Which Computation Runs in Visual Cortical Columns?

Steven W. Zucker*

Center for Computational Vision and Control
Depts. of Computer Science and Electrical Engineering
Yale University
New Haven, CT U.S.A.
steven.zucker@yale.edu

Abstract. It is often assumed that the unit of neural computation is either the neuron or the synapse. In either case a complexity barrier arises when such units are composed to realistic levels: namely, how can function on the scale of visual cortex be understood? The answer entails specifying the class of problems that is natural for visual cortex, and the computations that are natural on it. Any such solutions must agree with the time and accuracy constraints imposed by the physiology. By analogy with computers, we submit that answers will be found by supplementing modeling at the microscopic level (transistors vs. synapses) with formal analysis at the macroscopic level. We thus propose seeking an abstract theory of neural computation. As an example, we focus on vision and show how one class of computations (linear complementarity problems) generalizes the computational competence of visual cortex from filtering, local selection, and constraint satisfaction to solving polymatrix games. The natural "unit" of computation is a particular cell assembly, when the abstraction is reduced to neurons, and emergent properties are described in terms of differential geometry. The rapidity and reliability of computations (a few spikes in about 25 msec) provide a surprising consequence.

To appear in: **Problems in Systems Neuroscience**
J. Leo van Hemmen and T. J. Sejnowski (eds.)
Oxford University Press.

1 Introduction

Behaviour results from the solution of problems: we plan activities; we sense our environment; we accomplish motor tasks. Each of these vague statements

* Research supported by AFOSR and Yale University. S. Alibhai implemented the stereo computations, L. Iverson computed the co-circular compatibilities, and P. Huggins the Klein bottle examples. I thank J. Allman, A. Hasenstaub, and P. Huggins for comments.

actually implies a universe of specific problems, such as object identification, trajectory planning, and motor control. But even these problems need further specification before they can be examined in the laboratory: in the limit of this specification, data (e.g., spike trains) characterize the response of a particular neuron in V1 to a moving bar stimulus at a particular orientation, contrast, and velocity, to take one (still partially specified) example. To understand this neuron further, synaptic arrangements can be measured, and neurotransmitters identified. How can such data be put together into a theory of visual cortex?

The task normally ascribed to theory is to develop models that predict data such as that above; and a close relationship between modeling and experiment (Jennings and Aamodt, 2000) is sought. For example, one might model synaptic facilitation and nonlinearities. This encourages a "devil's in the details" perspective. The hunt is on for "the basic unit of computation" (Zador, 2000), which, according to many, may reside in synaptic nonlinearities. Little, if any, analysis is aimed toward models at a global, abstract level, and few connections, if any, are attempted between the capabilities of the model and the problem that originally motivated it.

There is no question that such data, and such models, are a necessary part of understanding information processing in the brain. The question is whether they are sufficent. Consider what could happen with success. The trend toward specifics would create the danger of burying theory under mountains of detail. An experimental program that provided direct measurements for all the neurons in the visual system, even limiting physiology to receptive-field characteristics and anatomy to a listing of synaptic interactions, would result in a suffocating amount of data. In the limit, modeling would become simulation, and the effective complexity of the simulation would be of the same order as the data. How could it be explained; generalized; applied to other problems? With such high dimensionality, how could it be related to the original problem? To be concrete: How much of the edge detection problem does the moving bar experiment capture? To answer questions such as these requires abstraction, and our goal in this paper is to raise questions about the nature of this abstraction. In order of specificity, the questions are as follows.

Perhaps the most natural place to seek abstraction is in a model for neural computation, and the basic question we ask is *what does "computation" mean in the phrase "neural computation?"* Digital computers immediately suggest an answer. In brief, since computers consist of logical circuits, it follows that, if certain primitive functions (such as AND, OR, NOT, etc.) could be implemented in neural circuitry, then any computation could be implemented by arranging these primitives in an appropriate fashion. This view is attractive because of its universality, and it provides a foundation for building computations out of neural components (Shepard, 1990; Koch, 1998).

I do not believe that such a strategy is enough, because a successful pairing between logical functions and synapses would, in the end, get buried exactly as above. There could be about as many "logical gates" as there are dendritic interactions (maybe more). In effect we would be no better off than the designers of VLSI circuits, and we know how difficult it is to design them (Lengauer, 1993). There are limits to how far a designer can go without tools to help with abstraction, in which complex circuits at the silicon level become "units" at the design level. Nevertheless, the most complex circuits constructed are still many orders of magnitude simpler than the visual cortex. "Silicon neurons" can be designed into circuits providing there is symmetry in the design, either in layout (e.g., spatially regular arrays such as those that arise in retinal models) or in interconnection (e.g., complete recriprical connections (Hahnloser et al., 2000)) , but it is unclear how these simplifications relate to information processing. Often the temptation is to model a unit in a complex, but without the interactions that define the complex (example: those models for a single cortical column that include no inter-columnar interactions (Hansel and Sompolinsky, 1998)); or to design the interactions, with no detail on the units (Mumford, 1994; Ullman, 1994). Surely both are required: understanding the behavior of a single ball is clearly part of understanding the game of billiards, but it is also clear that emergent properties of the game simply do not exist at the single ball level.

Thus we claim that, while local models are important, it is as important to explore the other way around; that is, to find the right types of abstraction for characterizing the cortical machine. This will relate to the classes of problems being solved on it, and it will highlight their emergent properties. To be useful, abstraction must help in framing problems, loosely in the way high-level programming languages provide a framework for conceptualization. (VLSI design environments share this feature with programming languages.) But our task is not to seek a programming language in the standard sense, because our abstraction must articulate those aspects of the physiology that are constraining, a point in direct contrast to the way in which programming languages often hide the computer's architecture from the programmer. From this perspective the separation of levels in (Marr, 1982) might need to be reconsidered; there are constructive senses in which the "problem", the "computational" level, and the "implementation" level are intimately related. One would not expect visual area V1, with its highly articulated structure (discussed below), to be Turing universal. We ask, instead, what V1 is good for, and which emergent properties of problems are naturally encoded in it.

Underlying this paper is an assumption that a deep connection exists between the functional architecture of cortical areas and the computational abstractions they support. This may not be the case. It could be argued, for example, that columnar architectures exist because of evolutionary accidents and not because of emergent computations. This is not to say that there is no genetic component to brain evolution, which is incorrect, but to raise the possibility that, during evolution, certain genes that say control body segmentation processes became encapsulated in the brain's developmental sequence and they simply remained in place since then. To our knowledge no evidence currently exists to support this accidental position (Allman, 1999). Or it may be argued that there is no deep connection between global computations and function; rather it is just a result of learning (Koch, 1997), and no satisfying global explanation exists. If so, then the comments above about generalization and applicability apply again. Showing that a basic computational abstraction exists would argue against both of these positions, and could, in particular, provide an explicit criterion for the evaluation of learning strategies.

To introduce the relationship between problems and computations, in the next section we begin with the abstract hierarchy developed in theoretical computer science. Our treatment of the hierarchy is very informal, and it is not altogether perfectly suited for questions in computational neuroscience. Nevertheless, it does suggest two notions that will be fundamental for what follows: (i) how can a problem statement be related to a computation; and (ii) given a problem, how can an instance of it be encoded. But this hierarchy is too general, so we then focus on vision, and in particular on early vision, where specific cell properties, functional architecture, and behavioural constraints all exist. This allows us to be more specific, and we eventually arrive at the question: *what is the natural abstraction for computation in cortical columns.* We illustrate one style of analysis that could answer such questions, and a possible solution opens the door to considering the delicacy of cortical dynamics and spike trains. Thus very abstract analysis can suggest very concrete, testable results, a topic that relates back to experimental formulations (see opening paragraph).

2 Problem Abstractions and Computational Abstractions

Intuitively, when one thinks about solving problems by neural computation, one thinks about what to represent and how to use it. "How to" implies a procedure, and procedures run on problem instances. These notions are inter-related, and we now deal with both.

The difficulty of solving a problem can be formalized in terms of algorithms for solving it. To illustrate, imagine a traveling salesman who seeks a tour among the cities in his territory. Knowing the "distance" between each pair of cities, it is natural for the salesman to attempt to minimize his travel. Starting from home, an impetuous salesman might strategize that always driving to the next closest city until all are visited will result in a shortest path. This "greedy" strategy has the advantage that it involves little planning, but it has the disadvantage that it will not necessarily find the minimum (shortest tour). A compulsive salesman might attempt to evaluate the distance for every possible tour, and then sort them to choose the shortest. While this is guaranteed to give the right answer, the combinatorics make it totally impractical. For only hundreds of cities, a computer might have to work for centuries to do a total enumeration! Neural networks provide another approximation (Hopfield and Tank, 1985). Thus "how to"-questions force us to consider limits: How well can the salesman do on the traveling salesman problem, given the amount and type of his resources?

The second issue has to do with the domain over which the problem is defined. Inter-city distances are fundamental to the salesman because transportation abstracts part of his job. Emergent structure at that level allows him to relate intercity distances to minimizing fuel costs, pollution, and guaranteeing that all customers are served. Although this example is artificial, consider how an experimentalist might evaluate the salesman's behavour. The given "experimental data" is a list of cities visited. What is his strategy? Although our context suggests minimizing total distance, the experimentalist might try to establish that the salesman is trying to alternate cities according to size of purchase, so two large orders in consecutive cities do not overload his production staff.

Our problem in building an abstraction for vision is to find, in a manner of speaking, the equivalent of a theory of transportation. This should abstract over images, as transportation abstracts over customers, and in effect it is a theory by which we should be able to calculate the (vision analog to) intercity distances. As with inferences about the salesman, different abstractions are possible, and we shall illustrate two of them. Before doing so, however, we continue the general discussion.

2.1 Complexity of Computations

How well can the salesman do on the traveling salesman problem? For a small number of cities–say, fewer than 5–the difficulty of the problem hardly matters, since the total enumeration can be done easily. Computers could handle maybe 20 or 30 cities brute force, but after that the situation becomes intractable. Clearly the traveling salesman problem is more difficult than, say, counting all of the cities, but how much more difficult. Should the salesman attempt to find an optimal solution to his problem or should he settle for something less?

When we talk about problems, normally we mean a particular instance of a problem; e.g., the traveling salesman problem (TSP) over all McDonald's restaurants in Woods Hole, MA. Complexity theorists seek solutions by general methods that will work for *any* instance of a problem, assuming computing or other resources are available. These general procedures become well specified algorithms when a particular machine model is assumed, such as a deterministic Turing machine or a well-defined programming language. With such a machine model, it is possible to specify the *time complexity function* or the number of steps required to solve the problem. We know from the above example that this will be a function of the particular *instance* of the problem, and the input to the problem specifies the instance. For TSP this input includes the number of cities as well as the distances between pairs of cities.

With this structure we can define classes of problem complexity in terms of time complexity: those problems whose complexity grows no faster than a linear function of the size of the input (such as counting the number of cities) are said to be easy, while those that grow faster than a polynomial function in the size of the input are hard (such as finding the salesman's tour). An important class of hard problems are NP-COMPLETE(Garey and Johnson, 1979), meaning that they can be solved in polynomial time on a "nondeterministic Turing machine" (a machine that can attempt an unbounded number of independent computational streams in parallel), and that they can be reduced to one another. Thus a solution to one problem makes others in this class easy, in the sense that it can be used to solve another in polynomial time. The famous open question in computer science, "does P = NP?" asks whether there exist polynomial solutions for nondeterministic polynomial problems. The traveling salesman problem is NP-complete, which is why heuristics, such as the greedy approach, are attempted for it. Even more difficult decision problems than TSP exist, as do some totally undecidable problems (e.g., the famous "halting problem").

The TSP involves a mix of discrete notions (the set of cities and the roads connecting them), and continuous notions (the distances between cities are numbers). Abstractly the cities can be thought of as nodes in a graph, existing roads as edges in this graph, and distances as continuous weights attached to the edges. The traveling salesman problem was to find a tour through this graph such that the sum of the weights attached to the edges along the tour was minimal. (How such graphs and numbers are encoded is a key part of the representation question.) This mixture of continuous and discrete quantities leads to very deep questions mathematically and computationally (Grotschel et al., 1993); it is important that we keep such distinctions in mind for this paper, because the neural computational literature separates them, with the idea that continuous or analogue computations are multiplications and amplifications of signals, while discrete computations involve bits (Koch, 1997).

A type of parallelism entered through nondeterminism, and the combinatorics derived from the number of tours through the graph. The solution was still obtained in a very general class of machines. In comparison to this, the problems that we face in abstracting visual function are already highly constrained, and parallelism arises in a somewhat different guise. As we review in the next section, primary visual cortex has a special structure, and we believe this structure casts light on the class of problems for which it is naturally suited. As with the TSP, it involves a mix of discrete and continuous structures. Understanding it will lead us to suggest a "cortical columnar machine" and a characterization of the computations it supports. One of the interesting aspects of it is that both continuous and discrete views of it are possible.

2.2 Functional Architecture of Visual Cortex

Visual cortex is organized largely around orientation; that is, around selective responses to local oriented bars. In a classical observation (Hubel and Wiesel, 1977), recordings along a tangential penetration encounter a sequence of cells with regular shifts in orientation preference, while normal penetrations reveal cells with similar orientation and position preferences but different receptive field sizes. Together they define an array of orientation columns, and combined with eye of origin, these columns provide a representation for visual information processing. In effect these columns represent an instance of the cortical columnar machine specialized for problems in vision. Our goal is to untangle the machine from the specific problem encoding.

The widespread appearance of columns is normally explained as a packing problem: since an array of different features is calculated for each point in the retinotopic array, including orientation, scale, eye-of-origin, etc., more than a

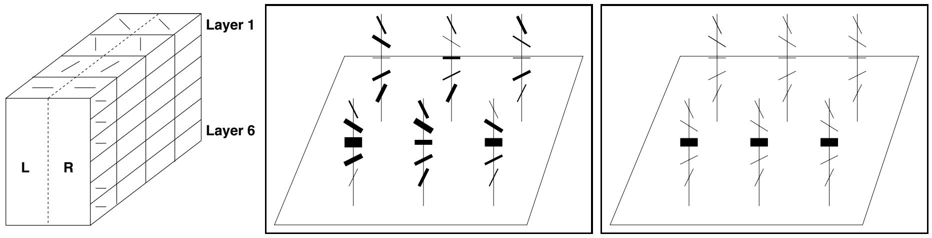

Fig. 1. Two views of the functional architecture of visual cortex. (LEFT) The standard Hubel-Wiesel "ice cube" model, which, although it is a cartoon, expresses the fundamental observation that each local retinotopic area is covered by receptive fields that span a range of orientations, sizes (normal penetration), and eye-of-origin. (MIDDLE) A re-drawing of the ice-cube model, emphasizing a retinotopic array (the tilted plane) and a sampling of upper-layer cells drawn as short segments to show their orientation preference. (The projection is from one eye only; receptive field scale is not shown). As will be developed in the text, when organized in this fashion, a geometric view of processing emerges, in which the fibre of orientations at each position in the retinotopic array abstracts the orientation column, and the arrangement of neighboring fibres suggests an architecture that would support interaction between orientations. The thickness of the orientated bars denotes activity of cells; initial values in time (MIDDLE) and then later (RIGHT) denote processing in time, which may be viewed as a selection process along each fibre. Thus there are "temporal" as well as spatial and orientation dimensions to visual information processing; dynamics are not shown.

3-dimensional arrangement of information is required. If there were only one feature, say orientation, then a columnar architecture would not be required, since this feature could be arranged along the third dimension of cortex, with retinotopic position the other two. (This would simply be a scalar map.) A minimal wiring length constraint then predicts that different features will be clustered, and arranged as closely as possible to their retinotopic coordinates.

It is essential to note that the above argument–multiple superimposed feature maps and minimal wiring length–does not directly address questions of problem encoding, of computation, or of information processing: what are the relevant features, their interactions, and how they should be computed. (Machine models that include communication costs have been developed; (Pippenger, 1993)). A functional abstraction suitable for the domain is required, and if the domain is taken to be images, and statistical regularities over images are sought, then a coding view in which receptive fields model filters arises (Field et al., 1993; Barlow and Blakemore, 1989). Information is processed by composing these filters (e.g., Volterra series), and the layer-to-layer variation with scale amounts to scale-spaces of filters. The computation here is convolution, and linear operators, wavelets, and related abstractions are applied. Channel models are simplified versions of this.

Filters provide an enhancement; they map images into images. For example, lateral inhibitory networks can implement filters for contrast enhancement, and Mach first noticed the importance of this for emphasizing edges. But if such filters are playing a role in edge detection, as is normally supposed, then an additional computational requirement arises: *detection* requires a non-linear decision, a selection process that maps filter outputs into discrete classes, say 1 (to signal that an edge is present at a particular location and with a particular orientation) and 0 (for edge absent). Filtering followed by selection is now a composite computation, and different ways to implement it have been studied (Nowlan and Sejnowski, 1995). This generalizes, in artificial neural networks, to independent "winner-take-all" operations among filter outputs at the same position.

The filtering abstraction leads to emergent notions of "energy" in the response, and the composition of filters followed by selection of, e.g., the maximal energy response can clearly be done in columns. But does the filtering abstraction capture the limits of what can be done in a columnar architecture? Is this edge detection?

A major clue that this is not yet a complete model for columnar computation derives from the interdependence between columns. The anatomical evidence for inter-columnar interactions is classical (Szentagatai), and the physiological evidence is accumulating (Kapadia et al., 1995; Nelson and Frost, 1985; Malach et al., 1993; T'So et al., 1986). This suggests, forcefully, that the decisions at each location are not independent, but are coupled. Decisions are a function of neighbors, and neighbors' decisions are functions of their neighbors. Returning to edge detection, this neighbor interaction is commonly explained as co-aligned receptive field facilitation, although it is not clear how to explain this purely as an emergent property within the filtering abstraction. Rather, it seemed a plausible notion to theorists nearly 20 years ago (Mitchison and Crick, 1982), it is driven by evidence for interactions beyond the classical receptive field and, in particular, it suggests a role for long-range horizontal processes. At first glance the physiology appears to be consistent with this assertion, although not completely.

We now have the outline of a model for edge detection–filtering and selection–that can be implemented and tested on natural images. It is only an outline because the variations in filters, their interactions, and the detection process remain unspecified. Research in computer vision has considered these issues, so it is relevant to examine how their best detectors perform (Canny, 1986). Again, at first glance their performance seems reasonable (Fig. 2), but further consideration questions this. While edge detection seems straightforward, these results illustrate how subtle a notion it really is.

The problem of edge detection is not just philosophical but is constructive: given the component operations (filtering, detection), how should they be put together? Cortical columns arrange the filters for efficent evaluation, but what is the role of scale? How can the results in Fig. 2 be evaluated? The more such questions are asked, the more arise: In which step of the above model–the filtering, the decision, or both–do the problems arise? Canny implemented a type of hysteresis in the decision process, which echoes the notion of co-aligned facilitation in physiology and psychophysics. Does the problem lie in the hysteresis, or in the co-aligned facilitation? Many examples in the literature (e.g., Kapadia et al, 1995; Fig. 10), show facilitation between cells with up to 50 deg orientation differences. Are such exceptions just a physiological "smear" in the connections, or is there more going on? How can these ideas be extended to deal with stereo (recall the ocular dominance bands) or shading? In short, *how can visual information processing be structured on orientation hypercolumns?*

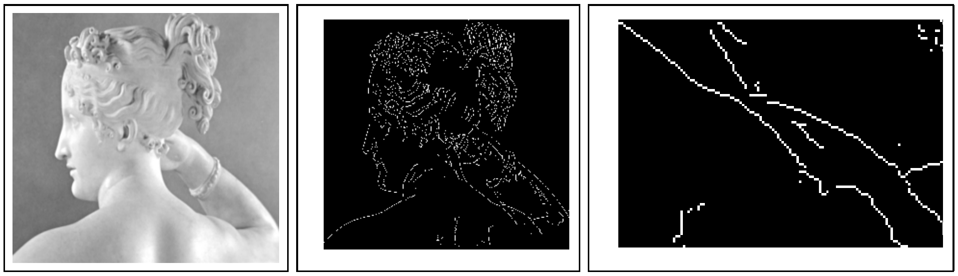

Fig. 2. An illustration of the geometric problems in interpreting standard approaches to edge detection. The edge map (MIDDLE) is obtained from the Canny operator, Matlab implementation, scale=3. This operator is designed with a hysteresis stage, in effect to enforce co-aligned facilitation, and normalization so that the "best" edge orientation can be selected for each position. Thus, it effectively realizes the two basic postulates for filters. While the result seems solid at first glance, a detail from the shoulder/arm region (RIGHT) shows the incorrect topology; neither the detail of the shoulder musculature, nor the separation from the arm, are localized correctly. A more careful examination of the full edge map in (MIDDLE) now confirms the incorrect topology in many locations. Any attempt to follow along the right shoulder would get lost in the arm. We contend that a sufficent approach to edge detection should be topologically correct, which casts doubt on the claim that the Canny operator is doing edge detection.

3 Abstracting the Curve Inference Problem

We are now at the point where we need to build an abstraction for vision that is, in the senses discussed earlier, an analog to transportation. To illustrate a possible approach, we review a computational model that is under development.

We take orientation to be fundamental. Let visual orientation selectivity be a substrate for representing those tangents that approximate the curves that bound objects, that define highlights and other surface markings, and that group into sets of curves to provide texture flows (e.g., hair, fur), and other visual patterns. As suggested above, horizontal interactions between orientations are thought to reduce the errors inherent in locally estimating orientation, so that orientation change can be used to localize corners and discontinuities (as occur at the point where one object occludes another in depth, for example), and for grouping and completing contour fragments obscured by highlights and specularities (Zucker et al., 1989). But how can we derive the connections between tangents: by what rules do they constrain one another? We suggest that differential geometry is the natural abstraction to adopt, because it lies intermediate between images and objects, and because it provides a theoretical level whose constructs emerge as the natural constraints on interaction.

3.1 Curve Inference and Tangent Maps

Local measurements of orientation are inherently ambiguous; while often correct (but coarse), sometimes local intensity structure, or noise, can conspire to affect the maximal local response. To introduce global constraints, we adopt the mathematics of differential geometry, which dictates that any such interactions must involve curvature (Parent and Zucker, 1989). The formal question is how to transport a tangent at one location to a nearby location, and the analysis is not unlike driving a car, in the following sense. At each instant of time the axis of the car defines its (tangent) orientation, and the relationship between the orientation of the car at one instant with that at the next depends on how much the road curves; in operational terms, it depends on how much the steering wheel has to be turned during transport. This requires that curvature must be represented systematically with respect to orientation in cortex, and we (and others) have established that another property of cortical neurons – "endstopping" – is sufficent for achieving this (Dobbins et al., 1987). The majority of superficial, interblob orientation selective cells in V1 are also endstopped to some extent, and these bi-selective dimensions of orientation and endstopping are precisely what is required to represent tangent and curvature. We have used such notions of transport to derive the strength of horizontal interactions (Parent and Zucker, 1989; Zucker et al., 1989), which agree with available data for straight situations (curvature = 0) (Nelson and Frost, 1985; Malach et al., 1993; T'So et al., 1986), but generalize as well to explain data such as (Kapadia et al., 1995); see Fig. 3. Other more recent models (Yen and Finkel, 1997) adopt only the curvature = 0 case, and do not predict the non-co-linear data. Such results also question the interpretation of psychophysical data (e.g., (Field et al., 1993)).

We emphasize that transport and curvature are emergent concepts at the geometric level. They predict a system of interconnections that appears roughly co-linear, with some smearing; this smearing is not noise, but is the result of curvature.

An analysis of the results in Fig. 4 is also possible at the abstract level. Rather than make comparisons with "intuitive" notions of edge, we appeal again to the basic mathematics of the situation. Whitney has classified maps from smooth surfaces into smooth surfaces (Golubitsky and Guillemin, 1973), and has shown that only two situations can occur generically (i.e., without changing under small changes in viewpoint): the fold and the cusp (the position where the fold disappears into the surface); see Fig. 5.

Folds clearly indicate boundaries when viewed from a given position; in fact, the word implies that the tangent plane to the surface "folds" away from the viewer's line of sight. They thus become singular, the two-dimensional tangent space collapses to a 1-D tangent, and this is the object we seek.

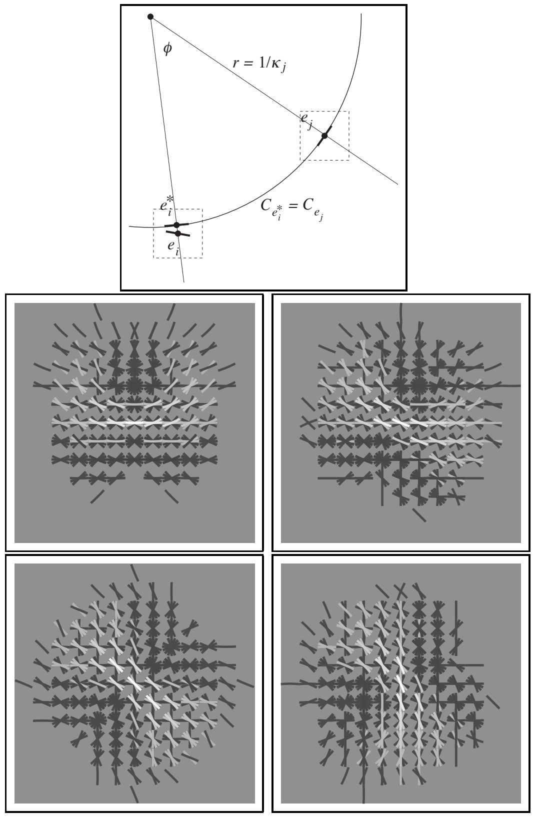

Fig. 3. The geometry of inter-columnar interactions. (TOP) Co-circularity indicates how consistent a neighboring tangent e_j is with a given tangent e_i. Since the positions i and j are close, the actual edge curve can be approximated by its osculating circle, and curvature is approximately constant. The neighboring tangent can be transported along the osculating circle, and the mismatch in position and orientation (between e_i^* and e_i) provides provides a (distance) measure between them. The larger the mismatch, the bigger the distance. Networks can be designed to select those tangents that minimize such distances (Hummel and Zucker, 1983; Miller and Zucker, 1999). (BOTTOM) In network implementations, the transport results are precomputed and embedded in the connections. A very small mismatch results in an excitatory connection, and a larger mismatch in an inhibitory one. Four examples of the compatibilities derived from co-circularity are shown, as follows. Again think of a sampling of neurons, in particular layer II-III pyramidal cells. The bars indicate orientation preference, and all compatibility fields are with respect to the central neuron e_i (shown at maximal brightness). The brightness for each bar (tangent e_j) is the strength of the synapse with the central tangent. Multiple bars at the same position indicate several cells in the same orientation hypercolumn. The connections are intended to model long-range horizontal interactions. Four cases are shown, clockwise from lower-left: co-aligned facilitation ($\theta_i = 45$ deg and $\kappa_i = 0.0$), curved a large amount in the negative sense ($\theta_i = 0$ deg and $\kappa_i = -0.2$), curved a small amount in the negative sense ($\theta_i = 22.5$ deg and $\kappa_i = -0.1$), and curved a small amount in the positive sense ($\theta_i = 67.5$ deg and $\kappa_i = 0.1$). Notice in particular that most of the excitatory connections are between co-aligned cells, given the loose definition of alignment commonly used in the physiological literature (e.g., $\pm 15°$); however, in the high curvature example there are cases of excitatory connections with approx. 50 deg relative orientation. This corresponds with the outlier data from Kapedia et al. discussed earlier.

Fig. 4. Performance of our model for boundary and edge detection. (LEFT) The Canny output at a scale larger than Fig. 2(center). Note how the topological problems remain despite the scale variation. This places doubt on standard scale-space models for edge detection. (MIDDLE) The tangent map obtained from our logical/linear operators (Iverson and Zucker, 1995). Note differences in the edge topology, with the shoulder musculature clearly indicated and proper T-junctions around the neck and chin. Such T-junctions signal orientation discontinuities. (RIGHT) The result of our relaxation process using the co-circularity compatibilities (5 iterations of (Hummel and Zucker, 1983)). Note how the isolated responses through the hair have been removed, and how the details in high curvature regions (such as the ear) have been improved.

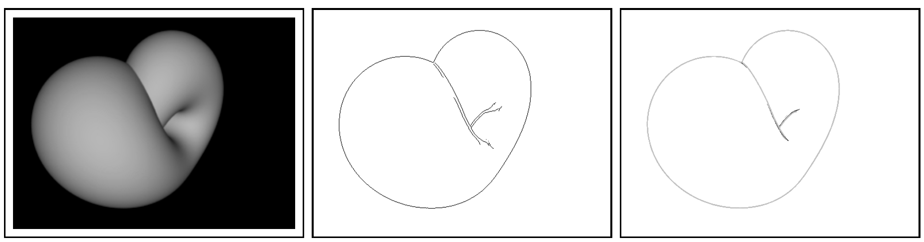

Fig. 5. The motivation from differential topology for early vision. The image of the Klein bottle in (LEFT) shows how "T"-junctions can arise from occlusion relationships (e.g., at the top of the figure), and how certain interior edges can end (e.g., where the fold smoothly joins the body). The Canny edge structure shown in (MIDDLE) is inconsistent with both of these topological observations. Notice how the boundary "T"-junction is not connected, how it smooths the outline, and how the interior folds blur into the shading. In (RIGHT) is shown the output of the logical/linear operator. Notice how the "T"-junctions are maintained, and how the contours end at cusps. Such configurations resemble the shoulder musculature in the Paolina image. Dark tangents signify lines; gray ones edges; see text for a description of how this structure is loaded into the columnar machine.

3.2 The Position-Orientation Representation

The geometry just developed begins to articulate the relationship between problems and representations developed in the introduction. The requirement is a space of tangents at each position, which a redrawing of the classical "ice cube" model elaborates (Fig 1). This can be viewed as a structure "on top of" the image, with retinotopic (x,y) coordinates extended into a third dimension ("height") corresponding to orientation. But the orientation axis is somewhat different than the length axis, because orientation wraps around 2π, or the circle S^1. Thus we speak of this space not as (x,y,z), but as (x,y,θ), where θ is the tangent angle. A point in this space lives in $R^2 \times S^1$. In differential geometry this space is related to the unit tangent bundle.

It is instructive to consider a few examples of how curves in the plane lift into $R^2 \times S^1$, to underline the uses of this encoding within a cortical columnar machine. First, a straight line in the plane lifts to a "horizontal" straight line in $R^2 \times S^1$, whose "height" depends only on the angle θ. A smooth, closed curve in the plane, say an ellipse, lifts into a smooth, closed curve in $R^2 \times S^1$ Discontinuities in orientation lift into broken curves (see Fig. 6).

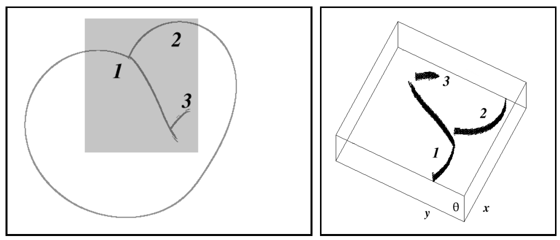

Fig. 6. An illustration of the lift into position (x,y), orientation (θ) space. Three contour fragments from the tangent map in Fig. 5(right) are highlighted. Notice how the discontinuity in orientation at the 1-2 T-junction is separated, highlighting multiple orientations at the same position, the natural columnar representation for orientation discontinuities.

3.3 Encoding the Problem Instance

We have been considering the superficial layers as a kind of geometric machine, and we will be formalizing this shortly. Before doing that, however, we demonstrate another use for the emergent notion of tangent: we can examine the intensity profiles in the tangent and the normal directions separately. Note that linear receptive fields would average these together. We observe immediately that,

- *Normal direction:* The fold condition can take on a different intensity profile for a bounding edge (which involves a dark-to-light transition) from an interior fold (which often involves a light-to-dark-to-light transition, or two edges very close together) or vice versa. This latter profile is often called a line. Standard linear operators, such as those underlying the Canny, can confuse these conditions.
- *Tangential direction:* The definition of a tangent demands that continuity conditions exist (that is, that the limit of one point approaching another must exist). This corresponds to continuity constraints on the intensity pattern.

A necessary condition for a tangent to exist is that the continuity condition be satisfied for both lines and edges, and we have developed a class of non-linear local operators, called logical/linear operators (Iverson and Zucker, 1990), that use Boolean conditions to test whether the above structural criteria are met; if so, they return the average; if not, they veto to zero. "Edge operators" are separated from "line operators", and lines can arise either in light-dark-light conditions (typical of a crack or a crease) or dark-light-dark conditions (typical of a highlight). These non-linearities are different from the compatibility fields above, because they are taking place at much smaller spatial scales. In effect this processing can be used to "load" the orientation and curvature information into the columnar machine. For this we recall the differences in receptive field size from layer to layer in V1, which suggests the following sketch.

- the initial development of orientationally-selective responses at a small scale (layer IVC);
- the development of orientationally-selective responses over a larger scale (layers V and VI); shunting or other inhibition can implement the logical/linear non-linearities (Borg-Graham et al., 1998);
- the combination of the above into orientation/endstopped (or orientation/curvature) co-varying responses; this builds up an orientation column (projection from layer VI up to superficial layers via inhibitory interneuron);
- the refinement of responses in the orientation/curvature column (horizontal interactions in layers II-III) by computing with cliques (discussed next).

Thus the model, in crude terms, is that intra-columnar processing is used to extract initial data for the full columnar machine to process. While it is impossible to know at this time whether the above sketch is correct, notice how the non-linearities, which derive from emergent requirements such as tangent continuity, correspond with layer segregation.

4 The Columnar Machine

The (position, orientation) representation for early vision suggests an abstraction, in loose analogy with the traveling salesman problem. Discrete objects, such as orientations (analogous to cities), are connected via neighbor relations between orientations, loosely analogous to the roads connecting cities in the traveling salesman problem. The continuous distances associated with roads now take two forms: a measure of activity associated with each discrete object (which has no analog in the TSP), and a measure of consistency between objects (lengths of roads, which might be both positive and negative in this generalization). But a fundamental difference arises because the discrete objects are further organized into columns, and conditions can be enforced differently along and between the columns. For example, while two orientations along a column can signal an orientation discontinuity, all orientations at every position should not be allowed. Thus one would want constraints on activity over neighborhoods. The notation to capture these abstractions is shown in Fig. 7.

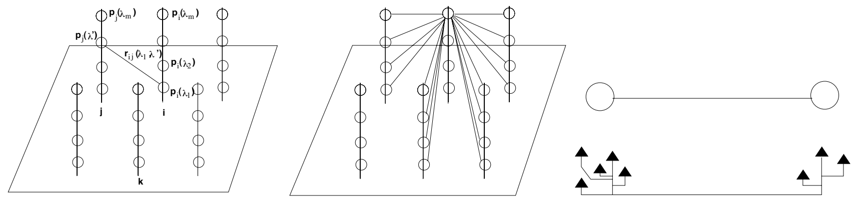

Fig. 7. Deriving a model for columnar computation. It is based on the re-drawn of columns in Fig. 1(right), which are first abstracted into mathematical symbols and then reduced to densely interconnected networks of pyramidal cells. (LEFT) In the abstract orientation columns (Fig. 1), the individual tokens representing a "cell" whose receptive field exhibits a particular orientation preference are represented by a label (λ); a set of such labels is defined for every (columnar) location $i, j, ..., n$. Attached to each label at each position is a number, the probability of label λ at node i. Interactions between labels at neighboring positions are weighted by synaptic coefficents, or compatibilities $r_{i,j}(\lambda, \lambda')$, that capture the influence that λ' at j has on λ at i. For the curve detection example the geometrically derived $r_{i,j}(\lambda, \lambda')$ were shown in Fig. 3. (MIDDLE) Each label at each node has a network of connections to labels at neighboring nodes; these model e.g. the horizontal interactions in layer II-III of V1. (RIGHT) The abstract model is reduced to a network of pyramidal cells, by realizing each $\bigcirc$—$\bigcirc$ abstract complex as a group of cells with rich excitatory interactions. Although only a handful of cells are shown in this drawing, calculations in (Miller and Zucker, 1999) suggest cliques of 30-40 cells are sufficent for orientation resolution at the level of hyperacuity. The polymatrix game abstraction further suggests dynamics in which about 5 spikes in 25 msec. signal the clique that corresponds to the game equilibrium.

We now develop a computational model consistent with this architecture. Although it embodies pieces of the constraint satisfaction and energy models now popular (e. g. Hopfield), it differs from them substantially. It takes form as a game, in a technical sense, which can then be specialized to neural networks in various ways. To illustrate the formal style, this section is slightly more mathematical than previous ones.

There are three steps to the development in this section. First, we introduce the concept of a polymatrix game, and show how this leads to a natural dynamical system. (Analog systems are related to dynamical systems.) We then rewrite the game as a linear complementarity problem, and note discrete (vertex pivoting) algorithms for solving it. Thus it can be related to complexity classes, such as NP-complete, discussed earlier. But most importantly, it is this discrete view that suggests a very different way of obtaining a solution–computation via cliques of neurons–that provides a very rapid answer. Such models should be compared with other attempts to formulate spike-based processing, where the pattern recognition problem has been simplified significantly (Hopfield, 1995)

4.1 Polymatrix Games

An *n-person game* (Nash 1951) is a set of n *players*, each with a set of m *pure strategies*. Player i has a real-valued payoff function $s_i(\lambda_1, \dots, \lambda_n)$ of the pure strategies $\lambda_1, \ldots, \lambda_n$ chosen by the n players. Players can adopt *mixed* strategies, or probability distributions on each player's m pure strategies. A player i's payoff for a mixed strategy is the expected value of i's pure strategy payoff given all players choose according to their mixed strategies. Notice how mixed strategy payoffs are only meaningful in terms of all players' simultaneous actions, which is an important generalization from the filtering/local detection model developed earlier. It captures the interactive component across position implied by the overlaping neuronal connections. In terms of the cortical machine (Fig. 7), you might think of each columnar position as a player, and each label as a pure strategy. $p_i(\lambda)$ is the probability distribution that defines player i's mixed strategy. (We shall consider other games shortly.)

A *competitive* or *Nash equilibrium* is a collection of mixed strategies for each player such that no player can receive a larger expected payoff by changing his/her mixed strategy given the other players stick to their mixed strategies. Nash showed that such equilibria always exist.

A *polymatrix game* is an n-person game in which each payoff to each player i in pure strategy is of the form

$$s_i(\lambda_1, \ldots, \lambda_n) = \sum_j r_{ij}(\lambda_i, \lambda_j)$$

where for all i, $r_{ii}(\lambda_i, \lambda_i') = 0$. We may interpret r_{ij} as i's payoff from j given their respective pure strategies λ_i and λ_j. This implies a payoff to i in mixed strategies of the form

$$\sum_{\lambda_i,\, j,\, \lambda_j} p_i(\lambda_i) r_{ij}(\lambda_i, \lambda_j) p_j(\lambda_j), \tag{1}$$

where $p_i(\lambda_i)$ is the probability player i chooses strategy λ_i.

The quadratic form of (1) can be generalized to include a penalty function in i's payoff (1) which is a convex quadratic function of i's mixed strategy. It is also possibile to include a constant payoff term $c_i(\lambda_i)$ for each strategy λ_i of each player i, regardless of the other players' choices. We make the $c_i(\lambda_i)$ explicit, since these will correspond to bias terms that are important below. Therefore our payoff (1) to i is of the more general form:

$$\frac{1}{2} \sum_{\lambda_i, \hat{\lambda}_i} p_i(\lambda_i) r_{ii}(\lambda_i, \hat{\lambda}_i) p_i(\hat{\lambda}_i) + \sum_{\lambda_i,\, j \neq i,\, \lambda_j} p_i(\lambda_i) r_{ij}(\lambda_i, \lambda_j) p_j(\lambda_j) + \sum_{\lambda_i} c_i(\lambda_i) p_i(\lambda_i) \tag{2}$$

The elements of a polymatrix game can be specified with an $mn \times mn$ consistency matrix R and an mn bias vector c given by

$$R = \begin{bmatrix} \left[r_{11}(\lambda_1, \hat{\lambda}_1)\right] & \cdots & [r_{1n}(\lambda_1, \lambda_n)] \\ \vdots & \ddots & \vdots \\ [r_{n1}(\lambda_n, \lambda_1)] & \cdots & \left[r_{nn}(\lambda_n, \hat{\lambda}_n)\right] \end{bmatrix} c = \begin{bmatrix} [c_1(\lambda_1)] \\ \vdots \\ [c_n(\lambda_n)] \end{bmatrix}. \tag{3}$$

For each player i it will also be convenient to refer to $[c_i(\lambda_i)]$ as c_i, to i's vector of mixed strategies $[p_i(\lambda_i)]$ as p_i, and to the vector of p_i's as p. If we let A be the $n \times mn$ matrix

$$\begin{bmatrix} -1 \ldots -1 \cdots & 0 \ldots 0 \\ \vdots \quad \ddots & \vdots \\ 0 \ldots \quad \cdots & -1 \ldots -1 \end{bmatrix}$$

and let $q^\top$ be the n-vector $(-1, \ldots, -1)$, then p is a vector of all players' mixed strategies if and only if

$$Ap = q, \quad p \geq 0. \tag{4}$$

We express the gradient of (2) as

$$\sum_{j=1}^{n} \left[r_{ij}(\lambda_i, \hat{\lambda}_j) \right] p_j + c_i. \tag{5}$$

Assume p satisfies (4) and is fixed except for player i, whose payoff is given by (2). Since this function is concave, and the constraint set (a simplex) is convex, a given mixed strategy for i will have a maximum payoff if and only if i's gradient (5) has a vanishing projection onto the constraint set (4). Miller and Zucker show that an alternative characterization of the competitive equilibria of the polymatrix game (3) in terms of the equilibria of the dynamical system

$$\begin{aligned} p' &= Rp + c, \\ Ap &= q, \quad p \geq 0. \end{aligned} \tag{6}$$

In other words, these equilibria are precisely the points at which the vector field of (6) vanishes. If R is symmetric then p' is the gradient of

$$1/2\, p^\top R p + c^\top p. \tag{7}$$

The first term in (7) corresponds to the *average local potential* in relaxation labeling (Hummel and Zucker 1983).

Although the above system seems extremely abstract, methods for solving it via gradient descent (when the r_{ij} are symmetric) are classical (Hopfield, 1984; Hummel and Zucker, 1983) Statistical methods, such as simulated annealing (Kirkpatrick et al., 1983), Gibbs sampling, and Markov chain monte carlo, can also be applied. However, such methods can be extremely delicate numerically, and it is difficult to imagine how they can be applied to neurons over short time periods (Nowak and Bullier, 1997). The obvious idea of relating iterations of the gradient descent process to neuronal firing is questionable, because what does "firing rate" mean over such short time periods?

We thus seek different ways of writing this system that permits a proper reduction to biophysical approximations of neurons. Using the Kuhn-Tucker theorem (Kuhn and Tucker, 1951) it can be shown that that p is an equilibrium for (6) if and only if there also exist vectors y, u, v such that p, y, u, v satisfy the system

$$\begin{bmatrix} R & -I_n \\ I_n & 0 \end{bmatrix} \begin{bmatrix} I_n & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} p \\ y \\ u \\ v \end{bmatrix} = \begin{bmatrix} -c \\ e \end{bmatrix} + \delta \begin{bmatrix} -\tilde{c} \\ 0 \end{bmatrix}$$
$$p, y, u, v \geq 0 \tag{8}$$
$$p^\top u + y^\top v = 0$$

Here I_n is the $n \times n$ identity matrix.

The above system of equations is an example of a *linear complementarity problem*, which in general is NP-complete. Identifying this abstract structure immediately opens connections to several important special cases, including linear and convex quadratic programming (Garey and Johnson, 1979), in which it is polynomial. Two-person zero sum games are equivalent to linear programs, and the selection problems discussed earlier are in this special class; some decision problems can be defined as "games against nature", or games against a completely random opponent.

4.2 Computing with Cliques of Neurons

Connections between different problems is not the only advantage to emerge at the abstract level. It also provides insight into possible cortical dynamics which differ fundamentally from gradient descent. Thus we are in a position to capitalize on the different perspectives provided by continuous (analog) and discrete (vertex pivoting) computations.

An important technique for linear complementarity problems is called *Lemke's algorithm* (Cottle et al., 1992), a vertex pivoting algorithm that suggests using a bias to find the solution. While we shall not discuss it in detail, we do return to the bias terms c in (8), and observe that these terms appear with a kind of switch δ indicating whether the bias is on or not.

This bias interpretation can be applied to neurons as follows. Think of neurons as players, and pure strategies as whether a neuron should depolarize (spike) or hyperpolarize (Miller and Zucker, 1992, 1999). Then, if neurons are modelled as piecewise-linear amplifiers, synapses (compatibilities) as conductances, etc., they can be placed in the above form. The key idea behind our model is to consider groups of tightly interconnected excitatory neurons capable of bringing themselves to saturation feedback response following a modest initial afferent bias current, like a match igniting a conflagration (in the phrase of Douglas and Martin, 1992).

The basic computation is in two phases, and builds upon the observed regular spiking behavior of pyramidal cells. Prior to Phase I the cells in a patch of cortex are initially quiescent. Among these hundreds of thousands of cells are several times that number of cortical *cliques*. (Miller and Zucker calculate each clique contains about 33 highly interconnected cells.) A "computation" amounts to activating the cells in a single clique, but no others, to saturation feedback response levels. In Phase I ($\delta = 1$ above): afferent stimulation produces a single spike in a majority of the cells in one clique (whose cells can be distributed among many different iso-orientation areas), as well as a certain number of other cells outside the clique (noise). Because the clique has a sufficient level of excitatory interconnections, all its cells drive themselves to saturation response levels of about 5 spikes in 25 msec (end of Phase I), whereas the initially activated cells outside the clique return to their resting membrane potentials and do not spike further (end of Phase II; $\delta = 0$). Thus the clique has been "retrieved" through a parallel analog computation, and it is this clique that defines the equilibrium point of the system (8).

The realization of a clique of orientationally-selective cells would be a family of receptive fields; one can predict that such a family provides a stable, high-resolution (hyperacuity) representation of short curve segments (Fig. 8).

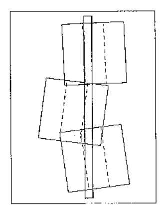

Fig. 8. Distributed representation for a thin line contour derives from a family of receptive fields covering it. Each of these receptive fields comes from a single cortical neuron, and a clique consists of about 33 neurons. In this example receptive fields are represented by rectangles, and a white slit contour stimulus (heavy black outline) excites a highly interconnected clique of simple cortical (S) cells to maximal saturated feedback response by crossing, in the appropriate direction and within a narrow time interval, the edge response region of a sufficiently large proportion of the clique's cells. Biophysical simulations suggest the clique is signalled by about 5 spikes in about 25 msec. Three such receptive fields, out of the approximately 33 required, are illustrated here.

4.3 The Geometry of Stereo Correspondence

The final anatomical simplification was to consider only one eye; we now consider both of the ocular dominance bands. Abstractly this implies an important construction for the columnar machine: the "product" of two machines, one for the left eye and the other for the right eye. Mathematically this suggests working in $(R^2 \times S^1) \times (R^2 \times S^1)$ and compatability fields that also take the product form. We have developed this product structure into an algorithm for computing stereo correspondences (Alibhai and Zucker, 2000) that generalizes the tangent

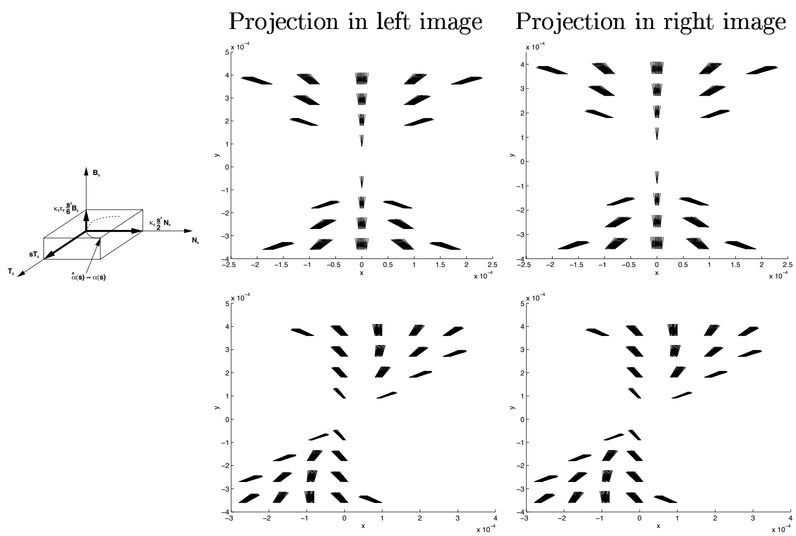

Fig. 9. The local approximation of a curve in three space involves its projection in the left and right images. (LEFT) A curve in three space can be described by the relationships between its tangent, normal and binormal. As the curve moves across depth planes, there exists a positional disparity between the projection of the curve in the left image and the projection in the right image, as well as higher order disparities, for example disparities in orientation. (RIGHT) For the stereo correspondence problem, we are given two edge maps (one for the left camera and one for the right); each of these will be consistent (in the sense that they satisfy the transport constraint); our goal now is to make them consistent with a local approximation to the space curve from which they project. The osculating notion for space curves is a helix. Two examples of positive discrete compatibility fields are shown. Note how they incorporate both position and orientation disparity; this is especially evident in the LOWER pair of compatibility fields.

fields for plane curves to those for general space curves. A curve in three space can be described by the relationships between its tangent, normal and binormal. As the curve moves across depth planes, there exists a positional disparity between the projection of the curve in the left image and the projection in the right image. However, there also exist higher order disparities, for example disparities in orientation, that occur. It is these types of relationships that can be capitalized upon when solving the correspondence problem. Rather than correlating left/right image pairs, we require that there exists a curve in three space whose projection in the left and right image planes is commensurate with the locus of tangent pairs in a neighborhood of the proposed match.

5 Summary and Conclusions

We have argued for the importance of the abstract approach to computational neuroscience, and have shown how structure emerges at both of these levels. For

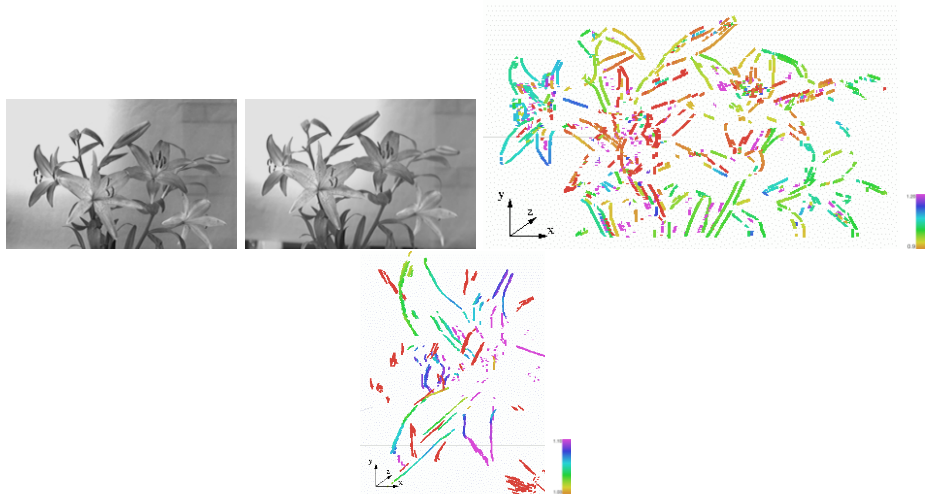

Fig. 10. The depth map associated with a stereo lily pair in (LEFT). The color bars indicate depth scales associated with the image to their left. Each point in the lily image is a tangent in three space that is geometrically consistent with its neighbors. (BOTTOM) Flower detail magnified.

early vision, representational structure emerged regarding tangents, curvatures, and continuity. For computation, linear complementarity emerged as a generalization of columnar operations. Algorithms for solving linear complementarity problems provided new insight into finding fast solutions by neural mechanisms. In all of the above cases it was clear that the resulting networks are quite plausible; but starting only with network components, it seems implausible that all of the above abstract functions would have been inferred.

Analysis at this abstract level is particularly immature, and much remains to be done before truly substantive answers will be found for questions about what "computation" means in the phrase "neural computation". The situation is, in a sense, complementary to the one that Hilbert faced a century ago in mathematics. For him the language and context were clear, and the questions could be posed crisply. In computational neuroscience, the questions are vaguely formulated at best (e.g., what are appropriate models of analog or hybrid computation); perhaps progress will accompany more precise formulations.

Bibliography

S. Alibhai and S. W. Zucker. Contour-based correspondence for stereo. In *Computer Vision - ECCV 2000, Lecture Notes in Computer Science 1842*, June 2000.

J. Allman. *Evolving Brains.* Freeman, 1999.

H. Barlow and C. Blakemore, editors. *Visual Coding and Efficiency.* Cambridge University Press, Cambridge, England, 1989.

L. Borg-Graham, C. Monier, and Y. Fregnac. Visual input evokes transient and strong shunting inhibition in visual cortical neurons. *Nature*, 393:369–373, 1998.

J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

R. W. Cottle, J.-S. Pang, and R. Stone. *The Linear Complementary Problem.* Academic Press, 1992.

A. Dobbins, S. W. Zucker, and M. S. Cynader. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329:438–441, 1987.

D. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: evidence for a local association field. *Vision Research*, 33:173 – 193, 1993.

M. Garey and D. Johnson. *Computers and Intractability.* Freeman, 1979.

M. Golubitsky and V. Guillemin. *Stable Mappings and their Singularities.* Springer-Verlag, 1973.

M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1993.

R. H. Hahnloser, R. Sapreshkar, M. Mahowald, R. Douglas, and H. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–951, 2000.

D. Hansel and H. Sompolinsky. Modeling feature selectivity in local cortical circuits. In C. Koch and I. Segev, editors, *Methods in Neuronal Modeling.* Oxford, 1998.

J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA*, 81: 3088–3092, 1984.

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

J. J. Hopfield and D. W. Tank. “neural” compuation of decisions in optimatization problems. *Biological Cybernetics*, 52:141–152, 1985.

D. H. Hubel and T. N. Wiesel. Functional architecture of macaque monkey visual cortex. *Prod. R. Soc. Lond. B*, 198:1–59, 1977.

R. Hummel and S. W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:267–287, 1983.

L. Iverson and S. W. Zucker. Logical/linear operators for measuring orientation and curvature. Technical Report TR-CIM-90-06, McGill Research Center for Intelligent Machines, Montreal, Canada, July 1990.

L. A. Iverson and S. W. Zucker. Logical/linear operators for image curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1995.

C. Jennings and S. Aamodt. Computational approaches to brain function. *Nature Neuroscience*, 2000.

M. Kapadia, M. Ito, C. Gilbert, and G. Westheimer. Improvement in visual sensitivity by changes in local context: Parallel studies in human observers and in v1 of alert monkeys. *Neuron*, 15:843–856, 1995.

S. Kirkpatrick, J. C. Gellatt, and M. Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, 1983.

C. Koch. Computation and the single neuron. *Nature*, 385:207–210, 1997.

C. Koch. *Biophysics of Computation.* Oxford, 1998.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Second Berkeley symposium on mathematical statistics and probability.* 1951.

T. Lengauer. Vlsi theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol A.*, pages 835–868. Kluwer, 1993.

R. Malach, Y. Amir, M. Harel, and A. Grinvald. Relationship between intrinsic connections and functional architecture revealed by optical imging and in vivo targeted biocytin injections in primate striate cortex. *Proc. Natl. Adad. Sci. (USA)*, 90:10469 – 10473, 1993.

D. Marr. *Vision.* W.H. Freeman, San Fransico, 1982.

D. A. Miller and S. W. Zucker. Efficent simplex-like methods for equilibria of nonsymmetric analog networks. *Neural Computation*, 4:167–190, 1992.

D. A. Miller and S. W. Zucker. Computing with self-excitatory cliques: A model and an application to hyperacuity-scale computation in visual cortex. *Neural Computation*, 11:21–66, 1999.

G. Mitchison and F. Crick. Long axons within the striate cortex: Their distribution, orientation, and patterns of connection. *Proc. Natl. Acad. Sci. (USA)*, 79:3661–3665, 1982.

D. Mumford. Neuronal architectures for pattern-theoretic problems. In C. Koch and J. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 125–152. MIT Press, 1994.

J. Nelson and B. Frost. Intracortical facilitation among co-oriented, co-axially aligned simple cells in cat striate cortex. *Exp. Brain Res.*, 61:54–61, 1985.

L. Nowak and J. Bullier. The timing of information transfer in the visual system. In K. Rockland, J. Kaas, and A. Peters, editors, *Cerebral Cortex, vol. 12*, pages 125–152. Plenum Press, 1997.

S. J. Nowlan and T. J. Sejnowski. A selection model for motion processing in area mt of primages. *J. Neuro.*, 15:1195–1214, 1995.

P. Parent and S. W. Zucker. Trace inference, curvature consistency and curve detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(8): 823–839, August 1989.

N. Pippenger. Communication networks. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol A.*, pages 805–834. Kluwer, 1993.

G. Shepard. *The Synaptic Organization of the Brain*. Oxford, 1990.

D. T'So, C. Gilbert, and T. N. Wiesel. Relationships between horizontal interactions and functional architecture in cat striate cortex as revealed by cross-correlation analysis. *J. Neuroscience*, 6:1160–1170, 1986.

S. Ullman. Sequence seeking and counter streams: a model for bi-directional information flow in the cortex. In C. Koch and J. Davis, editors, *Large-Scale Neuronal Theories of the Brain*, pages 257–270. MIT Press, 1994.

S. C. Yen and L. Finkel. Salient contour extraction by temporal binding in a cortically-based network. *Advances in Neural Information Processing Systems*, 9, 1997.

A. Zador. The basic unit of computation. *Nature Neuroscience*, 2000.

S. W. Zucker, A. Dobbins, and L. Iverson. Two stages of curve detection suggest two styles of visual computation. *Neural Computation*, 1:68–81, 1989.