containing any subset of the edges exists at $(\mathbf{u},\phi)$. It is the sum of three terms:

$$c(\mathbf{u},\phi) = \frac{p_0(\mathbf{u},\phi)\bar{p}_0(\mathbf{u},\phi)+p_0(\mathbf{u},\phi)\bar{p}_1(\mathbf{u},\phi)+p_1(\mathbf{u},\phi)\bar{p}_0(\mathbf{u},\phi)}{\lambda\int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{x}d\theta\; s(\mathbf{x},\theta)\bar{s}(\mathbf{x},\theta)} \quad (10)$$

where $p_m(\mathbf{u},\phi)$ is a source field, and $\bar{p}_m(\mathbf{u},\phi)$ is a sink field:

$$p_m(\mathbf{u},\phi) \;=\; \int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{x}d\theta\; P_m(\mathbf{u},\phi \mid \mathbf{x},\theta)b(\mathbf{x})^{\frac{1}{2}}s(\mathbf{x},\theta)$$

$$\bar{p}_m(\mathbf{u},\phi) \;=\; \int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{x}d\theta\; \bar{s}(\mathbf{x},\theta)b(\mathbf{x})^{\frac{1}{2}}P_m(\mathbf{x},\theta \mid \mathbf{u},\phi).$$

[4] The model could easily be extended to include non-isotropic input by making $b(.)$ a function of both $\mathbf{x}$ and θ. This would allow us to model any orientation bias present in feedforward connections from LGN to V1.

The purpose of writing $c(\mathbf{u},\phi)$ in this way is to remove the contribution, $p_1(\mathbf{u},\phi)\bar{p}_1(\mathbf{u},\phi)$, of closed contours at scales smaller than Δ which would otherwise dominate the completion field. Given the above expression for the completion field, it is clear that the key problem is computing the eigenfunction, $s(.)$, of $Q(.)$ with largest positive real eigenvalue. To accomplish this, we can use the well known power method. See Golub and Van Loan (1996). In this case, the power method involves repeated application of the linear operator, $Q(.)$, to the function, $s(.)$, followed by normalization:

$$s^{(m+1)}(\mathbf{x},\theta) = \frac{\int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{u}d\phi\, Q(\mathbf{x},\theta|\mathbf{u},\phi)s^{(m)}(\mathbf{u},\phi)}{\int\int\int_{\mathbf{R}^2\times S^1}\int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{x}d\theta d\mathbf{u}d\phi\, Q(\mathbf{x},\theta|\mathbf{u},\phi)s^{(m)}(\mathbf{u},\phi)} \quad (11)$$

In the limit, as m gets very large, $s^{(m+1)}(\mathbf{x},\theta)$ converges to the eigenfunction of $Q(.)$, with largest positive real eigenvalue.[5] We observe that the above computation can be considered a continuous state, discrete time, recurrent neural network.

3 A neural implementation

The fixed-point of the neural network we describe is the eigenfunction with largest positive real eigenvalue of the linear integral operator, $Q(.)$, formed by composing the input independent linear operator, $P_0(.)$, and the input dependent linear operator, $b(.)$. The dynamics of the neural network are derived from the update equation for the standard power method for computing eigenvectors. It is useful to draw an analogy between our neural network for contour completion and the models for the emergence of orientation selectivity in primary visual cortex described by Somers *et al.* (1995) and Ben-Yishai *et al.* (1995). See Fig. 1. First, we can identify $s(.)$ with simple cells in V1 and the input bias function, $b(.)$, which modulates $s(.)$ in the numerator of the update equation, with the feedforward excitatory connections from the LGN. Second, we can identify $P_0(.)$ with the intra-cortical excitatory connections which Somers *et al.* hypothesize are primarily responsible for the emergence of orientation selectivity in V1. As in the model of Somers *et al.*, these connections are highly specific and mainly target cells of similar orientation preference. Third, we identify the denominator of the update equation with the orientation non-specific intra-cortical inhibitory connections which Somers *et al.* hypothesize keep the level of activity within bounds. Finally, we identify $c(.)$, the stochastic completion field, with the population of cells in V2 described by von der Heydt *et al.* (1984).

[5] It is only in the limit of large m, *i.e.*, after the power method iteration has converged, that $c(\mathbf{x},\theta)$ represents a distribution of **closed** contours through location $(\mathbf{x},\theta)$. Prior to convergence, $s^{(m)}(\mathbf{x},\theta)\bar{s}^{(m)}(\mathbf{x},\theta)$ can be considered to be a distribution of non-closed contours of length $2m+1$ edges centered at location $(\mathbf{x},\theta)$.

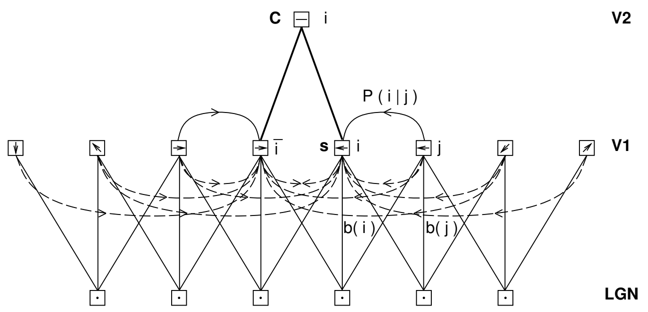

Fig. 1 Thin solid lines indicate feedforward connections from LGN which provide a weak orientation bias, *i.e.*, $b(.)$, to simple cells in V1, *i.e.*, $s(.)$. Solid lines with arrows indicate orientation specific intra-cortical excitatory connections, *i.e.*, $P_0(.)$. Dashed lines with arrows indicate orientation non-specific intra-cortical inhibitory connections. Thick solid lines indicate feedforward connections between V1 and V2, *i.e.*, $c(.)$.

4 A discrete implementation of the continuum formulation

Following Zweck and Williams (2000), the continuous functions comprising the state of the computation are represented as weighted sums of a finite set of *shiftable-twistable* basis functions. The weights form the coefficient vectors for the functions. The computation we describe is biologically plausible in the sense that all transformations of state are effected by linear transformations (or other vector parallel operations) on the coefficient vectors.

4.1 Shiftable-twistable bases

The input and output of the above computation are functions defined on the continuous space, $\mathbf{R}^2\times S^1$, of positions in the plane, $\mathbf{R}^2$, and directions in the circle, S^1. For such computations, the important symmetry is determined by those transformations, $T_{\mathbf{x}_0,\theta_0}$, of $\mathbf{R}^2\times S^1$, which perform a shift in $\mathbf{R}^2$ by $\mathbf{x}_0$, followed by a twist in $\mathbf{R}^2\times S^1$ through an angle, θ_0. A *twist* through an angle, θ_0, consists of two parts: (1) a rotation, R_{θ_0}, of $\mathbf{R}^2$ and (2) a translation in S^1, both by θ_0. The symmetry, $T_{\mathbf{x}_0,\theta_0}$, which is called a *shift-twist transformation*, is given by the formula,

$$T_{(\mathbf{x}_0,\theta_0)}(\mathbf{x},\theta) = (R_{\theta_0}(\mathbf{x}-\mathbf{x}_0)\,,\,\theta-\theta_0). \quad (12)$$

The relationship between continuous shift-twist transformations and computations in primary visual cortex has been hypothesized by Williams and Jacobs (1997) and by Bressloff *et al.* (2001). The continuous shift-twist group characterizes the symmetry of the Green's function of the Fokker-Planck equation described by Mumford (1994):

$$G(\mathbf{x},\theta\,;t_1\,|\,\mathbf{u},\phi\,;t_0) = G(R_\phi(\mathbf{x}-\mathbf{u}),\theta-\phi\,;t_1-t_0\,|\,0,0\,;0). \quad (13)$$

The **continuous** shift-twist symmetry of the Green's function can be understood by considering the analogous **discrete** symmetry of highest order. An idealized model of V1 might be based on a set of hypercolumns arranged in a hex lattice and with six discrete orientation preferences (see Fig. 2). By virtue of the shift-twist symmetry of the interconnection matrix, the long range, orientation specific, intra-cortical excitatory connections labeled A and B have equal weight. However, because this lattice is not invariant under continuous shift-twist transformations, the output of such a model (unlike ours) would not be Euclidean invariant. A visual computation, C, on $\mathbf{R}^2 \times S^1$ is called *shift-twist invariant* if, for **all** $(\mathbf{x}_0, \theta_0) \in \mathbf{R}^2 \times S^1$, a shift-twist of the input by $(\mathbf{x}_0, \theta_0)$ produces an identical shift-twist of the output. This property can be depicted in the following commutative diagram,

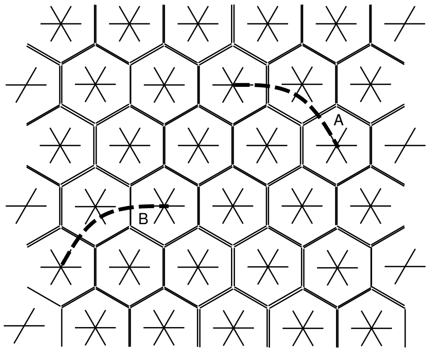

Fig. 2 The **continuous** shift-twist symmetry of the Green's function can be understood by considering the analogous **discrete** symmetry of highest order.

$$
\begin{array}{ccc}
b(\mathbf{x},\theta) & \stackrel{C}{\rightarrow} & c(\mathbf{x},\theta) \\
\downarrow T_{\mathbf{x}_0,\theta_0} & & \downarrow T_{\mathbf{x}_0,\theta_0} \\
b(R_{\theta_0}(\mathbf{x}-\mathbf{x}_0),\theta-\theta_0) & \stackrel{C}{\rightarrow} & c(R_{\theta_0}(\mathbf{x}-\mathbf{x}_0),\theta-\theta_0),
\end{array}
$$

where $b(.)$ is the input, $c(.)$, is the output, $\stackrel{C}{\rightarrow}$ is the computation, and $\stackrel{T_{\mathbf{x}_0,\theta_0}}{\rightarrow}$ is the shift-twist transformation. Correspondingly, we define a *shiftable-twistable basis*[6] of functions on $\mathbf{R}^2 \times S^1$ to be a set of functions on $\mathbf{R}^2 \times S^1$ with the property that whenever a function, $f(\mathbf{x},\theta)$, is in their span, then so is $f(T_{\mathbf{x}_0,\theta_0}(\mathbf{x},\theta))$, for every choice of $(\mathbf{x}_0,\theta_0)$ in $\mathbf{R}^2 \times S^1$. As such, the notion of a shiftable-twistable basis on $\mathbf{R}^2 \times S^1$ generalizes that of a shiftable-steerable basis on $\mathbf{R}^2$ introduced by Freeman and Adelson (1991) and Simoncelli *et al.* (1992).

Shiftable-twistable bases can be constructed as follows. Let $\Psi(\mathbf{x},\theta)$ be a function on $\mathbf{R}^2 \times S^1$ which is periodic (with period X) in both spatial variables, $\mathbf{x}$. In analogy with the definition of a shiftable-steerable function on $\mathbf{R}^2$, we say that Ψ is *shiftable-twistable* on $\mathbf{R}^2 \times S^1$ if there are integers, K and M, and *interpolation functions*, $a_{\mathbf{k},m}(\mathbf{x}_0,\theta_0)$, such that for each $(\mathbf{x}_0,\theta_0) \in \mathbf{R}^2 \times S^1$, the shift-twist of Ψ by $(\mathbf{x}_0,\theta_0)$ is a linear combination of a finite number of basic shift-twists of Ψ by amounts $(\mathbf{k}\Delta, m\Delta_\theta)$, *i.e.*, if

$$\Psi(T_{\mathbf{x}_0,\theta_0}(\mathbf{x},\theta)) = \textstyle\sum_{\mathbf{k},m} a_{\mathbf{k},m}(\mathbf{x}_0,\theta_0)\Psi(T_{\mathbf{k}\Delta,m\Delta_\theta}(\mathbf{x},\theta)). \quad (14)$$

Here $\Delta = X/K$ is the *basic shift amount* and $\Delta_\theta = 2\pi/M$ is the *basic twist amount*. The sum in equation (14) is taken over all pairs of integers, $\mathbf{k} = (k_x, k_y)$, in the range, $0 \le k_x, k_y < K$, and all integers, m, in the range, $0 \le m < M$. As in Simoncelli *et al.* (1993) the interpolation functions, $a_{\mathbf{k},m}$, are sinc functions.

The Gaussian-Fourier basis is the product of a shiftable-steerable basis of Gaussians in $\mathbf{x}$ and a Fourier series basis in θ. For the experiments in this paper, the standard deviation of the Gaussian basis function, $g(\mathbf{x}) = \frac{1}{\Delta} e^{-\|\mathbf{x}\|^2/2\Delta^2}$, equals the basic shift amount, Δ. We regard $g(\mathbf{x})$ as a periodic function of period, X, which is chosen to be much larger than Δ, so that $g(X/2, X/2)$ and its derivatives are essentially zero. For each frequency, ω, and shift amount, Δ (where $K = X/\Delta$ is an integer), we define the *Gaussian-Fourier basis functions*, $\Psi_{\mathbf{k},\omega}$, by

$$\Psi_{\mathbf{k},\omega}(\mathbf{x},\theta) = g(\mathbf{x} - \mathbf{k}\Delta)\, e^{i\omega\theta}. \quad (15)$$

Zweck and Williams (2000) showed that the Gaussian-Fourier basis is shiftable-twistable.

4.2 Power method update formula

Suppose that $Q(.)$'s approximate eigenfunction, $s^{(m)}(\mathbf{x},\theta)$, can be represented in the Gaussian-Fourier basis as

$$s^{(m)}(\mathbf{x},\theta) = \textstyle\sum_{\mathbf{k},\omega} s^{(m)}_{\mathbf{k},\omega} \Psi_{\mathbf{k},\omega}(\mathbf{x},\theta). \quad (16)$$

The vector, $\mathbf{s}^{(m)}$, with components, $s^{(m)}_{\mathbf{k},\omega}$, will be called the *coefficient vector* of $s^{(m)}(\mathbf{x},\theta)$. In the next two sections, we demonstrate how the following integral linear transform:

$$s^{(m+1)}(\mathbf{x},\theta) = \\ \textstyle\int\int\int_{\mathbf{R}^2\times S^1} d\mathbf{u}d\phi\, P_0(\mathbf{x},\theta \mid \mathbf{u},\phi) b(\mathbf{u}) s^{(m)}(\mathbf{u},\phi) \quad (17)$$

(*i.e.*, the basic step in the power method) can be implemented as a discrete linear transform in a Gaussian-Fourier shiftable-twistable basis:

$$\mathbf{s}^{(m+1)} = \mathbf{P}\mathbf{B}\mathbf{s}^{(m)}. \quad (18)$$

4.3 The propagation operator P

In practice, we do not explicitly represent the matrix, $\mathbf{P}$. This is because it is quite large and explicitly computing the above matrix-vector product using it would be prohibitively expensive. Instead we compute the matrix-vector product using the advection-diffusion-decay operator in the Gaussian-Fourier shiftable-twistable basis, $\mathbf{A} \circ \mathbf{D}$, which is derived and described in detail in Zweck and Williams (2000). The coefficient vector, $\mathbf{s}^{(m+1)}$, is a weighted average of solutions, $\mathbf{q}^{(m,n)}$, of a

[6] We use this terminology even though the basis functions need not be linearly independent.

Fokker-Planck initial value problem. The average is over all positive times, $t = n\Delta t$. That is,

$$\mathbf{s}^{(m+1)} = \sum_{n=0}^{\infty} \chi(n\Delta t)\mathbf{q}^{(m,n)} \tag{19}$$

where $\chi(.)$ is the cutoff function and the initial value for $\mathbf{q}$ is the product of the bias operator, $\mathbf{B}$, and the coefficient vector, $\mathbf{s}^{(m)}$,

$$\mathbf{q}^{(m,0)} = \mathbf{B}\mathbf{s}^{(m)}. \tag{20}$$

For convenience, we introduce a variable, $\mathbf{p}^{(m,n)}$, to represent the partial sum of $\mathbf{q}^{(m,n)}$ from $t = 0$ to $t = n\Delta t$. This leads to the following update equations:

$$\mathbf{q}^{(m,n+1)} = (\mathbf{A} \circ \mathbf{D})\,\mathbf{q}^{(m,n)} \tag{21}$$

$$\mathbf{p}^{(m,n+1)} = \mathbf{p}^{(m,n)} + \chi(n\Delta t)\,\mathbf{q}^{(m,n+1)}, \tag{22}$$

where $\mathbf{A}$ is the the advection operator and $\mathbf{D}$ is the diffusion decay operator. In the shiftable-twistable basis, the advection operator, $\mathbf{A}$, is a discrete convolution:

$$q_{\ell,\eta}^{(m,n+\frac{1}{2})} = \sum_{\mathbf{k},\omega}\ \hat{a}_{\ell-\mathbf{k},\eta-\omega}(\Delta t)\ q_{\mathbf{k},\omega}^{(m,n)} \tag{23}$$

with the kernel,

$$\hat{a}_{\mathbf{k},\eta}(\Delta t) = \frac{1}{2\pi}\int_0^{2\pi} a_{\mathbf{k}}(\Delta t[\cos\theta, \sin\theta]^{\mathrm{T}})\ \exp(-i\eta\theta)\,d\theta \tag{24}$$

where the $a_{\mathbf{k}}$ are interpolation functions. Let N be the number of Fourier series frequencies, ω, used in the shiftable-twistable basis, and let $\Delta\theta = 2\pi/N$. The diffusion-decay operator, $\mathbf{D}$, is given by the diagonal matrix-vector product,

$$q_{\mathbf{k},\omega}^{(m,n+1)} = e^{-\Delta t/\tau}(\lambda e^{-i\omega\Delta\theta} + (1-2\lambda) + \lambda e^{i\omega\Delta\theta})\ q_{\mathbf{k},\omega}^{(m,n+\frac{1}{2})}, \tag{25}$$

where $\lambda = \frac{\sigma^2}{2}\frac{\Delta t}{(\Delta\theta)^2}$.

4.4 The bias operator B

In the continuum, the bias operator multiplies the function, $s(.)$, by the input bias function, $b(.)$. Our aim is to identify an equivalent linear operator in the shiftable-twistable basis. Suppose that both $s(.)$ and $b(.)$ are represented in a Gaussian basis, $g_{\mathbf{k}}(\mathbf{x}) = g(\mathbf{x} - \mathbf{k}\Delta)$. Their product is:

$$b(\mathbf{x})s(\mathbf{x}) = \sum_{\mathbf{k}} s_{\mathbf{k}}\, g_{\mathbf{k}}(\mathbf{x})\ \cdot\ \sum_{\ell} b_{\ell}\, g_{\ell}\,(\mathbf{x}) \tag{26}$$

$$= \sum_{\mathbf{k}}\sum_{\ell} s_{\mathbf{k}} b_{\ell}\, g_{\mathbf{k}}(\mathbf{x}) g_{\ell}\,(\mathbf{x}). \tag{27}$$

Now, the product of two Gaussian basis functions, $g_{\mathbf{k}}$ and g_{ℓ}, is a Gaussian of smaller variance (with higher-frequency content) which cannot be represented in the Gaussian basis, $g_{\mathbf{k}}$. Because $b(\mathbf{x})s(\mathbf{x})$ is a linear combination of the products of pairs of Gaussian basis functions, it cannot be represented in the Gaussian basis either. However, we observe that the convolution of $b(\mathbf{x})s(\mathbf{x})$ and a Gaussian, $h(\mathbf{x}) * [b(\mathbf{x})\ s(\mathbf{x})]$, where

$$h(\mathbf{x}) = \frac{1}{\Delta^2\pi} e^{-\|\mathbf{x}\|^2/\Delta^2} \tag{28}$$

can be represented in the Gaussian basis. It follows that there exists a matrix, $\mathbf{B}$, such that

$$h(\mathbf{x}) * [b(\mathbf{x})\ s(\mathbf{x})] = \sum_{\mathbf{k}} [\mathbf{B}\mathbf{s}]_{\mathbf{k}}\, g_{\mathbf{k}}(\mathbf{x}). \tag{29}$$

The formula for the matrix, $\mathbf{B}$, is derived by first completing the square in the exponent of the product of two Gaussians to obtain:

$$g_{\mathbf{k}}(\mathbf{x}) g_{\ell}(\mathbf{x}) = g(\sqrt{2}(\mathbf{x} - \tfrac{\Delta}{2}(\mathbf{k} + \ell\,)))g(\tfrac{\Delta}{\sqrt{2}}(\mathbf{k} - \ell\,)). \tag{30}$$

This product is then convolved with h to obtain a function, $f(\mathbf{x})$, which is a shift of the Gaussian basis function, $g(\mathbf{x})$. Finally we use the shiftability formula,

$$g(\mathbf{x} - \mathbf{x}_0) = \sum_{\mathbf{k}} a_{\mathbf{k}}(\mathbf{x}_0) g_{\mathbf{k}}(\mathbf{x}), \tag{31}$$

where $a_{\mathbf{k}}$ are the interpolation functions, to express $f(\mathbf{x})$ in the Gaussian basis. The result is

$$B_{\mathbf{k},\ell} = \sum_{\mathbf{i}} b_{\mathbf{i}} \exp(-\|\mathbf{i} - \ell\,\|^2/4) a_{\mathbf{k}}(\Delta(\mathbf{i} + \ell\,)/2). \tag{32}$$

Although this formula is very general, using it to compute the matrix-vector product would be prohibitively expensive due to the non-sparseness of the matrix, $\mathbf{B}$. To overcome this problem we have developed an alternate representation for the bias operator in the Gaussian-Fourier basis. For this representation, we assume that the input bias function is of the form,

$$b(\mathbf{x}) = \sum_{j=1}^{N} g(\mathbf{x} - \mathbf{x}_j), \tag{33}$$

where the centers, $\mathbf{x}_j$, of the Gaussian spots are assumed to be far apart from each other relative to their widths. In this case, the matrix $\mathbf{B}$ representing the bias operator can be expressed as a sum of N outer products of vectors representing values of the Gaussian basis functions g_{ℓ} and values of the sinc interpolation functions $a_{\mathbf{k}}$ at the locations $\mathbf{x}_j$ of the Gaussian spots:

$$B_{\mathbf{k},\ell} = \sum_{j=1}^{N} g_{\ell}\,(\mathbf{x}_j)\ a_{\mathbf{k}}(\mathbf{x}_j). \tag{34}$$

Consequently the action of the bias operator on a coefficient vector is given by:

$$[\mathbf{B}\mathbf{s}]_{\mathbf{k}} = \sum_{j=1}^{N} [\sum_{\ell} s_{\ell}\, g_{\ell}\,(\mathbf{x}_j)]\ \cdot\ a_{\mathbf{k}}(\mathbf{x}_j). \tag{35}$$

In this form, the bias operator is relatively inexpensive to compute. For this reason, it was used in the experimental results described below. However, in the continuum, this representation of the bias operator does not agree with the representation given in Equation 32. To understand how the second representation is related to the first, we now describe how it was derived. Assume that the input bias function is simply $b(\mathbf{x}) = g(\mathbf{x} - \mathbf{x}_0)$ and that $s(\mathbf{x}) = \sum_{\ell} s_{\ell}\, g_{\ell}\,(\mathbf{x})$. Then

$$\sum_{\ell} [\mathbf{B}\mathbf{s}]_{\ell}\, g_{\ell}\,(\mathbf{x}) = b(\mathbf{x})s(\mathbf{x}) \sum_{\ell} s_{\ell}\, g(\mathbf{x} - \mathbf{x}_0) g_{\ell}\,(\mathbf{x}) \tag{36}$$

$$\simeq [\sum_{\ell} s_{\ell}\, g_{\ell}\,(\mathbf{x}_0)] \cdot g(\mathbf{x} - \mathbf{x}_0), \tag{37}$$

where the final equality holds exactly in the limit that $g(\mathbf{x} - \mathbf{x}_0) \to \delta(\mathbf{x} - \mathbf{x}_0)$, where δ denotes the Dirac delta function. The second representation of the bias operator in the basis then follows by expressing $g(\mathbf{x} - \mathbf{x}_0)$ in the Gaussian basis using the shiftability formula.

5 Experimental results

In our experiments the Gaussian-Fourier basis consisted of $K = 192$ translates (in each spatial dimension) of a Gaussian (of period, $X = 70.0$), and $N = 92$ harmonic signals in the orientation dimension. The standard deviation of the Gaussian was set equal to the shift amount, $\Delta = X/K$. For illustration purposes, all functions were rendered at a resolution of 256×256. The diffusion parameter, σ, equaled 0.1473, and the decay constant, τ, equaled 12.5. The time step, Δt, used to solve the Fokker-Planck equation in the basis equaled $\Delta/2$. The parameters for the cut-off function used to eliminate self-loops were $\alpha = 4$ and $\mu = 15$.

In the first experiment, the input bias function, $b(\mathbf{x})$, consists of eight Gaussian spots spaced equally on a circle. The positions are real valued, *i.e.*, they do not lie on the grid of basis functions. The stochastic completion field, $\int_{S^1} c(\mathbf{u},\phi)\, d\phi$, after five iterations of the power method, is shown in Fig. 3. Although the local orientations of the circle at the locations of the spots were not specified in the input bias function, they are an emergent property of the completion field.

In the second experiment, the input bias function was created by adding twenty randomly positioned Gaussian spots to twenty spots on the boundary of an avocado. See Fig. 4 (left). The stochastic completion field computed using 32 iterations of the power method is shown in Fig. 4 (right). In the last experiment, the input bias function from the first experiment was rotated by 45° and translated by half the distance between the centers of adjacent basis functions, $b(R_{45^\circ}(\mathbf{x} - [\frac{\Delta}{2}, \frac{\Delta}{2}]^{\mathrm{T}}))$. See Fig. 5 (left). The stochastic completion field is identical (up to rotation and translation) to the one computed in the first experiment. See Fig. 5 (right). Finally, we plot the estimate of the largest positive real eigenvalue, λ, as a function of m, the power method iteration for both the rotated and non-rotated input patterns. See Fig. 6. The final value of λ (and all intermediate values) are identical in both cases. It follows that the $Q(.)$ operator has the same eigenvalues for the rotated and non-rotated inputs, *i.e.*, the $Q(.)$'s are related by a similarity transform. This vividly demonstrates the Euclidean invariance of the computation.

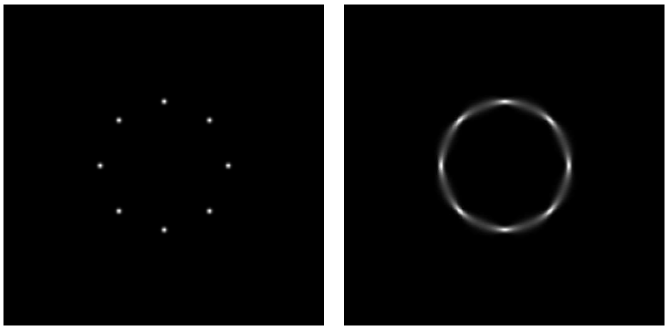

Fig. 3 Left: The input bias function, $b(\mathbf{x})$, consists of eight Gaussian spots spaced equally on a circle. The positions are real valued, *i.e.*, they do not lie on the grid of basis functions. Right: The stochastic completion field, $\int_{S^1} c(\mathbf{u},\phi)\, d\phi$, computed using $192 \times 192 \times 92$ basis functions.

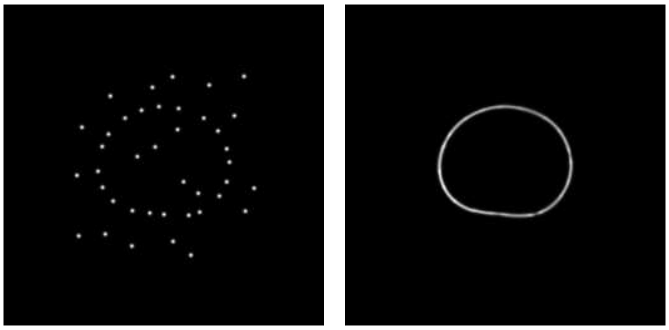

Fig. 4 Left: The input bias function was created by adding twenty randomly positioned Gaussian spots to twenty spots on the boundary of an avocado. The positions are real valued, *i.e.*, they do not lie on the grid of basis functions. Right: The stochastic completion field, $\int_{S^1} c(\mathbf{u},\phi)\, d\phi$, computed using $192 \times 192 \times 92$ basis functions.

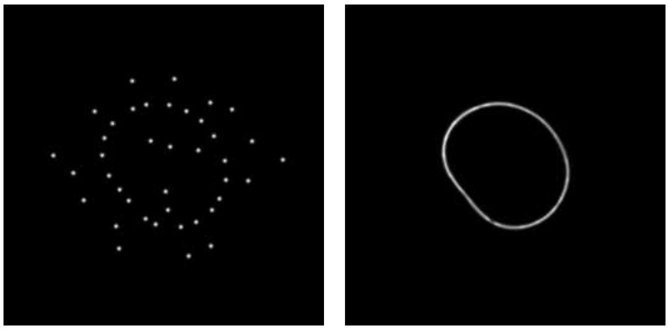

Fig. 5 Left: The input bias function from Fig. 4, rotated by 45° and translated by half the distance between the centers of adjacent basis functions, $b(R_{45^\circ}(\mathbf{x} - [\frac{\Delta}{2}, \frac{\Delta}{2}]^{\mathrm{T}}))$. Right: The stochastic completion field, is identical (up to rotation and translation) to the one shown in Fig. 4. This demonstrates the Euclidean invariance of the computation.

6 Discussion

In this section we discuss the relationship of our model to the known structure and function of areas V1 and V2 of visual cortex.

6.1 Isotropy vs. anisotropy

Given the physical anisotropy of our environment (a consequence of the gravitational field) the observation of a corresponding anisotropy in the statistics of natural scenes is not surprising. However, it does not follow that this environmental anisotropy is necessarily manifest in the computations performed by V1, or in its structure, especially since it is not manifest in the circularly symmetric shapes of retinal and

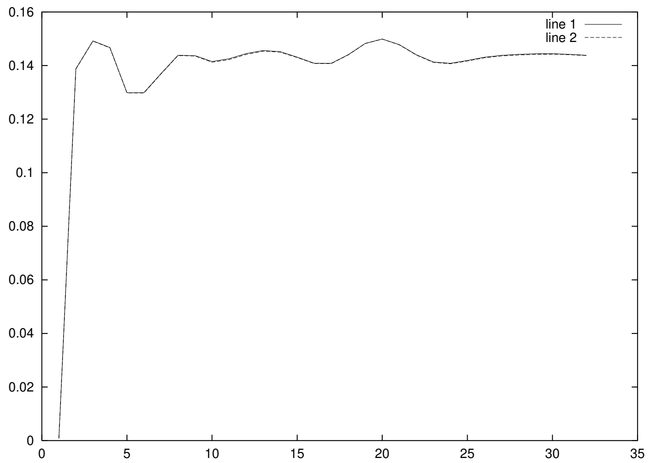

Fig. 6 The estimate of the largest positive real eigenvalue, λ, as a function of *m*, the power method iteration for the rotated (line 2) and non-rotated (line 1) input patterns. The final value of λ (and all intermediate values) are identical in both cases, hence line 1 and line 2 overlap completely.

LGN receptive fields–areas which immediately precede V1 in the visual pathway.

As far as the representation of orientation in V1 is concerned, the evidence from neuroscience is inconclusive. On the one hand, there is evidence for the overrepresentation of horizontal and vertical orientations in the primary visual cortex of ferrets. See Chapman *et al.* (1998). On the other hand, we are aware of no corresponding result for primates, and many authors have described the distribution of orientation preference in area V1 of monkeys as uniform, *e.g.*, Bartfeld and Grinwald (1992), Hubel and Wiesel (1974).

6.2 Continuity vs. isotropy

A computation is continuous if and only if a continuous transformation of the input produces a continuous (not necessarily identical) transformation of the output. It follows that continuity with respect to rotation is a weaker property than isotropy. Indeed, there are well known examples of visual computations which are continuous w.r.t. rotation but not isotropic, *e.g.*, the dependence of shape-from-shading on illumination direction. See Sun and Perona (1995). Admittedly, in the model we describe here, we have achieved the first (weaker) property by enforcing the second (stronger) property. However, to the best of our knowledge, it is an open question whether contour computations in V1 are isotropic or are merely continuous w.r.t. rotation.

In anycase, it would be a mistake to conclude that because human vision is not completely isotropic, that there is a lack of continuity w.r.t. rotation in contour computations in V1. Yet, apart from our own, the output of no existing model of contour completion varies continuously under continuous rotation of its input. This is due to the non-shiftability-twistability of the representations of position and orientation used in these models.

6.3 Receptive fields vs. basis functions

In neuroscience, a receptive field characterizes the response of a neuron to a stimulus which varies in position, size, phase, and/or orientation. In contrast, a basis function is a concept from linear algebra. We hypothesize that the functions which comprise the states of visual computations are represented by expansions in bases consisting of a family of functions related by translation, dilation, phase-shift, and/or rotation. In other words, basis functions are the building blocks of visual intermediate representations.

Under the assumption that the retina, LGN, and V1 can be modeled as linear shift-invariant systems, the notion of a receptive field can be given a precise meaning. Namely, it can be regarded as the *impulse response function* of the system. Under this assumption, the relationship of receptive fields to basis functions can also be formally characterized as a *frame* and its *dual*. Recently, Salinas and Abbott (2000) have argued that simple cell receptive fields do not form a *tight frame*. If they are correct, then the corresponding basis functions, which are only implicitly defined, do not have the same shape as the receptive fields. Consequently, even under the strong assumption of linearity, and even when the dimensionalities of the input and the intermediate representation are the same, it is a mistake to equate receptive fields and basis functions.

With regard to V1 and V2, this is a moot point, since the assumption of linearity is known to be unrealistic. Many researchers have demonstrated that the response of simple cells in V1 is modulated by stimuli outside of the classical (*i.e.*, linear) receptive field. Furthermore, the dynamics of this modulatory activity is consistent with a computation implemented in a recurrent network, not a simple feedforward convolution of the input from LGN. The assumption of linearity is even less justified in V2, where von der Heydt *et al.* (1984) have identified neurons which "respond" to illusory contours of specific position and orientation. Since these neurons do not respond to an isolated spot, the concept of impulse response function is not useful.

6.4 Relative resolution of V1 and V2

The variance of the Gaussian basis functions required to represent the completion field is half of that required to represent the source and sink fields. See Zweck and Williams (2000). It follows that our model would be supported if the spatial resolution of V2 was twice as high as that of V1. However, the fact that V2 receptive fields are (on average) larger than those of V1, does not mean that our model is wrong. Given the non-linearity of the proposed computation, it would be a mistake to equate the size of receptive fields and the size of the basis functions used to represent the completion field. Indeed, given their spatial support in V1, one would *expect* V2 receptive fields to be significantly larger. A better measure of spatial resolution would be average distance (in visual angle) between orientation "pinwheel" centers.

However, it may simply be the case that the completion field is not computed at its theoretically maximum resolution relative to the resolutions of the source and sink fields, i.e., V2 represents a lowpass, downsampled completion field. We note that V1 directly projects to brain regions other than V2, primarily area MT but also back to LGN. It is possible that the computational requirements of MT (or perhaps another computation within V2) constrains the resolution of V1 more strongly than does contour completion.

6.5 Locality in orientation

One significant difference between our computer model and the visual cortex is the exact form of the basis functions which span $\mathbf{R}^2 \times S^1$. In a previous section, we proposed that observed long range, intra-cortical, excitatory connections within V1 are the neural embodiment of the $P_0(.)$ operator. The fact that these connections are orientation specific implies that the basis functions which represent the source field are localized in S^1. However, our implementation uses harmonic signals, which have zero locality, to represent the orientation dimension. The problem of identifying a basis of shiftable-twistable functions which are localized in orientation, and specifying the corresponding $P_0(.)$ operator, is a topic for future research.

7 Conclusion

We described a neural network which enhances and completes salient closed contours. Even though the computation is implemented in a discrete network, its output is invariant under continuous rotations and translations of the input.

Acknowledgements

L.R.W. was supported (in part) by Los Alamos National Laboratory. J.W.Z. was supported (in part) by the Albuquerque High Performance Computing Center. We wish to thank Jonas August and Steve Zucker for insightful comments.

References

1. August J (2000) The curve indicator random field, Ph.D. dissertation, Dept. of Computer Science, Yale University.
2. Bartfeld E and Grinvald A (1992) Relationships between orientation preference pinwheels, cytochrome oxidase blobs and ocular dominance columns in primate striate cortex, *Proc. Natl. Acad. Sci. USA* **89**, pp. 1195-1199.
3. Ben-Yishai R, Lev Bar-Or R and Sompolinksy H (1995) Theory of orientation tuning in visual cortex, *Proc. Natl. Acad. of Sci. (USA)* **92**, pp. 3844-3848.
4. Bressloff PC, Cowan JD, Golubitsky M, Thomas P and Wiener M (2001) Geometric visual hallucinations, Euclidean symmetry, and the functional architecture of striate cortex, *Phil. Trans. Roy. Soc. (Lond.)* **B 356**, pp. 1-32.
5. Chapman B, Bonhoeffer T (1998) Overrepresentation of horizontal and vertical orientation preferences in developing ferret area 17. *Proc Natl Acad Sci (USA)* **95**, pp. 2609-14
6. Freeman W and Adelson E (1991) The design and use of steerable filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(9) pp.891-906.
7. Mumford D (1994) Elastica and computer vision, *Algebraic Geometry and Its Applications*, Chandrajit Bajaj (ed.) Springer-Verlag, New York.
8. Golub GH and Van Loan CF (1996) *Matrix Computations*, Baltimore, MD, Johns Hopkins Univ. Press.
9. Grossberg S and Mingolla E (1985) Neural dynamics of form perception: boundary completion, illusory contours and neon color spreading, *Psychology Review* **92**, pp. 173-211.
10. Guy G and Medioni G (1996) Inferring global perceptual contours from local features, *Intl. Journal of Computer Vision* **20**, pp. 113-133.
11. Heitger R and von der Heydt R (1993) A computational model of neural contour processing, figure-ground and illusory contours, *Proc. of 4th Intl. Conf. on Computer Vision*, Berlin, Germany.
12. Hubel DH and Wiesel TN (1974) Sequence regularity and geometry of orientation columns in the monkey striate cortex, *J. Comp. Neurol.* **158**, pp. 267-294.
13. Iverson L (1993) Toward discrete geometric models for early vision, Ph.D. dissertation, McGill University.
14. Li Z (1998) A neural model of contour integration in primary visual cortex, *Neural Computation* **10**(4) pp. 903-940.
15. Parent P and Zucker SW (1989) Trace inference, curvature consistency and curve detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, pp. 823-889.
16. Salinas E and Abbott LF (2000) Do simple cells in primary visual cortex form a tight frame?, *Neural Computation* **12**(2), pp. 313-335.
17. Shashua A and Ullman S (1988) Structural saliency: The detection of globally salient structures using a locally connected network, *2nd Intl. Conf. on Computer Vision*, Clearwater, FL, pp. 321-327.
18. Simoncelli E, Freeman W, Adelson E and Heeger D (1992) Shiftable multiscale transforms, *IEEE Trans. Information Theory* **38**(2) pp. 587-607.
19. Somers DC, Nelson SB and Sur M (1995) An emergent model of orientation selectivity in cat visual cortical cells, *Journal of Neuroscience* **15**, pp. 5448-5465.
20. Sun J and Perona P (1998) Where is the sun?, *Nature Neuroscience***1**, pp. 183-184.
21. Sompolinsky H and Shapley R (1997) New perspectives on the mechanisms for orientation selectivity, *Current Opinion in Neurobiology***7**, pp. 514-522.
22. von der Heydt R, Peterhans E and Baumgartner G (1984) Illusory contours and cortical neuron responses, *Science***224**, pp. 1260-1262.
23. Williams LR and Jacobs DW (1997) Stochastic completion fields: A neural model of illusory contour shape and salience, *Neural Computation* **9**(4), pp. 837-858.
24. Williams LR and Thornber KK (2001) Orientation, scale, and discontinuity as emergent properties of illusory contour shape, *Neural Computation* **13**(8), pp. 1683-1711.
25. Yen S and Finkel L (1996) Salient contour extraction by temporal binding in a cortically-based network, *Neural Information Processing Systems* **9**, Denver, CO.
26. Zweck JW and Williams LR (2000) Euclidean group invariant computation of stochastic completion fields using shiftable-twistable functions, *Proc. European Conf. on Computer Vision (ECCV '00)*, Dublin, Ireland.