

Neural Modeling of Flow Rendering Effectiveness

Daniel Pineo*
University of New Hampshire

Colin Ware†
University of New Hampshire

Abstract

It has been previously proposed that understanding the mechanisms of contour perception can provide a theory for why some flow rendering methods allow for better judgments of advection pathways than others. In the present paper we develop this theory through a numerical model of the primary visual cortex of the brain (Visual Area 1) where contour enhancement is understood to occur according to most neurological theories. We apply a two-stage model of contour perception to various visual representations of flow fields evaluated by Laidlaw et al [2001]. In the first stage, contour *enhancement* is modeled based on Li's [1998] cortical model. In the second stage, a model of contour *integration* is proposed designed to support the task of advection path tracing. The model yields insights into the relative strengths of different flow visualization methods for the task of visualizing advection pathways.

CR Categories: H.1.2 [Models and Principles]: User/Machine Systems—Human Factors, human information processing.

Keywords: Flow visualization, contour perception, visual cortex, visualization, perceptual theory.

1 Introduction

Many techniques for 2D flow visualization have been developed and applied. These include grids of little arrows, still the most common for many applications, equally spaced streamlines [Turk and Banks 1996], and line integral convolution (LIC) [Cabral and Leedom 1993]. But which is best and why? Laidlaw et al [2001] showed that the “which is best” question can be answered by means of user studies in which participants are asked to carry out tasks such as tracing advection pathways or finding critical points in the flow field. Ware [2008] proposed that the “why” question may be answered through the application of recent theories of the way contours in the environment are processed in the visual cortex of the brain. But Ware only provided a descriptive sketch with minimal detail and no formal expression. In the present paper, we show, through a numerical simulation of neural processing in the cortex, how the theory predicts which methods will be best for an advection tracing task.

*e-mail: dspineo@comcast.net

†e-mail: cware@ccom.unh.edu

Copyright © 2008 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

APGV 2008, Los Angeles, California, August 9–10, 2008.

© 2008 ACM 978-1-59593-981-4/08/0008 \$5.00

Our basic rationale is as follows: Tracing an advection pathway for a particle dropped in a flow field is a perceptual task that can be carried out with the aid of a visual representation of the flow. The task requires that an individual attempts to trace a continuous contour from some designated starting point in the flow until some terminating condition is realized. This terminating condition might be the edge of the flow field or the crossing of some designated boundary. If we can produce a neurologically plausible model of contour perception then this may be the basis of a rigorous theory of flow visualization efficiency.

The mechanisms of contour perception have been studied by psychologists for at least 80 years, starting with the Gestalt psychologists. A major breakthrough occurred with the work of Hubel and Wiesel [1962; 1968] since which time, neurological theories of contour perception have begun to develop.

In the present paper, we show that a model of neural processing in the visual cortex can be used to predict which flow representation methods will be better. Our model has two stages. The first is a contour enhancement model. Contour enhancement is achieved through lateral connections between nearby local edge detectors. This produces a neural map in which continuous contours have an enhanced representation. The model or cortical processing we chose to apply is adapted from Li [1998]. The second stage is a contour integration model. This represents a higher level cognitive process whereby a pathway is traced.

We apply the model to a set of 2D flow visualization methods that were previously evaluated by humans carrying out an advection pathway prediction task [Laidlaw et al. 2001]. This allows us to carry out a qualitative comparison between the model and how humans actually performed.

Our paper is organized as follows. First we summarize what is known about the cortical processing of contours and introduce Li's model of the cortex. Next we show how a slightly modified version of Li's model differentially enhances various flow rendering methods. Following this we develop a perceptual model of contour tracing and show how it predicts different outcomes for an advection path estimation task based on Laidlaw et al's prior work. Finally we discuss how this work relates to other work that has applied perceptual modeling to data visualization and suggest other uses of the general method.



Figure 1. The neural response to oriented contours and edges can be approximated by means of a Gabor function for many V1 neurons. Three receptive fields of different sizes are shown. Each is excited when a bright horizontal bar falls on the central dark bar. Each is inhibited when the bar falls on the adjacent regions. These neurons are tuned to orientation. Their response pattern weakens as the orientation departs from horizontal.

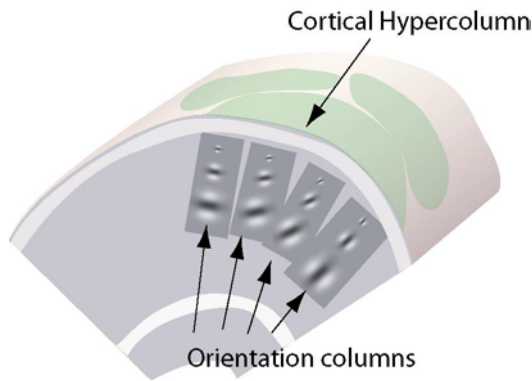


Figure 2. Neurons are arranged in V1 in a column architecture. Neurons in a particular column respond preferentially to different sizes of bar. Moving across the cortex (by a minute amount) yields columns responding to edges having different orientations.

A hypercolumn is a section of cortex that represents a complete set of orientations and sizes for a particular location in space.

2 Cortical Processing of Contours

Visual information passes along the optic nerve from the retina of the eye where it is relayed, via a set of synaptic junctions in the midbrain lateral geniculate nucleus, to the primary visual cortex at the back of the brain (Visual Area 1 or V1). It has been known since the Hubel and Wiesel's work in the 60's that the visual context contains billions of neurons that are sensitive to oriented edges and contours in the light falling on the retina. Such neurons have localized receptive fields each responding to the orientation information contained within the light imaged in a small patch of retina. A widely used mathematical model of a V1 neuron's receptive field is the Gabor function [Daugman 1985]. This is illustrated in Figure 1 and described in detail later. Hubel and Wiesel found that neurons responding to similar orientations were clustered together in a structure they called a "hypercolumn" which extended from the surface of the visual cortex to the white matter. (See Figure 2.) Later researchers discovered that orientation selectivity varied laterally across the cortex. Overall, V1 contains a topographic map of the visual field having the property that every part of the retinal image is processed in parallel for a range of orientation and sizes.

These orientation selective neurons have provided the basis for all subsequent theories of contour and edge detection. There remains the problem of how the output of orientation sensitive neurons, each responding to different parts of a visual contour, becomes combined to represent the whole contour. Part of the solution appears to be a contour enhancement mechanism. Field, Hayes, and Hess [1993] examined the human's ability to perceive a contour composed of discrete oriented elements. They placed a contour composed of separated Gabor patches, among a field of randomly orientated Gabor patches. Contours were detected when the patches were smoothly aligned. They were not detected when there was misalignment. This work suggests that there is some manner of lateral coupling among the visual elements involved in perceiving the Gabor patches in the contour. They and other researchers have suggested that similarly oriented aligned contours mutually excite one another, whereas they inhibit other neurons that are nearby. (Figures 3 and 4).

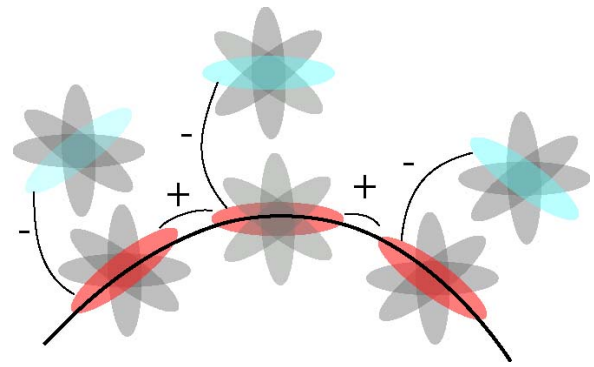


Figure 3. Neurons whose receptive fields are aligned along a continuous contour mutually reinforce each other. They inhibit nearby neurons with a similar orientation sensitivity.

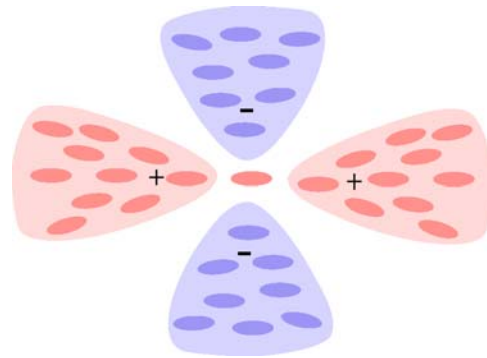


Figure 4. Each neuron has both excitatory and inhibitory regions of influence.

3 Li's Model V1 enhancement

Based on the observed organization of the neurons in the visual cortex by Hubel and Wiesel [1992; 1996], and the experimental evidence by Field, Hayes, and Hess [1993], Zhaoping Li [1998] constructed a simplified model of the behavior of V1 neurons and examined the model's ability to integrate contours across multiple V1 neurons. In Li's model the cortex is approximated by a set of hypercolumns arranged in a hexagonal grid. Each hexagonal cell has 12 orientation selective neuron pairs oriented in 15-degree increments. The mapping of the hexagonal grid to the image space was such that the hex centers were separated by 10 pixels. One of the main simplifications embodied in Li's model is that it fails to incorporate the way the mammalian visual systems scales with respect to the fovea. Real neural architectures have much smaller receptive fields in near the fovea at the center of vision than at the edges of the visual field.

The neurons in each hex cell were grouped into excitatory and inhibitory pairs responding to an edge of a particular orientation at that location. Thus there were a total of 24 neurons per cell. The firing rates of both the inhibitory and excitatory neurons were modeled with real values. The neuron pairs affected neighboring neuron pairs via a transfer function that depended on the alignment of the edge selectivity orientations. Neuron pairs that

ere aligned with one another exhibited an excitatory effect on each other, whereas pairs that were not aligned inhibited each other. Finally, Li's model also contains feedback pathways for higher-level visual areas to influence individual neurons.

3.1 Gabor Response Functions

Li's paper deals with the neural interactions of V1. It does not, however, specify the way the receptive fields of individual neurons process an image. We added this component choosing the Gabor model of V1 receptive fields. The response of an individual neuron field to part of the image is defined by an oriented Gabor function (Figure 1). The Gabor response function is defined by the product of a two-dimensional Gaussian function with a one-dimensional sinusoid:

$$Gabor(x, y, \lambda, \theta, \phi, \sigma, \gamma) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \cos\left(2\pi \frac{x'}{\lambda} + \phi\right)$$

Where,

$$x' = x * \cos(\theta) + y * \sin(\theta) \quad \text{and} \quad y' = -x * \sin(\theta) + y * \cos(\theta)$$

x and y are the coordinates of the function center. Defining the sinusoid, λ is the wavelength of the function, θ is the orientation, and ϕ is the phase. The Gaussian envelope is defined by the standard deviation σ , and γ is its aspect ratio. In our tests, we used $\phi = 0$, causing the Gabor function to respond to lines in the center of the receptive field. We used $\lambda = 21$ pixels and $\sigma = 7$ pixels, producing a Gabor function where the center maximum and the two neighboring minimum are significant, but more distance maxima and minima become negligible. For γ we used a value of 1.

3.3 The Neural Network

The neural network was implemented as described by Li [1998]. Here we give a brief description, but for further details the reader is referred to Li's work. The neural network consists of an array of neuron pairs. Each pair contains one excitatory and one inhibitory neuron, and encodes the presence of an edge in a specific area and orientation in the input image. The encoding is done via the neuron's voltage potential, which is represented by a floating point number.

The neuron values evolve over time according to the set of dynamics described by Li. These dynamics model the lateral connections between nearby neurons up to 10 hex grids away. The excitatory neurons receive additional input from the image within the receptive field corresponding to the neuron. With each iteration, the voltage potential of the excitatory neuron corresponding to location i and orientation θ changes over time by:

$$\frac{d}{dt} x_{i\theta} = -\alpha_x x_{i\theta} - \sum_{\Delta\theta} \psi(\Delta\theta) g_y(y_{i,\theta+\Delta\theta}) + J_0 g_x(x_{i\theta}) + \sum_{j \neq i, \theta} J_{i\theta, j\theta} g_x(X_{j\theta}) + I_{i\theta} + I_0$$

The first term describes the voltage decay of the neuron's value back to zero with a time constant of $1/\alpha_x$, for which we use a value of 1. The second term inhibits edges of similar orientation mapping to the same receptive field. This is done by receiving inputs from the inhibitory neurons of the same receptive field and similar orientations. This behavior is produced by the ψ function, defined by:

$$\psi(\theta) = \begin{cases} 1 & \text{if } \theta = 0 \\ 0.8 & \text{if } |\theta| = 15^\circ \\ 0.7 & \text{if } |\theta| = 30^\circ \\ 0 & \text{otherwise} \end{cases}$$

The activation function g_y defines the signal produced by an inhibitory neuron as a function of voltage potential. Similarly, the function g_x defines the signal produced by an excitatory neuron. It has effect of clamping the excitatory signal to between 0 and 1. These functions are defined by:

$$g_x(x) = \begin{cases} 0 & \text{if } x < 1 \\ (x-1) & \text{if } 1 \leq x \leq 2 \\ 1 & \text{if } x > 2 \end{cases}$$

$$g_y(y) = \begin{cases} 0 & \text{if } y < 0 \\ 0.21 * y & \text{if } 0 \leq y \leq 1.2 \\ 0.21 * 1.2 + 2.5(y-1.2) & \text{if } 1.2 \leq y \end{cases}$$

The third term is an excitatory neuron's feedback to itself. J_0 defines the strength of this feedback loop, for which we used a value of 0.8. The fourth term produces the edge enhancement; it models the excitatory signal to neighboring neurons that lie along the orientation direction in the manner depicted by figure 4. The function J defines the strength of the excitatory connection to a nearby neuron, and is defined by:

$$J_{i\theta, j\theta} = \begin{cases} 0.126 e^{-(\beta/d)^2 - 2(\beta/d)^7 - d^2/90} & \text{if } 0 < d \leq 10 \text{ and } \beta < \pi/2.69 \\ & \text{or } 0 < d \leq 10 \text{ and } \beta < \pi/1.1 \\ & \text{and } |\theta_1| < \pi/5.9 \text{ and } |\theta_2| < \pi/5.9 \\ 0 & \text{otherwise} \end{cases}$$

Where θ_1 is the angle from the neuron's orientation to the line connecting the two neurons, θ_2 is the angle from the neighboring neuron to this line, $\beta = 2|\theta_1| + 2 * \sin(|\theta_1 + \theta_2|)$, and d is the distance separating the neuron and its neighbor. Similarly, the function W defines the strength of the inhibitory connections from nearby neurons, and is defined by:

$$W_{i\theta, j\theta} = \begin{cases} 0 & \text{if } d = 0 \text{ or } d \geq 10 \text{ or } \beta < \pi/11 \\ & \text{or } |\Delta\theta| \geq \pi/3 \text{ or } |\theta_1| < \pi/11.999 \\ 0.14(1 - e^{-0.4(\beta/d)^{1.5}}) e^{-(\Delta\theta/(\pi/4))^{1.5}} & \text{otherwise} \end{cases}$$

The fifth term is the input from the receptive field, calculated using the Gabor function described in the previous section. The sixth term describes a background signal that all excitatory neurons receive, and is set such that average neuron voltage in the network remains constant.

The inhibitory neurons evolve by:

$$\frac{d}{dt} y_{i\theta} = -\alpha_y y_{i\theta} - g_x(x_{i\theta}) + \sum_{j \neq i, \theta} W_{i\theta, j\theta} g_x(X_{j\theta}) + I_C$$

Here the first term again acts to decay the value of the neuron back to zero. The second term is input from the excitatory neuron in the pair. The third term acts to inhibit edges that are of the same direction, but located orthogonally to the edge, in the pattern shown in figure 4. This prevents multiple parallel edges from being produced. The last term is a background signal to all inhibitory neurons.

4. Contour Integration Algorithm

Laidlaw et. al. compared the effectiveness of visualization techniques by presenting test subjects with the task of estimating where a particle placed in the center of a flow field would exit a circle. Six different flow field visualization methods were assessed by comparing the difference between the actual exit numerically calculated and the estimation of the exit by the human subjects. Laidlaw's experiment was carried out on humans, but in our work we apply this evaluation technique to our model of the human visual system and use a contour integration algorithm to estimate path of the particle.

We use the term contour integration to describe the higher level process that must exist for people to judge an advection pathway. We call it contour integration because the task seems to require the user to make a series of judgments, starting at the center, whereby the path of a particle dropped in the center is integrated in a stepwise pattern to the edge of the field.

Perception is a combination of top-down and bottom up processes. Bottom up processes are driven by information on the retina and are what is simulated by Li's model. Top down processes are much more varied and are driven in the brain by activation from regions in the frontal and temporal cortex that are known to be involved in the control of pattern identification and attention [Lund 2001]. All of the flow visualizations evaluated by Laidlaw et al, except for LIC, contain symbolic information regarding the direction of flow along the contour elements (e.g. an arrowhead). In a perpetual/cognitive process this would be regarded as a top-down influence.

Contour integration, and hence path finding, is a combination of top down and bottom up processes. Broadly speaking, top down processes reflect task demands and the bottom up processes reflect environmental information. In our case the bottom up information comes from the different types of visualization while the top down information is an attempt to model the cognitive process of advection pathway tracing.

Algorithm 1: Contour Integration Algorithm

```
CurrentHex = center hex
CurrentDirection = up
Contour = {CurrentHex}
while ( CurrentHex is inside circle ) do
    successors = all grid hexes less than 2 away from
        currentHex and no more than 30 degrees from
        the direction of the CurrentDirection.
    foreach successor in successors do
        successor score = currentHex edge value
        + successor edge value, for the
        orientation connecting the two hexes
    end for
    CurrentHex = successor with greatest score
    CurrentDirection = the orientation connecting the
        new CurrentHex with the previous
        CurrentHex.
    add CurrentHex to the contour
end while
```

Contour integration was modeled using a greedy pathfinding algorithm (Algorithm 1). The algorithm maintains a context that contains a current position and direction. Initially, the position is the center, and the direction set to upward. This context models the higher-order, top-down influence on the algorithm that results from knowledge of the meaning of the center dot and the

directionality indicated by the symbolic information contained by the visualization glyphs. The algorithm traces the contour by repeatedly moving the position to a nearby grid hex, in the approximate direction of the contour, until the position leaves the circle. The nearby hex is chosen to maximize the edge weights in the source and destination hexes, oriented in the direction connecting the two hexes.

5. Results of Contour Enhancement and Integration

Figures 5, 6, 7, 8, 9, and 10 show how the algorithm performed with a sample of the original images used by Laidlaw et al in their prior study. For greater clarity we only show a section of each image although the application of the algorithm to the whole image was computed. In each example the original visualization is shown in the lower left. The top center panel shows the effect of the Li algorithm on the image following 3 feedback iterations. The small bars show how strongly each neuron responds, with redder meaning stronger. The lower right panel shows the path traced out by the contour integration algorithm.

Regular Small Arrows: In the image containing small arrows on a regular grid, we see the bias of the underlying grid upon the chosen path. The chosen path simply travels vertically through the arrows on the starting column. One criticism of the regular grid as a flow visualization technique is the tendency for the underlying grid to become apparent in the visualization, resulting in strong biases in the horizontal, vertical, and 45 degree directions. Here, we see this effect in the contour pathway. We can also see that the responsive fields respond more strongly at the heads of the arrows, thus the contour traces up the right side of the column where the arrowheads are on the right side, and then the contour switches to the left side when the horizontal component of the flow is to the left, resulting in the arrowheads on the left side.

Irregular Small Arrows: While the contour is in the correct general path, it is strongly influenced by the placement of the arrows within the image. The arrows excite a small number of receptive fields, often one or two. This leaves most of the receptive fields without strong excitation. This results in the tendency for the algorithm to be influenced by arrow placement to become even more pronounced.

Irregular Triangles: For the visualization produced using irregular triangles, the path extracted is strongly influenced by the random arrangement of the triangles. Humans also have a tendency to allow such an arrangement to bias their estimation of flow direction, though it can be argued that they can use higher-level reasoning to correct this.

LIC: We can see that the LIC image results in the most uniform neural excitation of all the visualizations. Nevertheless, the algorithm extracts a plausible contour from the neural grid. Like the other visualizations with a random element to them, the contour is influenced by the placement of dark patches within the image. While LIC does not give any visual cues indicating the directionality of the flow, there are often other sources for determining this information. In this case, it can be assumed by comparing to the other visualizations. Thus, we continued to initialize algorithm's directionality in the upward direction, despite the lack of direct visual cues in this visualization.

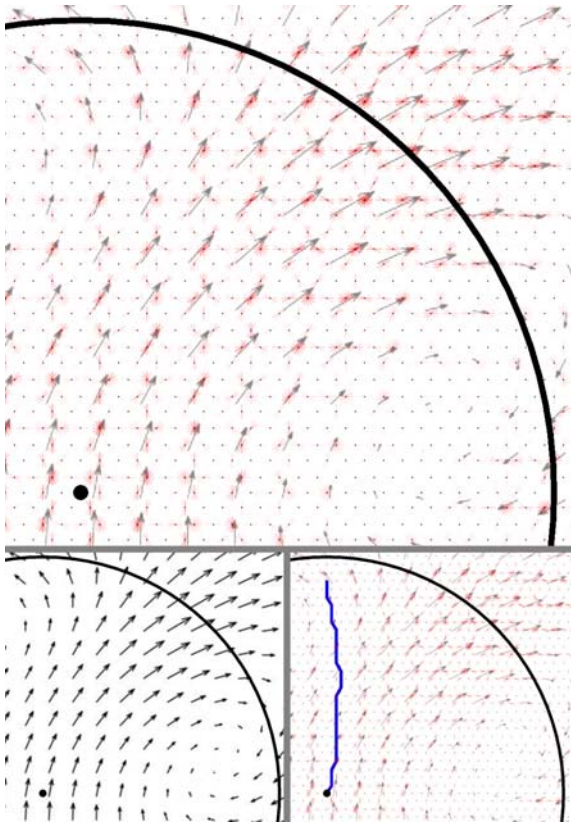


Figure 5: Regular grid of arrows.

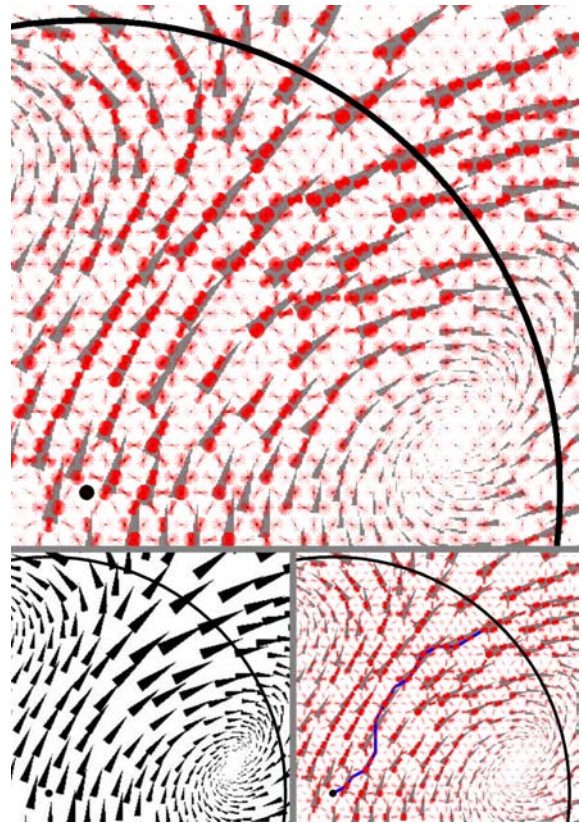


Figure 7: Oriented triangles (Kirby et al, 1999)

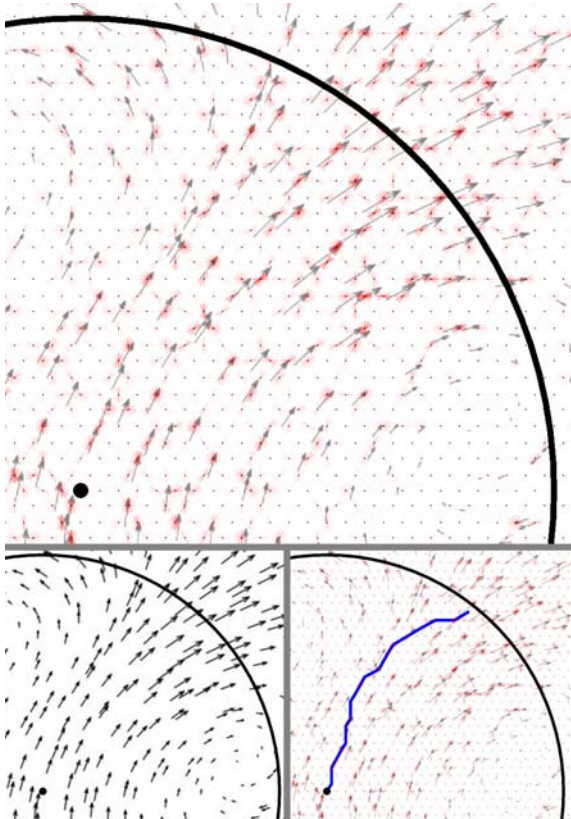


Figure 6: A jittered arrow grid.

Regular Large Curved Arrows: This image contains large, curved arrows arranged on a regular grid. As with the regular small arrows, the underlying grid causes a bias in the horizontal, vertical, and 45 degree directions. The contour produced on this image is strongly influenced by this bias. Initially, the contour follows the regular grid vertically. When it reaches a point where the arrows are oriented at 45 degrees, it departs from the grid and follows the flow until it exits the circle.

Head-to-Tail Arrows: The image with arrows arranged head-to-tail produced the best contour. As Ware argued theory predicts that head to tail placement of arrows should produce best results. The model supports this. The evenly spaced streamlines created by the Jobard and Lefler algorithm provided the best stimulus for coherent chains of excited neurons to develop. The contour simply followed a nearby path of arrows until it exited the circle. One issue, however, is that since the center point is in actually slightly to the side of an arrow path, we know the advection path will remain to the side of the arrow path. Humans correct their advection direction estimates based on this knowledge to increase the accuracy of their estimations. To model this would likely require modeling higher level reasoning centers of the brain, which is out of the scope of this work, but an interesting problem nonetheless.

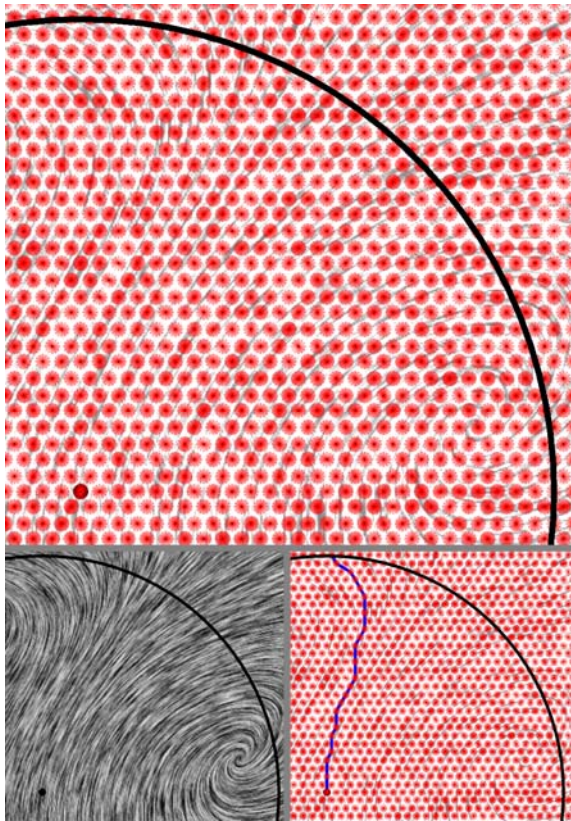


Figure 8: LIC (Cabral and Leedom, 1993)

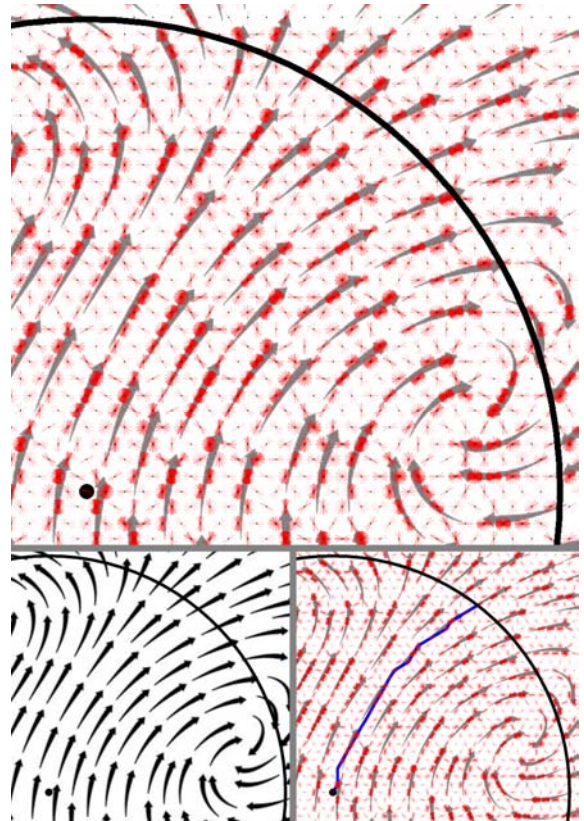


Figure 10: Head-to-tail arrows (Turk and Banks, 1996)

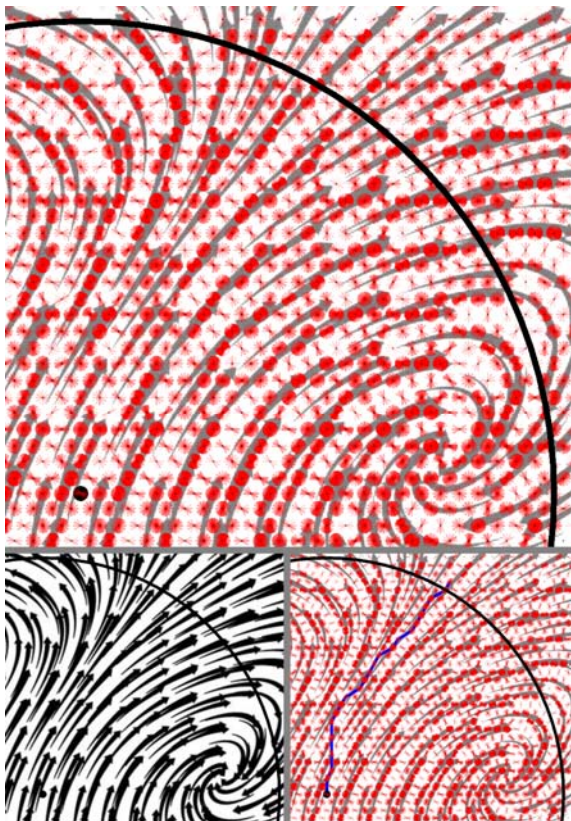


Figure. 9: Curved arrows on a grid. (Turk and Banks, 1996)

6. Discussion

It is encouraging that our results agree qualitatively with Laidaw et al's. The head to tail placement using Turk and Banks algorithm (Figure 10) gave the best result whereas the regular grid of little arrows and LIC methods gave the worst (Figures 5 and 8). The model provides strong support for use of head-to-tail placement of arrows in 2D flow visualization. This Turk and Banks style of visualization produced clear chains of mutually reinforcing neurons along the flow path and this representation should therefore make the flow pathway easy to find.

The model we applied is a considerable simplification over what actually occurs. It only uses the simplest model of the simplest orientation sensitive neurons, and fails to include cortical magnification, among other shortcomings. Real cortical receptive fields are not arranged in a rigid hexagonal grid as they are in Li's model. Furthermore, the neurons of V1 respond to many frequencies, however our model only uses one in its present form. In addition, besides the so-called 'simple' cells modeled by Li, other neurons in V1 and V2 called complex and hypercomplex cells all have important functions. For example, end-stopped cell respond best to a contour that terminates in the receptive field and understanding these may be important in showing how the direction of flow along a contour can be unambiguously shown. Moreover, visual information is processed through several stages following the primary cortex, including V3, V4 and the IT cortex. Each of these appears to abstract more complex, less localized patterns. Researchers are far from having sufficient information to model the operations of these stages all of which may have a role in tracing flow pathlines.

The problem of simulating perception at a higher more cognitive level is even more challenging. We chose not to use existing contour integration models (e.g. Elder et al. [2003]) because these are general purpose edge finding algorithms and do not take into account the specific task of finding advection pathways. It is likely that people adopt different perceptual strategies depending on the visualization. For example, with a regular arrow grid people are likely to rely less on the contour information available in the display, which is weak, and more on strategies, such as looking ahead from a particular arrow, mentally interpolating the flow direction at the look-ahead point and repeating to the edge of the field. Conversely, in the case of the head-to-tail arrows a simpler contour following finding strategy will be more successful.

A complete and accurate simulation of the perceptual process for even what seems to be a quite simple task is probably decades away. Nevertheless, we believe that even simple first order simulations of the kind that we report here can provide important insights into the reasons why certain methods are better than others. Despite the many deficiencies we have outlined having such a model is still better than having no model. By implementing algorithms loose arguments can be carefully examined and properly tested.

The application of a neural simulation of contour perception has the potential to be applied to many visual applications where the perception of form and pattern is critical. We believe that the approach can be fruitfully applied to other application domains including map design and graph layout. In both of these instances it has frequently been observed that “Gestalt” theories of perception may yield insights to the problem (e.g. Ware 2004). However, this observation has never been developed into a rigorous mathematical model. The theory of contour integration we have outlined here could, with minor modifications, be used as a model of the Gestalt concept of continuity and thereby be applied to evaluate design alternatives for many common forms of charts and diagrams.

References

- Blake, R. and Holopigan, K. 1985. Orientation of selectivity in cats and humans assessed by masking. *Vision Research* 23, 1: 1459-1467.
- Cabral, B., and Ledom, L. 1993. Imaging vector fields using line integral convolution. In *Proceedings of ACM SIGGRAPH 93*, 263-272.
- Daugman, J. 1985. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional cortical filters. *Journal of the Optical Society of America*, A/2: 1160-1169.
- Elder, J.; Krupnik, A.; Johnston, L., 2003. Contour grouping with prior models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 6, 661-674.
- Field, D., Hayes, A., and Hess, R. 1993. Contour integration by the human visual system: Evidence for a local “association field”. *Vision Research* 33, 2, 173-193.
- Hubel, D., and Wiesel, T. 1962. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.*, 160, 106-154.
- Hubel, D., and Wiesel, T. 1968. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol*, 195, 215-243.
- Janiszewski, C. 1998. The influence of display characteristics on visual search exploratory search behavior. *Journal of Consumer Research* 25, 290-301.
- Kirby, R., Marmanis, H., and Laidlaw, D. 1999. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings of IEEE Visualization*. 333-340.
- Laidlaw, D., Kirby, R., Davidson, S., Miller, T., Silva, M., Warren, W., and Tarr, M. 2001. Quantitative Comparative Evaluation of 2D Vector Field Visualization Methods. In *Proceedings of IEEE Visualization 2001*, 143-150
- Li, Z. 1998. A Neural Model of Contour integration in the primary visual cortex. *Neural Computation*, 10 903-940.
- Lund, N. 2001. *Attention and Pattern Recognition*, Routledge.
- Singer, W., and Gray, C. 1995. Visual feature integration and the temporal correlation hypothesis. *Annu. Rev. Neuroscience*, 18, 555-586.
- Turk, G. and Banks, D. 1996. Image guided streamline placement. In *Proceedings of ACM SIGGRAPH 96*, 453-460.
- Mostafawy, S. Kermani, O., Lubatschowski, H. 1997. Virtual Eye: Retinal Image Visualization of the Human Eye, *IEEE Computer Graphics and Applications*, vol. 17, no. 1, 8-12
- Ware, C. 2004. *Information Visualization: Perception for Design*. Morgan Kaufman.
- Ware, C. 2008. Toward a perceptual theory of flow visualization. *IEEE Computer Graphics and Applications*, 28(2) 6-11.

