Field Computation in Natural and Artificial Intelligence

Extended Version

Technical Report UT-CS-99-422

Bruce J. MacLennan*

Computer Science Department
University of Tennessee, Knoxville
maclennan@cs.utk.edu

April 14, 1999

Abstract

We review the concepts of *field computation*, a model of computation that processes information represented as spatially continuous arrangements of continuous data. We show that many processes in the brain are described usefully as field computation. Throughout we stress the connections between field computation and quantum mechanics, especially including the important role of *information fields*, which represent by virtue of their form rather than their magnitude. We also show that field computation permits simultaneous nonlinear computation in linear superposition.

1 Motivation for Field Computation

In this paper we discuss the applications of *field computation* to natural and artificial intelligence. (More detailed discussions of field computation can be found in prior

 It is based on an invited presentation for the *Third International Conference on Computational Intelligence and Neuroscience* (October 24–28, 1998), special session on Neuro-Quantum Information Processing.

publications, e.g. MacLennan 1987, 1990, 1993b, 1997.) For this purpose, a *field* is defined to be a spatially continuous arrangement of continuous data. Examples of fields include two-dimensional visual images, one-dimensional continuous spectra, two- or three-dimensional spatial maps, as well as ordinary physical fields, both scalar and vector. A *field transformation* operates in parallel on one or more fields to yield an output field. Examples include summations (linear superpositions), convolutions, correlations, Laplacians, Fourier transforms and wavelet transforms. Field computation may be nonrecurrent (entirely feed-forward), in which a field passes through a fixed series of transformations, or it may be recurrent (including feedback), in which one or more fields are iteratively transformed, either continuously or in discrete steps. Finally, in field computation, the topology of the field (that is, of the space over which it is extended) is generally significant, either in terms of the information it represents (e.g. the dimensions of the field correspond to significant dimensions of the stimulus), or in terms of the permitted interactions (e.g. only local interactions).

Field computation is a theoretical model of certain information processing operations and processes that take place in natural and artificial systems. As a model, it is useful for describing some natural systems and for designing some artificial systems. The theory may be applied regardless of whether the system is actually discrete or continuous in structure, so long as it is approximately continuous. We may make an analogy to hydrodynamics: although we know that a fluid is composed of discrete particles, it is nevertheless worthwhile for most purposes to treat it as a continuum. So also in field computation, an array of data may be treated as a field so long as the number of data elements is sufficiently large to be treated as a continuum, and the quanta by which an element varies are small enough so that it can be treated as a continuous variable.

Physicists sometimes distinguish between *structural fields*, which describe phenomena that are physically continuous (such as gravitational fields), and *phenomenological fields*, which are approximate descriptions of discontinuous phenomena (e.g. velocity fields of fluids). Field computation deals with phenomenological fields in the sense that it doesn't matter whether their realizations are spatially discrete or continuous, so long as the continuum limit is a good mathematical approximation to the computational process. Thus, we have a sort of "complementarity principle," which permits the computation to be treated as discrete or continuous as convenient to the situation (MacLennan 1993a).

Neural computation follows different principles from conventional, digital computing. Digital computation functions by long series of high-speed, high-precision discrete operations. The degree of parallelism is quite modest, even in the latest "massively parallel" computers. We may say that conventional computation is *deep but narrow*. Neural computation, in contrast, functions by the massively parallel application of low-speed, low-precision continuous (analog) operations. The sequential length of computations is typically short (the "100 Step Rule"), as dictated by the real-time response requirements of animals. Thus, neural computation is *shallow but* *broad.* As a consequence of these differences we find that neural computation typically requires very large numbers of neurons to fulfill its purpose. In most of these cases the neural mass is sufficiently large — 15 million neurons/cm^2 (Changeux 1985, p. 51) — that it is useful to treat it as a continuum.

To achieve by artificial intelligence the levels of skillful behavior that we observe in animals, it is not unreasonable to suppose that we will need a similar computational architecture, comprising very large numbers of comparatively slow, low precision analog devices. Our current VLSI technology, which is oriented toward the fabrication of only moderately large numbers of precisely-wired, fast, high-precision digital devices, makes the wrong tradeoffs for efficient, economical neurocomputers; it is unlikely to lead to neurocomputers approximating the 15 million neurons/cm^2 density of mammalian cortex. Fortunately, the brain shows what can be achieved with large numbers of slow, low-precision analog devices, which are (initially) imprecisely connected. This style of computation opens up new computing technologies, which make different tradeoffs from conventional VLSI. The theory of field computation shows us how to exploit relatively homogeneous masses of computational materials (e.g. thin films), such as may be produced by chemical manufacturing processes. The theory of field computation aims to guide our design and use of such radically different computers.

2 Overview of Field Computation

A field is treated mathematically as a continuous function ψ over a bounded set Ω representing the spatial extent of the field. Typically, the value of the function is restricted to some bounded subset of the real numbers, but complex- and vector-valued fields are also useful. Thus we may write $\psi : \Omega \to K$ for a K-valued field.

We write $\psi(u)$ or ψ_u for the value of a field ψ at $u \in \Omega$. If the field is time-varying, we write $\psi(t)$ for the field, and $\psi(u,t)$ or $\psi_u(t)$ for its value at $u \in \Omega$. Further, to stress the connections between field computation and quantum mechanics, we may denote real or complex fields with Dirac's (1958) bracket notation, $|\psi\rangle$ or $|\psi(t)\rangle$, as appropriate. With this notation, the value of $|\psi\rangle$ at u is given by the inner product $\langle u \mid \psi\rangle$, where $\langle u| = \langle\delta_u|$ is a Dirac delta function (unit impulse) located at u.[1]

Fields are required to be *physically realizable*, which places restrictions on the allowable functions. I have already mentioned that fields are continuous functions over a bounded domain that take their values in a bounded subset of a linear space. Furthermore, it is generally reasonable to assume that fields are uniformly continuous square-integrable (e.g. finite-energy) functions, $\|\psi\|^2 = \langle\psi \mid \psi\rangle < \infty$, and therefore that fields belong to a Hilbert space of functions. Thus Hilbert spaces provide the vocabulary of field computation as they do of quantum mechanics. (To stress this

[1]If $\phi, \psi \in \Phi(\Omega)$ are fields of the same type, we use $\langle\psi \mid \psi\rangle$ and $\langle\phi, \psi\rangle$ for the appropriate inner product on these fields. If they are real- or complex-valued, then $\langle\phi \mid \psi\rangle = \int_\Omega \phi_u^* \psi_u \mathrm{d}u$, where ϕ_u^* is the complex conjugate of ϕ_u. If the fields are vector-valued, then $\langle\phi \mid \psi\rangle = \int_\Omega \phi_u \cdot \psi_u \mathrm{d}u$, where $\phi_u \cdot \psi_u$ is the ordinary scalar product of the vectors.

commonality, this paper will follow the notational conventions of quantum mechanics.) Nevertheless, not all elements of a Hilbert space are physically realizable, so we write $\Phi_K(\Omega)$ for the set of all K-valued fields over Ω (the subscript K is omitted when clear from context). (See Pribram 1991 and MacLennan 1990, 1993a, 1993b, 1994a, 1997 for more on Hilbert spaces as models of continuous knowledge representation in the brain; see MacLennan 1990 for more on the physical realizability of fields.)

A *field transformation* is any continuous (linear or nonlinear) function that maps one or more input fields into one or more output fields. Since a field comprises an uncountable infinity of points, the elements of a field cannot be processed individually in a finite number of discrete steps, but a field can be processed sequentially by a continuous process, which sweeps over the input field and generates the corresponding output sequentially in finite time. Normally, however, a field transformation operates in parallel on the entire input field and generates all elements of the output at once.

One important class of linear field transformations are *integral operators of Hilbert-Schmidt type*, which can be written

$$\psi_u = \int_\Omega K_{uv} \phi_v \mathrm{d}v \tag{1}$$

where $\psi \in \Phi(\Omega')$, $\phi \in \Phi(\Omega)$ and K is a finite energy field in $\Phi(\Omega' \times \Omega)$. Equation (1) may be abbreviated $\psi = K\phi$ or, as is common in quantum mechanics, $|\psi\rangle = K|\phi\rangle$. We also allow multilinear integral operators. If $\phi_k \in \Phi(\Omega_k), k = 1, \ldots, n$ and $M \in \Phi(\Omega' \times \Omega_n \times \cdots \times \Omega_2 \times \Omega_1)$, then $\psi = M\phi_1\phi_2 \cdots \phi_n$ abbreviates

$$\psi_u = \int_{\Omega_n} \cdots \int_{\Omega_2} \int_{\Omega_1} M_{uv_n \cdots v_2 v_1} \phi_1(v_1)\phi_2(v_2) \cdots \phi_n(v_n) \mathrm{d}v_1 \mathrm{d}v_2 \cdots \mathrm{d}v_n.$$

Many useful information processing tasks can be implemented by a composition of field transformations, which feeds the field(s) through a fixed series of processing stages. (One might expect sensory systems to be implemented by such feed-forward processes, but in fact we find feedback at almost every stage of sensory processing, so they are better treated as recurrent computations, discussed next.)

In many cases we are interested in the dynamical properties of fields: how they change in time. The changes are usually continuous, defined by differential equations, but may also proceed by discrete steps. As with the fields treated in physics, we are often most interested in dynamics defined by local interactions, although nonlocal interactions are also used in field computation (several examples are considered later). For example, Pribram (1991) has discussed a *neural wave equation*, $i\nu\dot{\psi} = (-\frac{\nu^2}{2}\nabla^2 + U)\psi$, which is formally identical to the Schrödinger equation.

One reason for dynamic fields is that the field may be converging to some solution by a recurrent field computation; for example, the field might be relaxing into the most coherent interpretation of perceptual data, or into an optimal solution of some other problem. Alternately, the time-varying field may be used for some kind of real-time control, such as motor control (MacLennan 1997).

An interesting question is whether there can be a universal field computer, that is, a general purpose device (analogous to a universal Turing machine) that can be programmed to compute any field transformation (in a large, important class of transformations, analogous to the Turing-computable functions). In fact, we have shown (Wolpert & MacLennan submitted) that any Turing machine, including a universal Turing machine, can be emulated by a corresponding field computer, but this does not seem to be the concept of universality that is most relevant to field computation. Another notion of universality is provided by an analog of Taylor's theorem for Hilbert spaces. It shows how arbitrary field transformations can be approximated by a kind of "field polynomial" computed by a series of products between the input field and fixed "coefficient" fields (MacLennan 1987, 1990). In particular, if $F : \Phi(\Omega) \to \Phi(\Omega')$ is a (possibly nonlinear) field transformation, then it can be expanded around a fixed field $\varpi \in \Phi(\Omega)$ by:

$$F(\varpi + \phi) = F(\varpi) + \sum_{k=1}^{\infty} \frac{D_k \phi^{(k)}}{k!},$$

where

$$D_k \phi^{(k)} = D_k \underbrace{\phi\phi \cdots \phi}_{k},$$

and the fields $D_k \in \Phi(\Omega' \times \Omega^k)$ are the kernels of the (both Fréchet and Gâteaux) derivatives of F evaluated at ϖ, $D_k = \mathrm{d}^k F(\varpi)$. More generally, nonlinear field transformations can be expanded as "field polynomials":

$$F(\phi) = K_0 + K_1\phi + K_2\phi^{(2)} + K_3\phi^{(3)} + \cdots.$$

Adaptation and learning can be accomplished by field computation versions of many of the common neural network learning algorithms, although some are more appropriate to field computation than others. In particular, a field-computation version of back-propagation is straight-forward, and Peruš (1996, 1998) has investigated field-computation versions of Hopfield networks. Learning typically operates by computing or modifying "coefficient fields" or connection fields in a computational structure of fixed architecture.

3 Field Computation in the Brain

There are a number of processes in the brain that may be described usefully as field computation. In this section we discuss axonal fields, dendritic fields, projection fields and synaptic fields. (There are, however, other possibilities, such as conformational fields on the surfaces of dendritic microtubules, which we will not discuss.)

3.1 Axonal Fields

Computational maps are ubiquitous in the brain (Knudsen et al. 1987). For example, there are the well-known maps in somatosensory and motor cortex, in which the neurons form a topological image of the body. There are also the *retinotopic* maps in the vision areas, in which locations in the map mirror locations on the retina, as well as other properties, such as the orientation of edges. Auditory cortex contains *tonotopic* maps, with locations in the map systematically representing frequencies in the manner of a spectrum. Auditory areas in the bat's brain provide further examples, with systematic representations of Doppler shift and time delay, among other significant quantities.

We may describe a computational map as follows. We are given some abstract space X, which often represents a class of microfeatures or stimuli (e.g. particular pitches, locations on the surface of the body, oriented edges at particular places in the visual field). If these stimuli or microfeatures are represented spatially over a brain region Ω, then there is a piecewise continuous map $\mu : X \to \Omega$ giving the location $u_x = \mu(x)$ optimally tuned to microfeature value $x \in X$. The presence of microfeature x will typically lead to strong activity at $\mu(x)$ and lesser activity at surrounding locations; we may visualize it as an approximate (typically two-dimensional) Gaussian centered at $\mu(x)$. In general we will use the notation γ_x or $|\gamma_x\rangle$ for a localized pattern of activity resulting from a stimulus x. When the pattern of activity is especially sharply defined, it may be approximated by δ_x (also written $|\delta_x\rangle$ or $|x\rangle$), a Dirac delta-function centered at the location corresponding to x. (We may write γ_u or δ_u when the neural coordinates $u = \mu(x)$ are more relevant than the microfeature $x \in X$.) The amplitude s of the peak $s\delta_x$ may encode the degree of presence of the microfeature or stimulus x.

In the presence of multiple stimuli, such maps typically represent a superposition of all the stimuli. For example, if several frequencies are present in a sound, then a tonotopic map will show corresponding peaks of activity. Similarly, if there are patches of light (or other visual microfeatures, such as oriented grating patches) at many locations in the visual field, then a retinotopic map will have peaks of activity corresponding to all of these microfeatures. Thus, if features $x_1, x_2, \ldots, x_n$ are all present, the corresponding computational map is $\gamma_{x_1} + \gamma_{x_2} + \cdots + \gamma_{x_n}$ (possibly with corresponding scale factors). In this way the *form* of the stimulus may be represented as a superposition of microfeatures.

Computational maps such as these are reasonably treated as fields, and it is useful to treat the information processing in them as field computation. Indeed, since the cortex is estimated to contain at least 146,000 neurons per square millimeter (Changeux 1985, p. 51), even a square millimeter has sufficient neurons to be treated as a continuum, and in fact there are computational maps in the brain of this size and smaller (Knudsen et al. 1987). Even one tenth of a square millimeter contains sufficient neurons to be treated as a field for many purposes. The larger maps are directly observable by noninvasive imaging technique, such as fMRI.

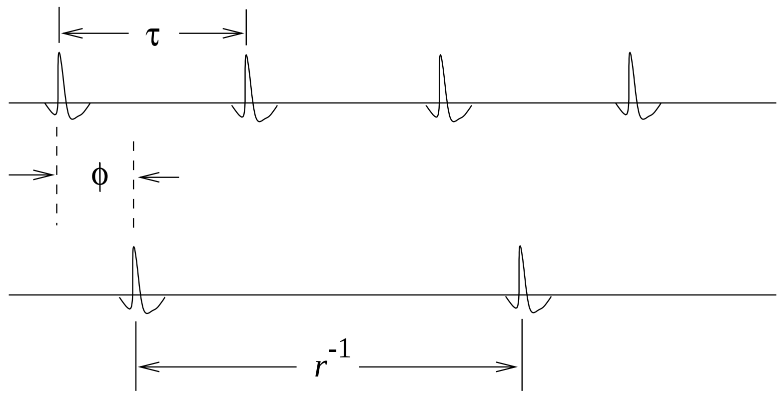

Figure 1: Phase-encoding of Axonal Signals

We refer to these fields as *axonal fields*, because the field's value at each location corresponds to the axonal spiking (e.g. rate and/or phase) of the neuron at that location. If only the rate is significant, then it is appropriate to treat the field as real-valued. If both rate and phase are significant (Hopfield 1995), then it is more appropriate to treat it as complex-valued.

To see this, consider the relation between an axonal signal and a fixed "clock signal" with period τ (Fig. 1). Two pieces of information may be conveyed (e.g. to a dendrite upon which both axons synapse). The first is the delay $\phi(t)$ between the clock and the signal (at time t), which is represented by the phase angle $\theta(t) = 2\pi\phi(t)/\tau$. (Such a delay might result from a difference in the integration times of a neuron representing a fixed standard and one encoding some microfeature or other property.) Second, the average impulse rate $r(t)$ may represent pragmatic factors such as the importance, urgency or confidence level of the information represented by the phase. (This dual representation of pragmatics and semantics is discussed further below, Section 5.) The two together constitute a time-varying complex-valued signal, which can be written as the complex exponential,

$$z(t) = r(t)e^{2\pi i\phi(t)/\tau} = r(t)e^{i\theta(t)}.$$

More generally, if we have multiple signals, then the information may be encoded in their relative phases, and the clock signal is unnecessary. This is especially the case for complex-valued axonal fields, in which the field value is represented in the rate and relative phase of the axonal impulses.

3.2 Projection Fields

Next we can consider *projection fields* (or *connection fields*), which are determined by the patterns of axonal connections between brain regions. Typically they operate on an axonal field and, in the process of transmitting it elsewhere in the brain, transform it to yield another axonal field. Projection fields usually correspond to the kernel of

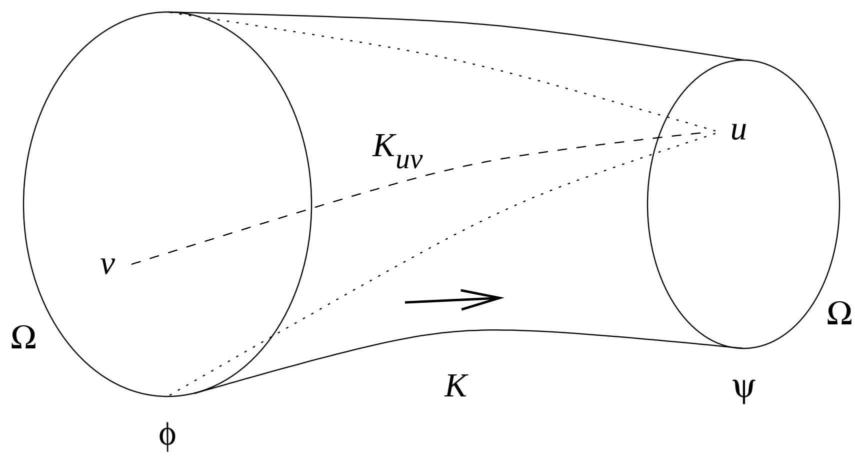

Figure 2: Projection Field

a linear operator. To see this, suppose that a bundle of axons projects from region Ω to region Ω'. For $u \in \Omega', v \in \Omega$, let K_{uv} represent the connection to u from v (Fig. 2). (K_{uv} could be a complex number representing the effect of the axon on the signal; it is 0 if there is no axon connecting v to u.) Then, the activity ψ_u at destination u is expressed in terms of the activities ϕ_v of source neurons v by $\psi_u = \int_\Omega K_{uv}\phi_v \mathrm{d}v$; that is, $\psi = K\phi$ or $|\psi\rangle = K|\phi\rangle$. Thus the projection field K is a linear operator. Since $K \in \Phi(\Omega' \times \Omega)$, the projection field's topology is determined by Ω and Ω', the topologies of the source and destination regions. Projection fields may be quite large (i.e. they are anatomically observable) and change quite slowly (e.g. through development); their information processing role is discussed further below.

A linear operator (of Hilbert-Schmidt type) can be resolved into a discrete neural network by methods familiar from quantum mechanics. Let $|\epsilon_k\rangle$ be the eigenfields (eigenstates) of a linear operator L with corresponding eigenvalues ℓ_k. Since the eigenfields can be chosen to be orthonormal, an input field $|\phi\rangle$ can be represented by a discrete set of coordinates $c_k = \langle \epsilon_k \mid \phi \rangle$. (The coordinates are discrete because there is no significant topological relationship among them.) Then, $|\psi\rangle = L|\phi\rangle$ can be expanded:

$$\begin{aligned}
|\psi\rangle &= L|\phi\rangle \\
&= L\sum_k |\epsilon_k\rangle\langle\epsilon_k \mid \phi\rangle \\
&= L\sum_k |\epsilon_k\rangle c_k \\
&= \sum_k L|\epsilon_k\rangle c_k \\
&= \sum_k \ell_k|\epsilon_k\rangle c_k.
\end{aligned}$$

Only a finite number of the eigenvalues are greater than any fixed bound, so the

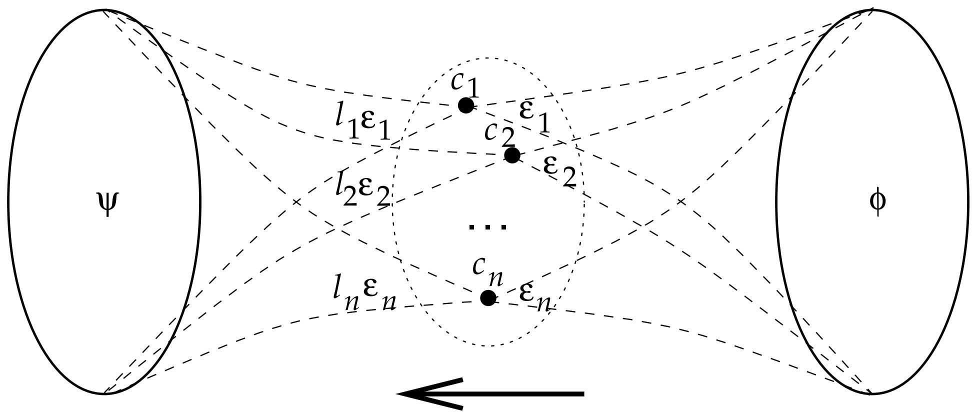

Figure 3: Computation of Linear Operator Factored through Eigenfield Basis

operator can be approximated by a finite sum. In the first part of the computation, the discrete set of coefficient c_k are computed by a finite number of neurons with receptive field profiles ϵ_k. In the second stage, each of these neurons projects its activity c_k with a profile $\ell_k \epsilon_k$ (Fig. 3).

It is not necessary to use the eigenfields of the operator, for we can resolve the input field into any set of orthonormal base fields $|\epsilon_k\rangle$ and the output field into any set of orthonormal base fields $|\zeta_j\rangle$. Then,

$$|\psi\rangle = \sum_j |\zeta_j\rangle\langle\zeta_j \mid \psi\rangle = \sum_j |\zeta_j\rangle\langle\zeta_j \mid L \mid \phi\rangle.$$

But,

$$L|\phi\rangle = \sum_k L|\epsilon_k\rangle\langle\epsilon_k \mid \phi\rangle.$$

Hence,

$$|\psi\rangle = \sum_j |\zeta_j\rangle \sum_k \langle\zeta_j \mid L \mid \epsilon_k\rangle\langle\epsilon_k \mid \phi\rangle = \sum_{jk} |\zeta_j\rangle\langle\zeta_j \mid L \mid \epsilon_k\rangle\langle\epsilon_k \mid \phi\rangle.$$

Let $c_k = \langle\epsilon_k \mid \phi\rangle$ be the representation of the input and $M_{jk} = \langle\zeta_j \mid L \mid \epsilon_k\rangle$ the representation of the operation. Then $d_j = \langle\zeta_j \mid \psi\rangle$, the representation of the output, is given by a discrete matrix product $\mathbf{d} = M\mathbf{c}$ (Fig. 4). When a linear operator is factored in this way, it can be computed through a neural space of comparatively low dimension. Such a representation might be used when the projection field (kernel) of L would be too dense.

Generally speaking, axons introduce phase delays, but do not affect the amplitudes or rates of the signals they transmit. Therefore the effect of a projection field can be described by an imaginary exponential field, $K_{uv} = e^{i\theta_{uv}}$. However, since multiple impulses are typically required to cause the exocytosis of neurotransmitter from an axon terminal, the axon terminal has the effect of scaling the impulse rate by a factor less than 1 (Fig. 5). Therefore, the combined effect of the axon and axon terminal

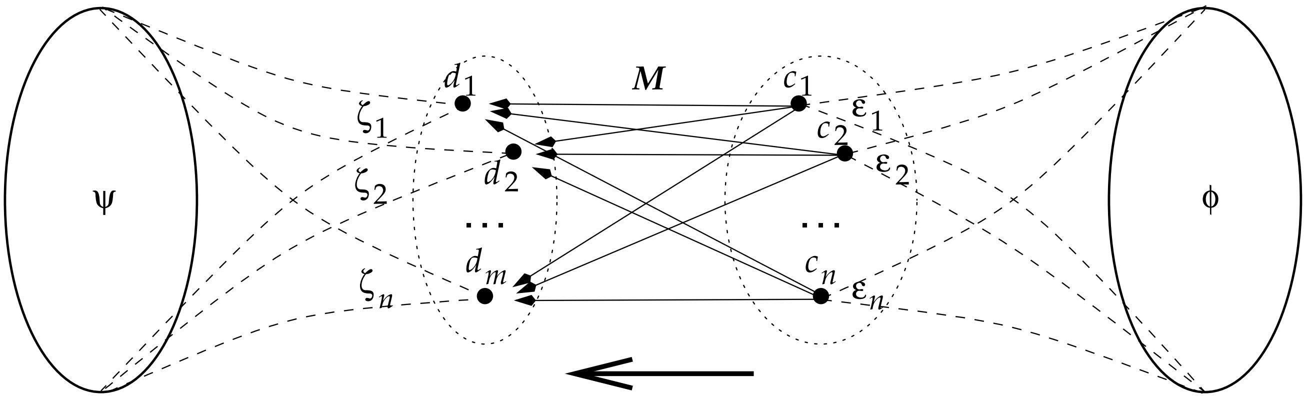

Figure 4: Computation of Linear Operator Factored through Arbitrary Bases

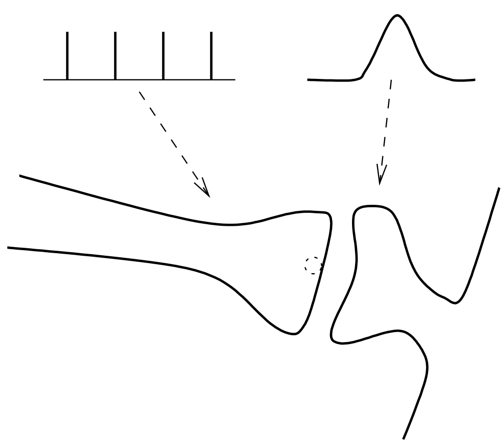

Figure 5: Rate Scaling at Axon Terminal

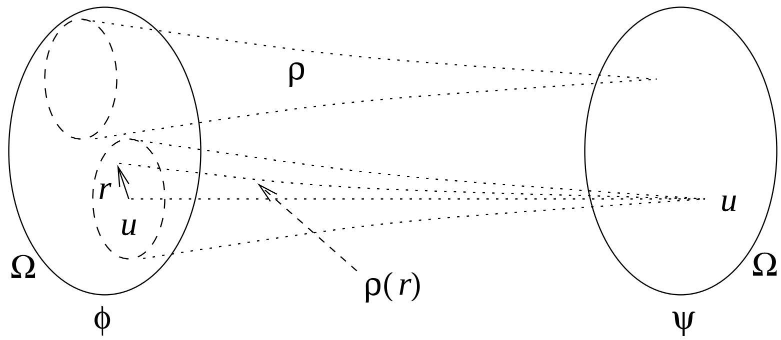

Figure 6: Correlation Field

is to multiply by a complex exponential, $K_{uv} = s_{uv} e^{i\theta_{uv}}$, where s_{uv} and θ_{uv} are real, $0 \leq s_{uv} \leq 1$ and $0 \leq \theta_{uv} < 2\pi$.

Two common kinds of projection fields are *correlation* and *convolution fields*; in each of these the destination neurons have identical receptive field profiles. For example, if the receptive field profile is approximately Gaussian, then the projection field *coarse codes* (by Gaussian smoothing) an input represented in a computational map.

More precisely, let ϕ and ψ be input and output fields defined over the same domain Ω (i.e., the source and destination regions have the same shape). Each output neuron u has the same receptive field profile ρ, defined as a field over Ω, but centered on the corresponding location u in the input region (Fig. 6).[2] The activity of output neuron u is the sum of the activities of the neurons surrounding input location u, but weighted by the receptive field profile:

$$\psi_u = \int_\Omega \rho^*(r)\phi(u+r)\mathrm{d}r.$$

(We use the complex conjugate ρ^* to accommodate complex-valued receptive field profiles.) By letting $s = u + r$ we can see that ψ is the *cross-correlation* of ρ and ϕ:

$$\psi_u = \int_\Omega \rho^*(s-u)\phi(s)\mathrm{d}s$$

or $\psi = \rho \star \phi$. Equivalently, if, as is often the case, the receptive field profile is symmetric, $\rho(-r) = \rho(r)$, we may write ψ as a convolution:

$$\psi_u = \int_\Omega \rho^*(u-s)\phi(s)\mathrm{d}s$$

or $\psi = \rho \circledast \phi$. (Convolution is easier to manipulate than correlation, since its properties are more like ordinary multiplication.) The complete projection field is given by $R_{us} = \rho^*(s-u)$ so that $\psi = R\phi$ or $|\psi\rangle = R|\phi\rangle$.

[2]This presumes that Ω is a linear space (e.g. a two-dimensional Euclidean space), so that it makes sense to translate the receptive fields.

3.3 Synaptic and Dendritic Fields

A projection field typically terminates in a *synaptic field*, which denotes the mass of synapses forming the inputs to a group of related neurons. Synaptic fields represent the interface between a projection field and a dendritic field (discussed next). The topology of a synaptic field is determined by the spatial arrangement of the synapses relative to the axon terminals and the dendrites that they connect. A synaptic field's value σ_u corresponds to the efficacy of synapse u, which is determined by the number of receptor sites and similar factors. In the case of synaptic fields, the transmitted signal is given by a pointwise product $\sigma(u)\psi(u)$ between the synaptic field σ and the input field ψ. Frequently a projection field and its synaptic field can be treated as a single linear operator,

$$\sigma_u \int_\Omega K_{uv}\phi_v \mathrm{d}v = \int_\Omega \sigma_u K_{uv}\phi_v \mathrm{d}v = \int_\Omega L_{uv}\phi_v \mathrm{d}v,$$

where $L_{uv} = \sigma_u K_{uv}$. Synaptic fields change comparatively slowly under the control of neurological development and learning (e.g. long-term potentiation).

Another place where field computation occurs in the brain is in the dendritic trees of neurons (MacLennan 1993a). The tree of a single pyramidal cell may have several hundred thousand inputs, and signals propagate down the tree by passive electrical processes (resistive and capacitive). Therefore, the dendritic tree acts as a large, approximately linear analog filter operating on the neuron's input field, which may be significant in dendritic information processing. In this case, the field values are represented by neurotransmitter concentrations, electrical charges and currents in the dendritic tree; such fields are called *dendritic fields*. Such a field may have a complicated topology, since it is determined by the morphology of the dendritic tree over which it's spread.

Analysis of the dendritic net suggests that the antidromic electrical impulse caused by the firing of the neuron could trigger a simple adaptive process which would cause the dendritic net to tune itself to be a *matched filter* for the recent input pattern (MacLennan 1993a, 1994a).

4 Examples of Field Computation

4.1 Gabor Wavelets and Coherent States

Dennis Gabor (1946) developed a theory of information by generalizing the Heisenberg-Weyl derivation of the Uncertainty Principle to arbitrary (finite-energy) signals. He presented it in the context of scalar functions of time; I will discuss it more generally (see MacLennan 1991 for further details). Let $\phi(\mathbf{x})$ be a field defined over an n-dimensional Euclidean space. We may define the uncertainty along the k-th

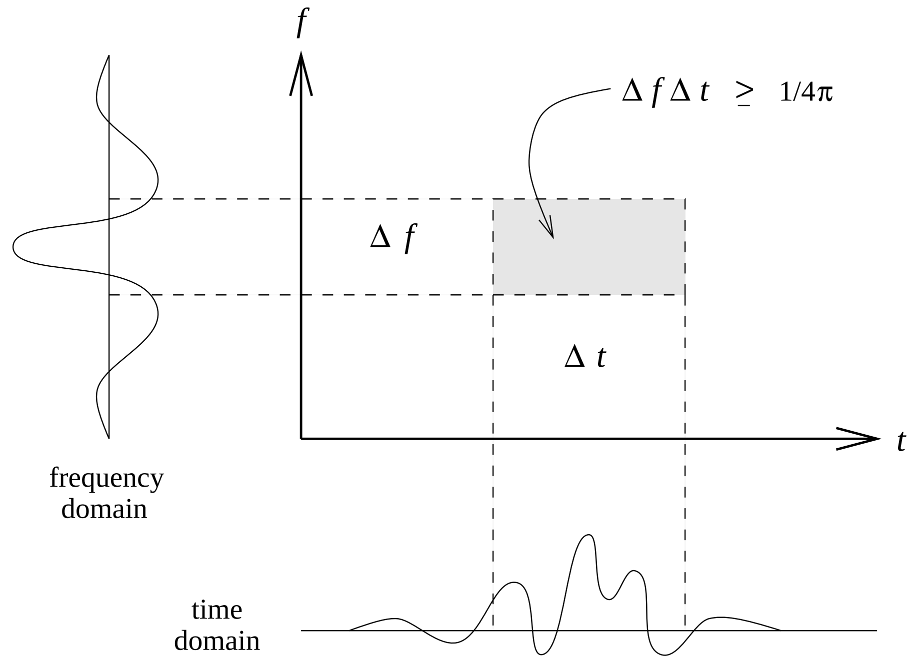

Figure 7: Minimum Uncertainty in "Gabor Space"

dimension by the root mean square deviation of x_k (assumed to have 0 mean):

$$\Delta x_k = \|x_k \phi(\mathbf{x})\| = \sqrt{\int_\Omega \phi^*_{\mathbf{x}} x_k^2 \phi_{\mathbf{x}} \mathrm{d}\mathbf{x}}.$$

Likewise, the uncertainty along the k-th conjugate axis is measured by the root mean square deviation of u_k for the Fourier transform $\Phi(\mathbf{u})$ of $\phi(\mathbf{x})$:

$$\Delta u_k = \|(u_k - \bar{u})\Phi(\mathbf{u})\| = \sqrt{\int_\Omega \Phi^*_{\mathbf{u}} u_k^2 \Phi_{\mathbf{u}} \mathrm{d}\mathbf{u}}.$$

As in quantum mechanics, we can show $\Delta x_k \Delta u_k \geq 1/4\pi$ (Fig. 7). The minimum joint uncertainty $\Delta x_k \Delta u_k = 1/4\pi$ is achieved by the *Gabor elementary functions*, which are Gaussian-modulated complex exponentials and correspond to the *coherent states* of quantum mechanics (Fig. 8):

$$G_{\mathbf{pu}}(\mathbf{x}) = \exp[-\pi \|S(\mathbf{x}-\mathbf{p})\|^2] \exp[2\pi i \mathbf{u} \cdot (\mathbf{x}-\mathbf{p})].$$

The second, imaginary exponential defines a plane wave; the frequency and direction of the wave packet are determined by the wave vector $\mathbf{u}$. The first, real exponential defines a Gaussian envelope centered at $\mathbf{p}$, which has a shape determined by the diagonal *aspect matrix* $S = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_n)$, which determines the spread in each variable and its conjugate,

$$\Delta x_k = \frac{\alpha_k}{2\sqrt{\pi}}, \quad \Delta u_k = \frac{\alpha_k^{-1}}{2\sqrt{\pi}}.$$

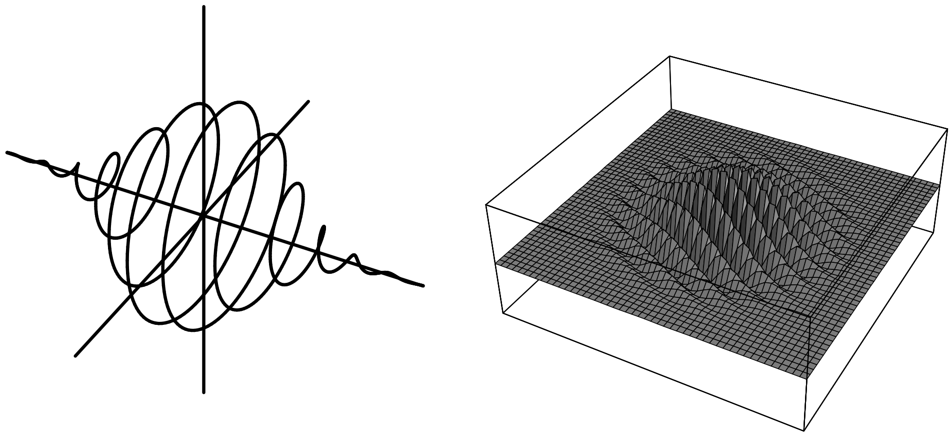

Figure 8: Gabor Elementary Function (Coherent State). On the left is a (complex-valued) Gabor elementary function of one dimension; on the right is the real part of a Gabor elementary function of two dimensions.

Each Gabor elementary function occupies a cell in $2n$-dimensional "Gabor space" of volume

$$\prod_{k=1}^{n} \Delta x_k \Delta u_k = \frac{1}{(4\pi)^n}.$$

Each of these cells corresponds to an elementary unit of information, which Gabor called a *logon*.

Now suppose we have a field $\phi(\mathbf{x})$, finite in extent and bandwidth in all dimensions; it occupies a bounded region in $2n$-dimensional Gabor space. A given choice of $\alpha_1, \alpha_2, \ldots, \alpha_n$ will divide this region into cells of minimum size. Corresponding to each cell will be a Gabor elementary function; we may index them arbitrarily $G_k(\mathbf{x}), k = 1, 2, \ldots, N$.

We may calculate N, the number of cells, as follows. Let X_k be the extent of ϕ along the k-th axis and let U_k be its bandwidth in the k-th conjugate variable. Then there are $m_k = X_k/\Delta x_k$ cells along the k-th axis and $n_k = U_k/\Delta u_k$ along the k-th conjugate axis. Therefore, the maximum number of cells is

$$N = \prod_{k=1}^{n} m_k n_k = \prod_{k=1}^{n} \frac{X_k}{\Delta x_k} U_k \Delta x_k = \prod_{k=1}^{n} X_k U_k.$$

That is, the maximum number of logons of information is given by the volume of the signal in Gabor space.

Gabor showed that any finite-energy function could be represented as a superposition of such elementary functions scaled by complex coefficients:

$$|\phi\rangle = \sum_{k=1}^{N} c_k |G_k\rangle.$$

However, the Gabor elementary functions are not orthogonal, so the complex coefficients are not given by $c_k = \langle G_k \mid \phi \rangle$. Nevertheless, for appropriate choices of the

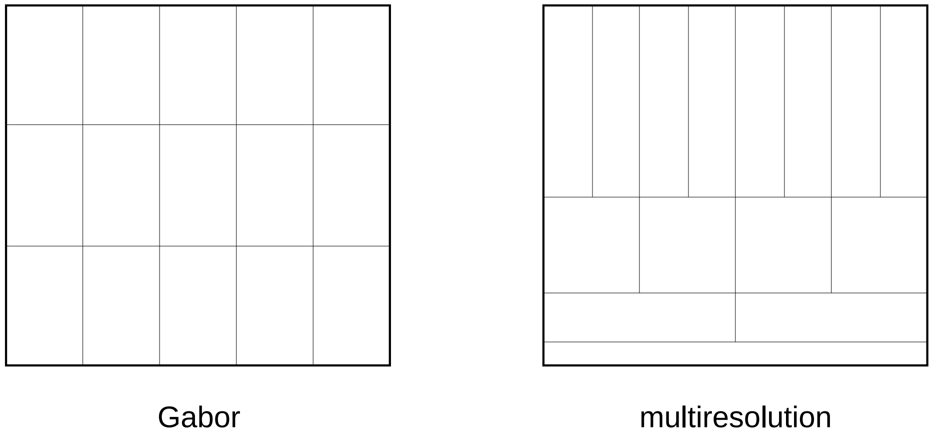

Figure 9: Fixed- versus Multiresolution Gabor Cells. In this example the field contains 15 logons of information.

parameters, the Gabor elementary functions constitute a *tight frame* (MacLennan 1991), for which

$$|\phi\rangle \approx \sum_{k=1}^{N} |G_k\rangle\langle G_k \mid \phi\rangle.$$

The consequence of the foregoing for information theory is that the field ϕ has exactly N independent degrees of freedom, and thus can represent at most N logons of information (ignoring noise etc.).

There is considerable evidence (reviewed in MacLennan 1991; see also Pribram 1991) that images in primary visual cortex (V1) are represented in terms of *Gabor wavelets*, that is, hierarchically arranged, Gaussian-modulated sinusoids. Whereas the Gabor elementary functions are all of the same shape (determined by S), Gabor wavelets scale Δu_k with frequency (and Δx_k inversely with frequency) to maintain a constant $\Delta u_k/u_k$, thus giving a multiresolution representation. (Typically, they are scaled by powers of 2; see Fig. 9.)

The Gabor-wavelet transform of a two-dimensional visual field generates a four-dimensional field: two of the dimensions are spatial, the other two represent spatial frequency and orientation. To represent this four-dimensional field in two-dimensional cortex, it is necessary to "slice" the field, which gives rise to the columns and stripes of striate cortex. The representation is nearly optimal, as defined by the Gabor Uncertainty Principle (Daugman 1984). Time-varying two-dimensional visual images may be viewed as three-dimensional functions of space-time, and it is possible that time-varying images are represented in vision areas by a three-dimensional Gabor-wavelet transform, which generates a time-varying five-dimensional field (representing two spatial dimensions, spatial frequency, spatial orientation and temporal frequency). The effect is to represent the "optic flow" of images in terms of spatially fixed, oriented grating patches with moving gratings. (See MacLennan 1991 for more details.) Finally, Pribram provides evidence that Gabor representations are also used for controlling the generation of motor fields (Pribram & al. 1984, Pribram 1991, pp. 139–144).

4.2 Motion in Direction Fields

Another example of field computation in the brain is provided by direction fields, in which a direction in space is encoded in the activity pattern over a brain region (Georgopoulos 1995). Such a region is characterized by a vector field $\mathbf{D}$ in which the vector value $\mathbf{D}_u$ at each neural location u gives the preferred direction encoded by the neuron at that location. The population code ϕ for a direction $\mathbf{r}$ is proportional to the scalar field given by the inner product of $\mathbf{r}$ at each point of $\mathbf{D}$, that is, $\phi_u \propto \mathbf{r} \cdot \mathbf{D}_u$. Typically, it will have a peak at the location corresponding to $\mathbf{r}$ and will fall off as the cosine of the angle between this vector and the surrounding neurons' preferred directions, which is precisely what is observed in cortex. (See MacLennan 1997, section 6.2, for a more detailed discussion.)

Field computation is used in the brain for modifying direction fields. For example, a direction field representing a remembered location, relative to the retina, must be updated when the eye moves (Droulez & Berthoz 1991a, 1991b), and the peak of the direction field must move like a particle in a direction determined by the velocity vector of the eye motion. The change in the direction field is given by a differential field equation, in which the change in the value of the direction field is given by the inner product of the eye velocity vector and the gradient of the direction field: $\mathrm{d}\phi/\mathrm{d}t = \mathbf{v} \cdot \nabla\phi$. Each component (x and y) of the gradient is approximated by a convolution between the direction field and a "derivative of Gaussian" (DoG) field, which is implemented by the DoG shape of the receptive fields of the neurons. (See MacLennan 1997, section 6.3, for a more detailed discussion.)

Other examples of field computation in motor control include the control of frog leg position by the linear superposition of convergent force fields generated by spinal neurons (Bizzi & Mussa-Ivaldi 1995), and the computation of convergent vector fields, defining motions to positions in head-centered space, from positions in retina-centered space, as represented by products of simple receptive fields and linear gain fields (Andersen 1995). (See MacLennan 1997, section 6, for more details.)

4.3 Nonlinear Computation in Linear Superposition

One kind of field transformation, which is very useful and may be quite common in the brain, is similar to a *radial basis function (RBF) neural network* (Fig. 10). The input field ϕ is a computational map, which encodes significant stimulus values by the location of peak activity within the field (similar to the direction fields already discussed). The transformation has two stages. The first stage is a correlation $\chi = \rho \star \phi$ between the input field and a local "basis field" ρ (such as a Gaussian); this

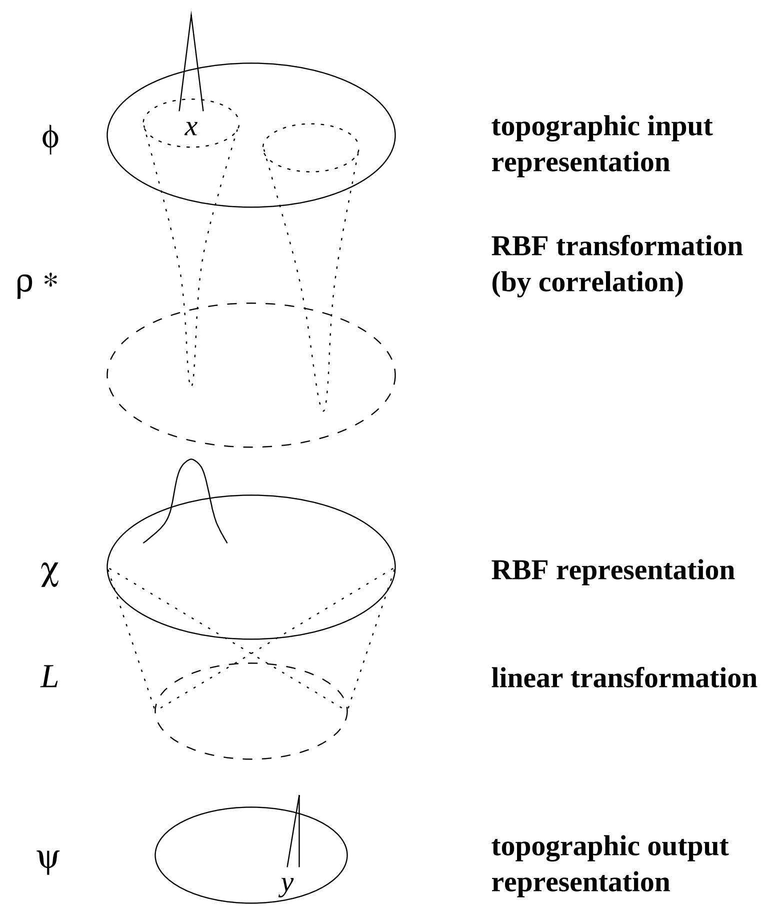

Figure 10: Field Computation Analogous to Radial Basis Function Network

"coarse codes" the stimulus as a pattern of activity. (We do not require the basis field to be strictly radial, $\rho(r) = f(\|r\|)$, although it commonly is.) This stage is implemented by a projection field to a layer of neurons with identical receptive field profiles given by the basis field ρ. The second stage is a linear transformation $L\chi$ of the coarse-coded field, which yields the output field; it is also implemented by a single layer of neurons. Thus the transformation is given by $L(\rho \star \phi)$, where ϕ is the input, ρ is the basis field, and L is the linear transformation.

Now we will carry out the construction in more detail. In an RBF network a function $F : X \to Y$ is approximated by a linear combination of radial functions of the form:

$$F(x) = \sum_{k=1}^{N} L_k f(\|x - x_k\|).$$

For a given F, the coefficients L_k, centers x_k and radial function f are all fixed. It has been shown (Lowe 1991, Moody & Darken 1989, Wettscherick & Dietterich 1992) that simple networks of this form are universal in an important sense, and can adapt through a simple learning algorithm.

In transferring these ideas to field computation, we make three changes. First, as a basis we use functions $\rho(x - x_k)$ which need not be radial, although radial functions are included as a special case. Second, we represent the input $x \in X$ by a computational map $\gamma_x \in \Phi(\Omega)$ or, more ideally, by δ_x; that is, the input will be encoded by a field with a peak of activity at the location corresponding to the input. Finally, in accord with the goals of field computation, we replace the summation with integration:

$$F(x) = \int_\Omega L_v \rho(x - x_v) \mathrm{d}v.$$

There are two parts to this operation, the coarse-coding χ of the input by the basis functions and the linear transformation of the result.

Because, in our continuous formulation, there is a radial function centered at each possible location in the input space, the coarse-coded result χ is defined over the same space as the input, so we may write $\chi_y = \rho(x - y)$. However, because the input is encoded by a map δ_x, the coarse coding can be accomplished by a correlation:

$$\chi_y = \rho(x - y) = \int_\Omega \rho(z - y) \delta_x(z) \mathrm{d}z,$$

so $\chi = \rho \star \delta_x$.[3] The output is then computed as a linear function of the correlation field:

$$\psi = \int_\Omega L_y \chi_y \mathrm{d}y = L\chi = L(\rho \star \phi).$$

(Note that the output ψ is typically a field, so that $\psi_z = \int_\Omega L_{zy} \chi_y \mathrm{d}y$.)

[3] This is the sort of projection field correlation that we have already discussed. Observe, however, that the computational map ϕ must preserve distances $x - y$ in X. This restriction may be avoided by using a slightly more complex projection field instead of the correlation (MacLennan 1997, 3.3.4).

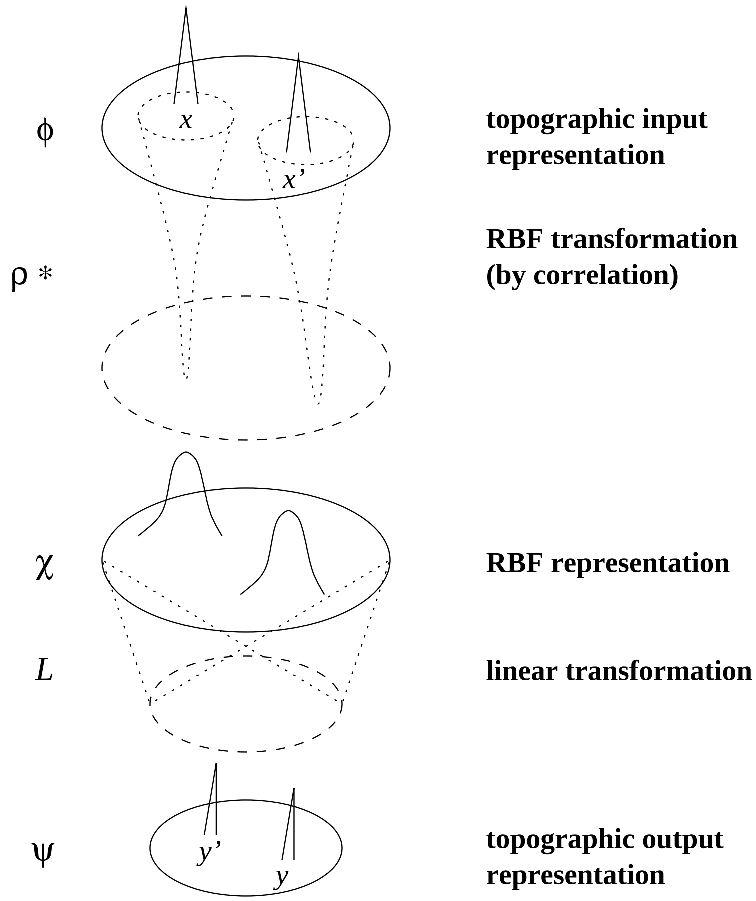

Figure 11: Nonlinear Computation in Linear Superposition

Observe that this transformation is linear in its input field (which does not imply, however, that F is a linear function of the stimulus values). Since, if there are several significant stimuli, the input field will be a superposition of the fields representing the individual stimuli, the output will likewise be a superposition of the corresponding individual outputs. Thus this transformation supports a limited kind of parallel computation in superposition. This is especially useful when the output, like the input, is a computational map, so we will explain this *nonlinear computation in linear superposition* in more detail.

Suppose that the input field is a superposition $\delta_x + \delta_{x'}$ of two sharp peaks representing distinct inputs x and x' (Fig. 11). Since the computation is linear we have $L[\rho\star(\delta_x + \delta_{x'})] = F(x) + F(x')$ in spite of the fact that F need not be linear. Further, if, as is often the case, F has been defined to produce a computational map $\delta_{f(x)}$ for some (possibly nonlinear) f, then the network computes both (nonlinear) results in superposition:

$$L[\rho \star (\delta_x + \delta_{x'})] = \delta_{f(x)} + \delta_{f(x')}.$$

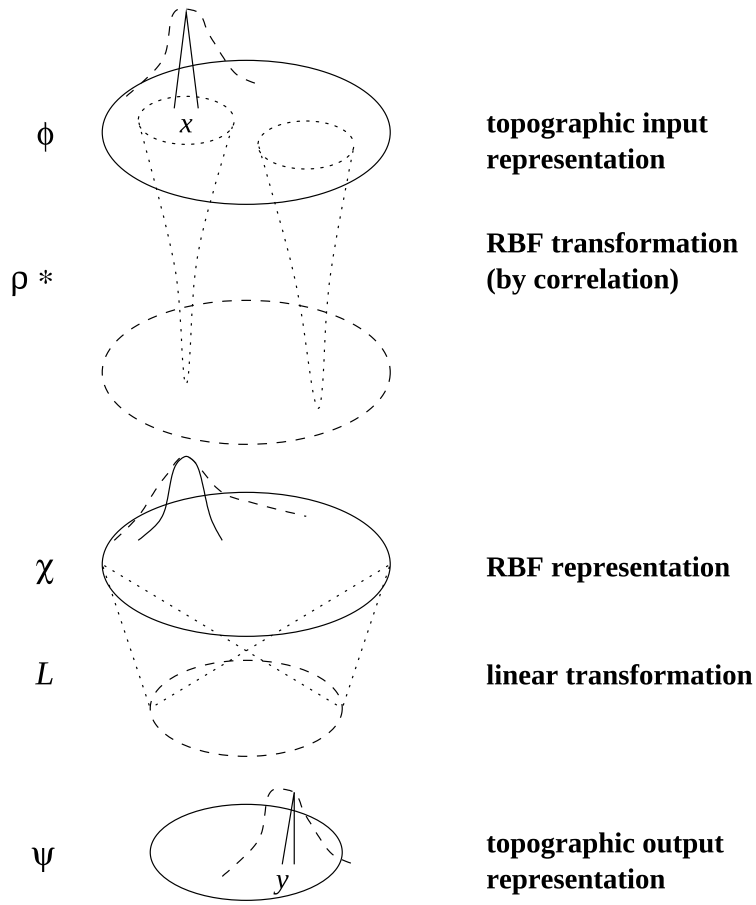

Figure 12: Fuzzy Computation

Further, due to linearity, if the input maps are weighted by s and s', perhaps reflecting pragmatic factors, such as the importance of the inputs, then the outputs are similarly weighted:

$$L[\rho \star (s\delta_x + s'\delta_{x'})] = s\delta_{f(x)} + s'\delta_{f(x')}.$$

Finally, we can consider the case in which the input is a field γ_x, such as a Gaussian, representing a fuzzy estimate of x. The fuzzy envelope γ is defined $\gamma(y - x) = \gamma_x(y)$ (Fig. 12). We may use the identity

$$|\gamma_x\rangle = \int_\Omega |y\rangle\langle y \mid \gamma_x\rangle \mathrm{d}y$$

to compute the output of the network:

$$\begin{aligned} |\psi\rangle &= L(\rho \star |\gamma_x\rangle) \\ &= L\left(\rho \star \int_\Omega |y\rangle\langle y \mid \gamma_x\rangle \mathrm{d}y\right) \\ &= L\int_\Omega \rho \star |y\rangle\langle y \mid \gamma_x\rangle \mathrm{d}y \end{aligned}$$

$$
\begin{aligned}
&= \int_\Omega L(\rho \star |y\rangle)\langle y \mid \gamma_x\rangle \mathrm{d}y \\
&= \int_\Omega F(y)\langle y \mid \gamma_x\rangle \mathrm{d}y \\
&= \int_\Omega F(x+r)\langle x+r \mid \gamma_x\rangle \mathrm{d}r \\
&= \int_\Omega F(x+r)\gamma(r)\mathrm{d}r.
\end{aligned}
$$

Therefore we get a superposition of the outputs $F(x+r)$ weighted by the strengths $\gamma(r)$ of the deviations r of the input. Alternately, since

$$\psi = \int_\Omega F(y)\gamma_x(y)\mathrm{d}y,$$

we might write $|\psi\rangle = F|\gamma_x\rangle$, although it must be recalled that F need not be linear.

4.4 Diffusion Processes

Diffusion processes can be implemented by the spreading activation of neurons, and they can be used for important tasks, such as path planning (Steinbeck & al. 1995) and other kinds of optimization (Miller & al. 1991, Ting & Iltis 1994). In a diffusion process the rate of change of a field is directly proportional to the Laplacian of the field, $\mathrm{d}\psi/\mathrm{d}t \propto \nabla^2\psi$. The Laplacian can be approximated in terms of the convolution of a Gaussian with the field, which is implemented by a simple pattern of connections with nearby neurons: $\mathrm{d}\psi/\mathrm{d}t \propto \gamma \star \psi - \psi$, where γ is a Gaussian field of appropriate dimension. (See MacLennan 1997 for more details.)

5 Information Fields

As previously remarked, Hopfield (1995) has proposed that in some cases the information content of a spike train is encoded in the *phase* of the impulses relative to some global or local clock, whereas the impulse *rate* reflects pragmatic factors, such as the importance of the information. Phase-encoded fields of this sort are typical of the separation of semantics and pragmatics that we find in the nervous system. Information is inherently idempotent: repeating a signal does not affect its semantics, although it may affect its reliability, urgency and other pragmatic factors; the idempotency of information was recognized already by Boole in his *Laws of Thought.*

This characteristic of information may be illustrated as follows:

YES NO

YES NO

The horizontal distinction is semantic, the vertical is pragmatic. The information is conveyed by the difference of form, 'YES' versus 'NO'. The difference of size may affect the urgency, confidence or strength with which the signal is processed. We may say that the form of the signal *guides* the resulting action, whereas its magnitude determines the *amount* of action (Bohm & Hiley 1993, pp. 35–36).

Likewise, an *information field* represents by virtue of its form, that is, the relative magnitude and disposition of its parts; its significance is a holistic property of the field. The overall magnitude of the field does not contribute to its meaning, but may reflect the strength of the signal and thereby influence the confidence or urgency with which it is used. Thus a physical field ψ may be factored $\psi = s\nu$, where $s = ||\psi||$ is its magnitude and ν is the (normalized) information field, representing its meaning. Information fields can be identified in the brain wherever we find processes that depend on the form of a field, but not on its absolute magnitude, or where the form is processed differently from the magnitude. Information fields are idempotent, since repetition and scaling affect the strength but not the form of the field:

$$\psi + \psi = 2\psi = (2s)\nu.$$

Therefore entropy is an information property, since it is depends only on the form of the field, independent of magnitude:

$$S(\psi) = \int_\Omega \frac{\psi_u}{\|\psi\|} \log\left(\frac{\psi_u}{\|\psi\|}\right) \mathrm{d}u = \int_\Omega \nu_u \log \nu_u \mathrm{d}u = \mathrm{tr}(\nu \log \nu) = S(\nu).$$

In the foregoing we have been vague about the norm $\|\psi\|$ we have used. In many cases it will be the familiar L_2 norm, $\|\psi\| = \sqrt{\langle\psi \mid \psi\rangle}$, but when we are dealing with information fields we should select the norm appropriate to the measure of "action" resulting from the field.

Information fields are also central to quantum mechanics. For example, the quantum mechanical state $|\psi\rangle$ is considered undetermined with respect to magnitude (e.g. Dirac 1958, p. 17), so $z|\psi\rangle$ is the same state as $|\psi\rangle$ for any (nonzero) complex z. That is, quantum mechanical states are idempotent. Conventionally, the state is normalized $\|\psi\|^2 = \langle\psi \mid \psi\rangle = 1$, so that its square is a probability density function, $\varrho_x = |\psi_x|^2$.

Of course, this independence of magnitude is also characteristic of the quantum potential, which has led Bohm & Hiley (1993) to characterize this field as *active information.* Thus (following Bohm & Hiley, pp. 28–29), if we write the wave function in polar form, $\psi_x = R_x e^{iS_x/\hbar}$, then the motion of a single particle is described

$$\frac{\partial S_x}{\partial t} + \frac{(\nabla S_x)^2}{2m} + V_x + Q_x = 0,$$

where the quantum potential is defined

$$Q_x = -\frac{\hbar^2}{2m}\frac{\nabla^2 R_x}{R_x}.$$

Notice that because the Laplacian $\nabla^2 R_x$ is scaled by R_x, the quantum potential depends only on the *local form* of the wave function. Further, since scaling the wave function does not affect the quantum potential, $Q(z\psi) = Q(\psi)$, we see that the quantum potential depends only on the form of the wave function. As with many fields in the brain, the strength and form affect the action in different ways: the particle moves under its own energy but the quantum potential controls the energy.

6 Discrete Symbols as Field Excitations

In quantum field theory discrete particles are treated as quantized excitations of the field. Similarly, we have seen particle-like motion of direction fields in the brain (Section 4.2). Therefore it will be worthwhile to see if field computation can illuminate the emergence of discrete symbols from continuous neurological processes. Although traditional, symbolic artificial intelligence takes discrete symbols as givens, understanding their emergence from continuous fields may help to explain the flexibility of human cognition (MacLennan 1994a, 1994b, 1995).

Mathematically, atomic symbols have a *discrete topology*, which means there are only two possible distances between symbols: 0 if they are the same and 1 if they are different. This property also characterizes orthonormal fields (base states), which means that orthonormal fields are a discrete set. To see this, observe that if w, w' are distinct orthonormal fields, then $\langle w \mid w' \rangle = 0$ and $\langle w \mid w \rangle = 1$. Therefore, we define the discrete metric, $d(w, w') = \frac{1}{2}\|w - w'\|^2 = 1$.

The simplest examples of such orthonormal fields are localized patterns of activity approximating Dirac delta functions. Thus distinct symbols w, w' might be represented by fields $\delta_w, \delta_{w'}$; $\langle \delta_w \mid \delta_{w'} \rangle = 0$ and $\langle \delta_w \mid \delta_w \rangle = 1$. More realistically we may have broader patterns of activity $\gamma_w, \gamma_{w'}$, so long as they are sufficiently separated, $\langle \gamma_w \mid \gamma_{w'} \rangle \approx 0$. (If this seems to be a very inefficient way of representing symbols, it is worth recalling that cortical density is approximately 146 thousand neurons per square millimeter.) Such localized patterns of activity may behave like particles, but they also may be created or destroyed or exhibit wave-like properties. However, the discrete topology is not restricted to localized patterns of activity. Nonlocal orthonormal fields v_w have exactly the same discrete properties: $\langle v_w \mid v_{w'} \rangle = 0, \langle v_w \mid v_w \rangle = 1$. (Such patterns are less easily detected through imaging, however.)

Further, wave packets, such as coherent states (Gabor elementary functions), can emerge from the superposition of a number of nonlocal oscillators of similar frequency. (A coherent state results from a Gaussian distribution of frequencies.) The position of the particle is controlled by the relative phase of the oscillators (recall Section 3.1) and its compactness by the bandwidth of the oscillators. (The frequency of the wave packet could encode the role filled by the symbol or establish symbol binding.)

The field approach allows discrete symbols to be treated as special cases of continuous field computation. This illuminates both how discrete symbols may be represented by continuous neural processes and how discrete symbol processing may merge with more flexible analog information processing.

7 Field Computing Hardware

Field computation can, of course, be performed by conventional digital computers or by special-purpose, but conventional digital hardware. However, as noted previously, neural computation and field computation are based on very different tradeoffs from traditional computation, which creates the opportunity for new computing technologies better suited for neural computation and field computation (which is broad but shallow). The ability to use slow, low precision analog devices, imprecisely connected, compensates for the need for very large numbers of computing elements. These characteristics suggest optical information transmission and processing, in which fields are represented by optical wavefronts. They also suggest molecular processes, in which fields are represented by spatial distributions of molecules of different kinds or in different states (e.g. bacteriorhodopsin). Practical field computers of this kind will probably combine optical, molecular and electrical processes for various computing purposes.

For example, Mills (1995) has designed and implemented *Kirkhoff machines*, which operate by diffusion of charge carriers in bulk silicon. This is a special purpose field computer which finds the steady state defined by the diffusion equation with given boundary conditions. Mills has applied it to a number of problems, but its full range of application remains to be discovered.

Further, Skinner & al. (1995) have explored optical implementations of field computers corresponding to feed-forward neural nets trained by back-propagation. The fields are represented in "self-lensing" media, which respond nonlinearly to applied irradiance. The concept has been demonstrated by means of both computer simulation and an optical table prototype.

To date, much of the work on quantum computing has focused on quantum mechanical implementation of binary digital computing. However, field computation seems to be a more natural model for quantum computation, since it makes better use of the full representational potential of the wave function. Indeed, field computation is expressed in terms of Hilbert spaces, which also provide the basic vocabulary of quantum mechanics. Therefore, since many field computations are described by the same mathematics as quantum phenomena, we expect that quantum computers may provide direct, efficient implementations of these computations. Conversely, the mathematics of some quantum-mechanical processes (such as computation in linear superposition) can be transferred to classical systems, where they can be implemented without resorting to quantum phenomena. This can be called *quantum-like computing*, and it may be quite important in the brain (Pribram 1991).

8 Concluding Remarks

In this article I have attempted to provide a brief overview of field computation, presenting it as a model of massively parallel analog computation, which can be applied to natural intelligence, implemented by brains, as well as to artificial intelligence, implemented by suitable field computers. Along the way we have seen many parallels with quantum mechanics, so each may illuminate the other. In particular, we have seen that (1) field computation takes parallel computation to the continuum limit, (2) much information processing in the brain is usefully described as field computation, (3) the mathematics of field computation has much in common with the mathematics of quantum mechanics, (4) computational maps permit nonlinear computation in linear superposition, and (5) information fields are important in both neurocomputation and quantum mechanics. It is my hope that this overview of field computation will entice the reader to look at the more detailed presentations listed in the references and perhaps to explore the field computation perspective.

9 References

1. Andersen, R. A. (1995). Coordinate transformation and motor planning in posterior parietal cortex. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 519–532.

2. Bizzi, E. & Mussa-Ivaldi, F. A. (1995). Toward a neurobiology of coordinate transformation. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 495–506.

3. Bohm, D. & Hiley, B. J. (1993). *The undivided universe: An ontological interpretation of quantum theory*. Routledge.

4. Changeux, J.-P. (1985). *Neuronal man: The biology of mind*, tr. by L. Garey. Oxford.

5. Daugman, J. G. (1984). Spatial visual channels in the Fourier plane. *Vision Res.* **24**: 891–910.

6. Dirac, P. A. M. (1958). *The principles of quantum mechanics*, fourth ed. Oxford.

7. Droulez, J. & Berthoz, A. (1991a). The concept of dynamic memory in sensorimotor control. In D. R. Humphrey & H.-J. Freund (eds.), *Motor Control: Concepts and Issues*, Wiley, pp. 137–161.

8. Droulez, J. & Berthoz, A. (1991b). A neural network model of sensoritopic maps with predictive short-term memory properties. *Proc. National Acad. Science USA* **88**: 9653–9657.

9. Gabor, D. (1946). Theory of communication. *Journ. of the Institution of Electrical Engineers* **93**, Part III: 429–457.

10. Georgopoulos, A. P. (1995). Motor cortex and cognitive processing. In M. S. Gazzaniga (ed.), *The Cognitive Neurosciences*, MIT Press, pp. 507–517.

11. Hopfield, J. J. (1995). Pattern recognition computation using action potential timing for stimulus representation. *Nature* **376**: 33–36.

12. Knudsen, E. J., du Lac, S. & Esterly, S. D. (1987). Computational maps in the brain. *Ann. Rev. of Neuroscience* **10**: 41–65.

13. Lowe, D. (1991). What have neural networks to offer statistical pattern processing? *Proc. SPIE Conf. on Adaptive Signal Processing*, San Diego, pp. 460–471.

14. MacLennan, B. J. (1987). Technology-independent design of neurocomputers: The universal field computer. In M. Caudill & C. Butler (eds.), *Proc. IEEE First Intl. Conf. on Neural Networks*, IEEE Press, Vol. 3, pp. 39–49.

15. MacLennan, B. J. (1990). Field computation: A theoretical framework for massively parallel analog computation, parts I-IV. *Tech. Rep. CS-90-100.* Comp. Sci. Dept., Univ. of Tennessee, Knoxville. Accessible via URL `http://www.cs.utk.edu/~mclennan`.

16. MacLennan, B. J. (1991). Gabor representations of spatiotemporal visual images. *Tech. Rep. CS-91-144.* Comp. Sci. Dept., Univ. of Tennessee, Knoxville. Accessible via URL `http://www.cs.utk.edu/~mclennan`.

17. MacLennan, B. J. (1993a). Information processing in the dendritic net. In K. Pribram (ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum, pp. 161–197.

18. MacLennan, B. J. (1993b). Field computation in the brain. In K. Pribram (ed.), *Rethinking Neural Networks: Quantum Fields and Biological Data*, Lawrence Erlbaum, pp. 199–232.

19. MacLennan, B. J. (1994a). Continuous computation and the emergence of the discrete. In K. Pribram (ed.), *Origins: Brain & Self-Organization*, Lawrence Erlbaum, pp. 121–151.

20. MacLennan, B. J. (1994b). Image and symbol: Continuous computation and the emergence of the discrete. In V. Hanovar & L. Uhr (eds.), *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration, Volume I: Basic Paradigms; Learning Representational Issues; and Integrated Architectures*, Academic Press, pp. 207–240.

21. MacLennan, B. J. (1995). Continuous formal systems: A unifying model in language and cognition. *Proceedings of the IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems*, pp. 161–172.

22. MacLennan, B. J. (1997). Field computation in motor control. In P. Morasso & V. Sanguineti (eds.), *Self-Organization, Computational Maps, and Motor Control*, Elsevier, pp. 37–73.

23. Miller, M.I., Roysam, B., Smith, K. R. & O'Sullivan, J. A. (1991). Representing and computing regular languages on massively parallel networks. *IEEE Trans. Neural Networks* **2**: 56–72.

24. Mills, J. W. (1995). Kirkhoff machines. Technical report, Comp. Sci. Dept., Indiana Univ., Bloomington.

25. Moody, J. E. & Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation* **1**: 281–294.

26. Peruš, M. (1996). Neuro-quantum parallelism in brain-mind and computers. *Informatica* **20**: 173–183.

27. Peruš, M. (1998). A quantum information-processing "algorithm" based on neural nets. In P. Wang, G. Georgiou, C. Janikow & Y. Yao (eds.), *JICS '98 Proceedings*, Vol. II, Assoc. for Intell. Mach., pp. 197–200.

28. Pribram, K. H. (1991). *Brain and perception: Holonomy and structure in figural processing*. Lawrence Erlbaum.

29. Pribram, K. H., Sharafat, A. & Beekman, G. J. (1984). Frequency encoding in motor systems. In H. T. A. Whiting (ed.), *Human Motor Actions — Bernstein Reassessed*, Elsevier, pp. 121–156.

30. Skinner, S.R., Behrman, E.C., Cruz-Cabrera, A.A. & Steck, J.E. (1995). Neural network implementation using self-lensing media. *Applied Optics* **34**: 4129–4135.

31. Steinbeck, O., Tóth, A. & Showalter, K. (1995). Navigating complex labyrinths: Optimal paths from chemical waves. *Science* **267**: 868–871.

32. Ting, P.-Y. & Iltis, R. A. (1994). Diffusion network architectures for implementation of Gibbs samplers with applications to assignment problems. *IEEE Trans. Neural Networks* **5**: 622–638.

33. Wettscherick, D. & Dietterich, T. (1992). Improving the performance of radial basis function networks by learning center locations. In J. E. Moody, S. J. Hanson & R. P. Lippmann (eds.), *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann, pp. 1133–1140.

34. Wolpert, D. H. & MacLennan, B. J. (submitted). A computationally universal field computer which is purely linear. See also Santa Fe Institute *TR 93-09-056* and Univ. of Tennessee, Knoxville, Comp. Sci. Dept. *TR CS-93-206.* Accessible via URL `http://www.cs.utk.edu/~mclennan`.