Data Visualization Optimization Computational Modeling of Perception

Daniel Pineo, Colin Ware

Abstract—We present a method for automatically evaluating and optimizing visualizations using a computational model of human vision. The method relies on a neural network simulation of early perceptual processing in the retina and primary visual cortex. The neural activity resulting from viewing an information visualization is simulated and evaluated to produce a metric of visualization effectiveness. Visualization optimization is achieved by applying this effectiveness metric is as the utility function to a hill-climbing algorithm.

We apply this method to the evaluation and optimization of 2D flow visualizations, using two visualization parameterizations: streaklets-based and pixel-based. An emergent property of the streaklet-based optimization is head-to-tail streaklet alignment. It had been previously hypothesized the effectiveness of head-to-tail alignment results from the perceptual processing of the visual system, but this theory had not been computationally modeled. A second optimization using a pixel-based parameterization resulted in a LIC-like result. The implications in terms of the selection of primitives is discussed. We argue that computational models can be used as a method for optimizing complex visualizations. In addition, we argue that they can provide a means of computationally evaluating perceptual theories of visualization, and a method for quality control of display methods.

Index Terms—Information Visualization, Perception, Human Information Processing, Neural Nets, Optimization

1 Introduction

Creating effective data visualizations continues to be more of an art than a science. Data visualization researchers are largely concerned with algorithms to transform data into a graphical form, but when it comes to deciding on the best form they mostly resort to their own aesthetic judgment. Seminal flow visualization papers that introduced widely used flow visualization methods, such those of Cabral and Leedom[1], Jobard and Lefer[2], Turk and Banks [3], and Wijk [4], give no indication of using perceptual theory in their development. Though more recently, Laidlaw et al. [5] carried out a psychophysical study to evaluate a number of flow visualization methods, this too was largely atheoretical. It showed which methods were superior for a defined set of tasks but said little about the reasons for the superiority. Thus, there is clearly a gap between perceptual theory and its application to data visualization, which is likely due to the difficulty in applying the theory, in a descriptive form, to data visualization in practice. This is despite the considerable body of scientific perceptual theory that can be applied to the problem.

The effectiveness of data visualizations can be largely attributed to the powerful processing mechanisms of human vision, with the most effective visualizations mapping data into perceptual mechanisms in such a way that the important patterns are easily perceived convey information [6].

For example, the human visual system contains a perceptual mechanism for quickly detecting visual contours [7]. This mechanism causes the perception of nearby mutually aligned edges to be enhanced, while the perception of nearby edges that are not aligned are inhibited (Figure 1). Successful flow visualizations take advantage of this perceptual mechanism by containing contours tangential to the flow field. We have argued, based on perceptual theory that head-to-tail aligned contours should be better for the advection pathway perception tasks [8] [9]. Nevertheless, although perceptual theory can provide us with intuitions about the best way of conveying certain kinds of data, it is descriptive in nature, and this limits its usefulness in data visualization design [10].

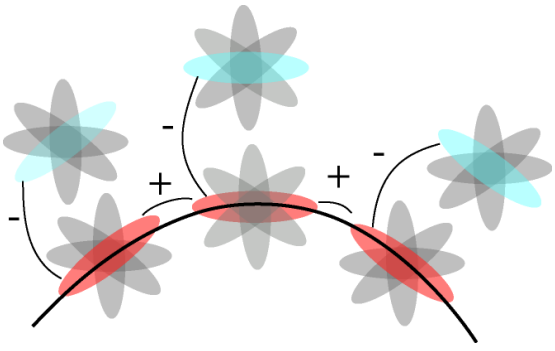

Fig. 1. The perceptual enhancement and inhibition of edges. The perception of nearby aligned edges are enhanced, while the perception of unaligned edges is inhibited.

We can say the same for issues such as the saliency of visual symbols used in displays. Various researchers have argued that we should apply the theory of "pre-attentive" processing to data visualization [11] [8], but exactly how this theory should be applied is only given

in vague descriptive terms. Thus the designer has no way of knowing how easily a particular symbol in a particular context can be seen.

In the present paper we present the case that a computational model of the human visual system can provide a necessary bridge between perceptual theory and its application in data visualization. We use the problem of 2D steady flow visualization as a motivating example throughout.

2 Computational Modeling of Vision

It has been said that computational modeling has now emerged as the third pillar of modern science, a peer alongside theory and physical experiment [12]. The basis for this claim is the enormous impact that computational modeling has had in the scientific disciplines where it has been successfully applied. For example, computational modeling has become an indispensable addition to the field of fluid mechanics allowing the use of simple equations to predict flow patterns around arbitrarily complex surfaces. Closed form algebraic solutions to flow modeling are completely incapable of predicting the complexity of flow around any but the simplest of shapes, but breaking the problem into finite elements and computing the result has allowed the theory to be applied to a diverse range of applications, from weather forecasting to the design of turbine blades.

There are good reasons to believe that the ingredients are present for computational modeling to make a similar contribution to the science of data visualization.

- The primary visual cortex (also known as area V1) of the brain contains a very large number of computational elements all performing the same task in parallel on different parts of an image. This means that high performance parallel processing can be applied.
- V1 has been extensively studied by vision researchers for several decades and its mechanisms are quite well understood.
- V1 is a critical area for pattern perception. It is here that the elements of form are processed. It is also here that the computations are made regarding which parts of an image will be visually salient. Although our understanding of the brain is too limited and computers are still insufficiently powerful to simulate the entire human brain we believe that useful progress can be made by simulating this critical brain area for visual processing.

In a previous paper we showed how a computational model of contour integration in the primary visual cortex could be used to evaluate 2D flow visualization methods [13]. The model produced results that were qualitatively similar to human observers. In the present work we show how a much more powerful computational model, that uses high performance GPU processors can be used in a perceptual *optimization* process.

3 The Neurophysiology of Perception

Neurobiological investigations of the visual system have taught us a great deal about the perceptual mechanisms activated when a visualization is viewed. When a visual image projects onto the retina at the back of the eye, receptors in the eye convert the image into electrical signals. These signals pass through the midbrain lateral genucleate nucleus (LGN), before proceeding to the primary visual cortex (V1) at the posterior of the brain (See figure 2). These early stages of visual processing provide the foundation of pattern perception, color perception, texture and visual salience. Understanding how a visualization is processed in these stages can tell us a lot about how effective it will be. In the following brief review of relevant neurophysiology, we concentrate on aspects of the visual system that are relevant to the modeling effort. Where specific numbers are given describing neurophysiological measurements, like receptive field size, this is because they are later incorporated into the model.

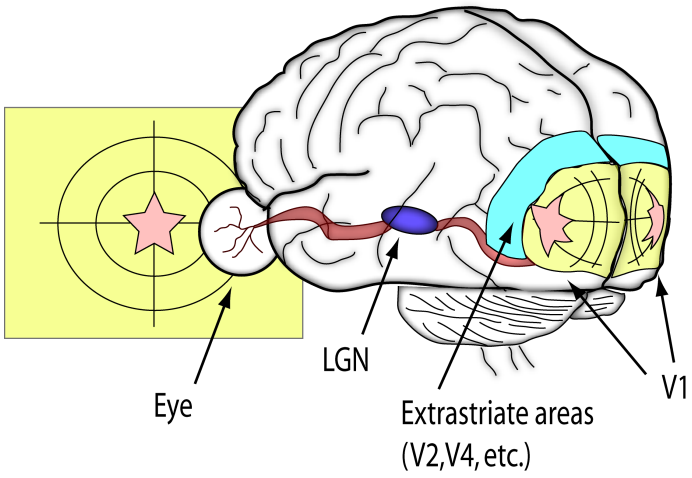

Fig. 2. The early stages of human vision. Light enters the eye and falls on the retina. Photoreceptors in the eye then send signals along the optic nerve, through the LGN, and finally to V1

3.1 Neurons

The most basic building block of the visual system is the neuron. Neurons contain many branches, called dendrites, that allow them to receive signals from other neurons. When a particular pattern of signals is received, the neuron may become excited and begin signaling other neurons along a branch called an axon. Neurons make on the order of 10,000 connections, called synapses, to other neurons. Depending on the neurotransmitter used by the neuron, its signals may have either excitatory or inhibitory effects on the recipient neuron. Signals from an excitatory neuron makes the neurons it signals more likely to become excited, and signals from inhibitory neurons make the neurons it signals less likely to become excited.

3.2 The Eye and Retina

The visual image is projected by the lens of the eye and focused onto the retinal lining at the back of the eye. The projection is such that a visual angle of 1° corresponds to a distance of approximately $288\mu m$ [14] on the retinal surface. The light travels through the layers of the retina and falls upon the light-sensitive photoreceptor neurons. There exist approximately 6 million color-sensitive photoreceptors, known as cones, and approximately 100 million color-insensitive photoreceptors, known as rods. In addition to processing color, cones are the principle photoreceptors responsible for processing high detail. Each cone is subtended by an angle of approximately .006 degrees.

The signals produced by these photoreceptors are then processed by the later stages of the retina. The receptive field surround signal is computed by horizontal [15] and amacrine cells that are connected laterally to nearby areas of the retina. Bipolar cells combine this surround signal with a center signal of an opposing polarity to produce a symmetrical, center-surround receptive field. Bipolar cells exist in both on-sensitive and off-sensitive forms, thus providing both on-center/off-surround and off-center/on-surround signals. Bipolar cells forward this signal to the retinal ganglion cells, which send the center surround signal along the optic nerve, through the LGN, and to the V1.

The central 2° of vision is processed by a $500\mu m$ diameter rod-free area of the retina known as the fovea [16]. It is this area that is responsible for processing fine detail. On average, there are approximately two retinal ganglion cells for each cone in this area, one on-center and one off-center [17]. Away from the fovea, the cone spacing and the ratio of photoreceptors to retinal ganglion cells increases quickly, causing a drop in visual acuity in the periphery.

3.3 V1

V1 is the first cortical processing area of vision, and is largely populated with neurons that are selective to orientated edges. These neurons are capable of detecting these edges at spatial frequencies up to 8 cycles per degree between 2° and 5° eccentricity [18], and presumably higher in the fovea. The receptive fields of V1 neurons are retinotopic, meaning that there is a direct mapping between a neuron's location in V1 and it's receptive field in visual space. One degree of visual angle corresponds to a traversal of 15-20mm across V1 in the fovea, and about 4mm at 2.5° eccentricity. The architecture of V1 can best be understood in terms of columns and hypercolumns. A column is a cortical area of neurons that respond to a specific edge orientation and receptive field location [19] [20] and contain about 200 to 250 neurons [21]. These columns form a repeating pattern across the surface of V1, such that within any $500\mu m$ diameter area columns of all orientations are represented (See figure 3). Such an area is known as a hypercolumn. Both columns and hypercolumns lack well-defined borders in the visual cortex, however, their treatment as discreet, well-defined areas provides a useful approximation for understanding and modeling cortical behavior.

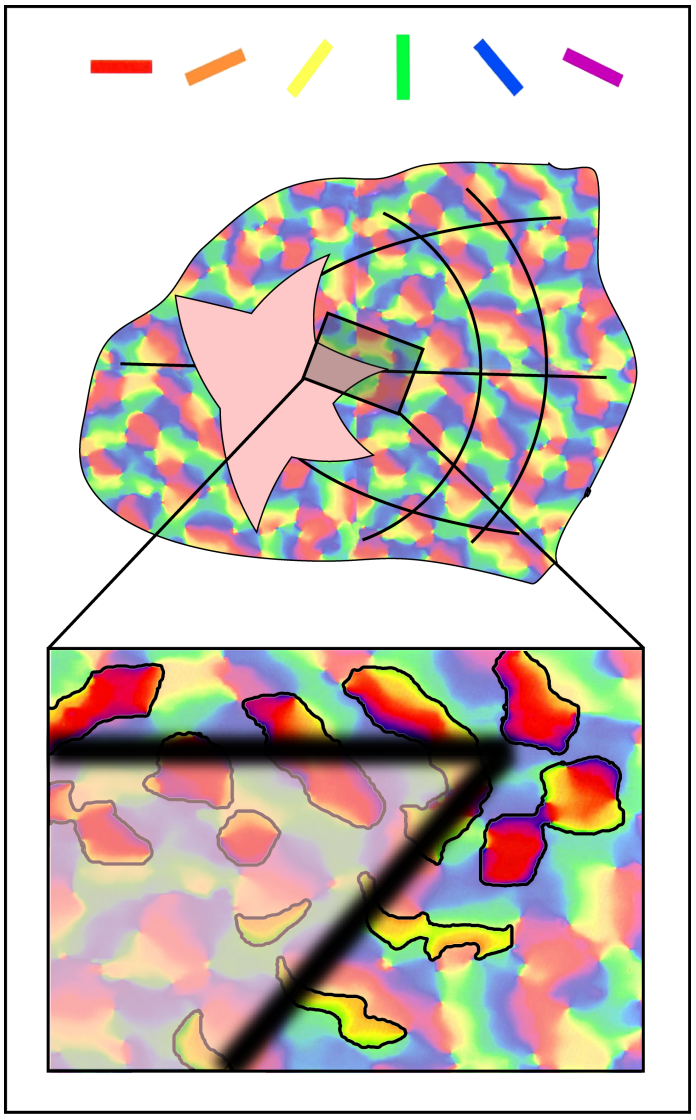

Fig. 3. The repeating columnar architecture of V1. Red patches are cortical columns containing neurons that respond to horizontal edges, yellow patches are columns sensitive to bars angled at 60°. Highlighted areas indicate columns that are active due to the edges of the star viewed in the visual scene.

Furthermore, there is evidence that the perception of an edge is be enhanced by nearby edges with an aligned orientation [7]. Studies of the dendritic trees of V1 neurons suggest that this behavior is achieved via lateral connections, up to 4mm long, between neighboring columns in V1 [22]. V1 is notable for detecting visual feature dimensions including color, orientation, and simple motion. These form the basic primitives that are used in all subsequent processing.

4 RELATED WORK

4.1 Computational models in data visualization

Models of visual perception have been used to study mechanisms such as visual popout, saliency, and clutter in data visualizations. Visual popout is the mechanism of visual perception whereby a conspicuous object in a visual scene can be easily found. Theories of popout suggest that visual features are processed in parallel to create a set of "feature maps" [23] [24]. Each feature map expresses the presence of some visual variable, such as size, orientation, or color, at all points in the visual space. Based on these theories, computational models predict the saliency of visual objects within a scene [25], and such models have been applied to data visualization to predict how a viewer's attention will be directed to relevant areas of a chart visualization [26] [27] [28].

Clutter is a concept closely related to visual popout. Rosenholtz et al. defined clutter as "the state in which excess items, or their representation or organization, lead to a degradation of performance at some task" [29], and offered two computational methods for measuring clutter. The first method is based on a statistical measure of saliency. This method calculates the ensemble of feature vectors in a scene, and derives a measure of clutter from the volume of the feature space spanned by the ensemble. The second method treats clutter from an encoding standpoint, predicting clutter from the subband entropy of the scene.

One notable point is that these methods are based on abstract concepts, such as feature maps, the existence of which has been deduced from psychophysical experimentation. A key difference with our model is that it is based on the physiology of the human visual system.

4.2 Physiologically-based models of perception

Physiologically-based models often idealize neuron behavior using mathematical descriptions of their receptive fields. The center surround response of retinal ganglion cells is often described mathematically using a difference-of-Gaussians (DoG) function (Equation 1) [30]. This function contains a narrow excitatory center, encompassed by a larger inhibitory surround (Figure 4).

$$DoG_{x,y,\sigma_1,\sigma_2} = \alpha_1 G_{x,y,\sigma_1} - \alpha_2 G_{x,y,\sigma_2} \tag{1}$$

Where $G_{x,y,\sigma}$ denotes the normalized 2-dimensional Gaussian kernel,

$$G_{x,y,\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

Psychophysical research has suggested that the visual system has mechanisms to rapidly detect contours. Experiments have shown that humans are capable of identifying a path of visual edge elements within a field of randomly oriented elements [7]. The edge elements used in these experiments were Gabor patches, which are mathematically formed by encapsulating a one-dimensional sinusoid within a two-dimensional Gaussian envelope (Equation 3, Figure 4).

$$Gabor_{x,y,\lambda,\sigma,\theta} = G_{x,y,\sigma} \cos(\frac{2\pi}{\lambda} x) \tag{3}$$

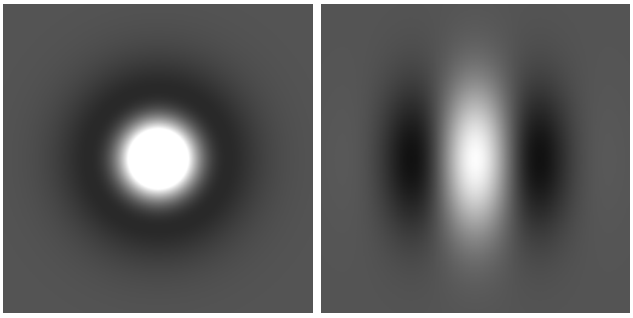

Fig. 4. **Left:** The retinal ganglion difference-of-Gaussians receptive field. This field contains an excitatory center (light) and an inhibitory off-surround (dark). **Right:** The Gabor response function.

This Gabor function is a good approximation of the edge patterns for which V1 neurons are selective [31]. Daugman argued that this function provides an optimal tradeoff between representing spatial position and frequency.

Previous models of the human visual system have produced the contour perception mechanisms important to data visualization perception. Our model incorporates the key elements of these previous works. While these models have been used to explain the emergence of perceptual mechanisms from a neural architecture, they have not been applied to the problem of data visualization.

Li's model [32] of contour perception focuses on the contour enhancement mechanisms of V1. This model does not include retinal processing, or even edge pattern recognition using a Gabor-like pattern matching elements, but instead focuses entirely on V1 lateral connections and how contour integration may emerge from them. Li's model uses a hexagonal array to represent the hypercolumn structure of V1. Each hexagon contains a set of 12 orientation selective columns. Lateral connections cause aligned cortical columns to excite each other, producing enhanced activity in the columns when a contour is viewed. In addition, the model also includes inhibitory connections between columns selective to similar orientations, but offset perpendicular to the orientation. The model includes a feedback path through the inhibitory connections that produce oscillatory behavior that Li proposed carries additional information regarding strength of the contour.

The Grossberg model is much more detailed and physiologically-based [33]. This model includes the center-surround processing of the retina and Gabor-like pattern matching of V1 neurons. It also divides the V1

columns into several layers, which are proposed to produce particular behaviors, including contour enhancement and convergence of neural activity. However, the Grossberg model contains only vertical and horizontally selective columns within each V1 hypercolumn, and uses a regular grid layout.

5 The Computational Model

Our model uses 12 orientation columns per hypercolumn like the Li model, but uses a regular layout like the Grossberg model. We also implement a difference-of-Gaussians retinal response and a V1 Gabor response, similar to the Grossberg model. Most importantly, while the Li and Grossberg models run until feedback produces a steady state, in contrast, our feedback consists of a single lateral excitation stage. The reasons for this are twofold. First, data visualizations are typically viewed in an exploratory manner, thus we seek to model perception in the moments after viewing, before steady-state activity is reached. Second, calculating the neural activity until steady state is reached is computationally time consuming, and is prohibitive in our optimization approach where evaluations must be calculated quickly.

Our perceptual model is based on the known physiology of the human visual system and seeks to produce the behavior found by psychophysical researchers and expressed by perceptual theory. We model the stages of human vision using a multi-layered, partially connected, artificial neural network (See figure 5). The layers of the model consists of a 2D array of either neurons (in the retina), or hypercolumns (in V1).

There exists a quite a bit of variation in the physiological measurements of the human visual system, between studies and even between individuals of a single study. This variation can often be as large as a factor of two or more. Thus, model parameters derived from physiological measurements are approximated, often to the nearest power of two for computational reasons.

The human visual system is known to process scenes at multiple scales [34] [35]. While the visual system is likely to process at a continuum of scales, our model approximates this by processes the visual image at three resolution scales, using powers of two for reasons of computational efficiency. The finest scale uses a 512x512 array, with each element corresponding to .006 degrees of visual space. The two subsequent scales reduce resolution by factors of two to minimize scaling aliasing. Thus the medium scale uses a 256x256 array and the coarsest scale uses a 128x128 array, which correspond to .012 and .024 degrees per element, respectively.

Biological neurons operate by transmitting and receiving voltage spikes, with active neurons being considered those that are transmitting voltage spikes most rapidly. We model this behavior with an overall spiking rate. For example, a rate of one corresponds to a neuron transmitting voltage spikes at its maximum rate. A neuron with a rate of zero corresponds to a biological neuron at a minimal, baseline firing rate. Negative rates are interpreted as a response to an inverse pattern. For example, if a neuron that is selective for a white-center, black surround responds negatively, this is interpreted as a response to a black-center, white surround pattern.

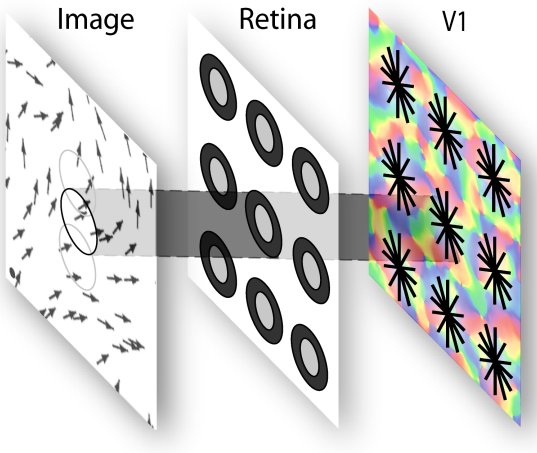

Fig. 5. The neural network architecture. The visual image is first processed at each position by the center surround retinal neurons, then in the second stage by the edge selective neurons in the primary visual cortex (V1).

Neurons send their signals to other nearby neurons within the layer, as well as to neurons in other layers. The connection strengths between neurons is specified by 16x16 kernel arrays in our model.

5.1 The image

The visualization is rendered to a 512x512 RGB video buffer array. The values in this array indicate the light illuminating the cone receptors of the retina. We take the red, green, and blue values to correspond to the intensity of light at the L, M, and S wavelengths. In the model, these values are referred to by I^r, I^g, and I^b, respectively. When viewed on a 133dpi monitor at 140cm, the visual angle of one pixel is approximately equal to one foveal cone. One angle corresponding to 128 pixel elements, or .008 degrees/pixel.

5.2 The retina

The first neural layer of the model, the retinal layer, models the perceptual processing of the visual scene done by the retina. The input to this layer is the illumination, I^r, I^g, and I^b, of the visual scene. The retina computes a center-surround field (figure 4) by combining the center signal of the bipolar cells with a surround signal of the horizontal and amacrine cells, and is done across the chromatic dimensions of red-green, yellow-blue, and black-white. This center-surround signal is output by the retinal ganglion cells of the retina to subsequent stages of the visual system. The center surround receptive field

output of the retinal ganglion cells is defined in the model as a difference-of-Gaussians (Equation 1, Figure 4). The black-white, red-green, and yellow-blue center surround retinal responses are defined as:

$$R^{w-b}_{i,j} = \sum_{x,y} DoG_{x,y,\sigma_1,\sigma_2}(I^r + I^g + I^b)_{i+x,j+y}/3 \quad (4)$$

$$R^{r-g}_{i,j} = \sum_{x,y} DoG_{x,y,\sigma_1,\sigma_2}(I^r - I^g)_{i+x,j+y}/2 \quad (5)$$

$$R^{y-b}_{i,j} = \sum_{x,y} DoG_{x,y,\sigma_1,\sigma_2}(I^r + I^g - 2I^b)_{i+x,j+y}/4 \quad (6)$$

The values of σ_1 and σ_2 specify the size of the center and the surround fields, we use values of 1 and 2 for these parameters, respectively. For the values of α_1 and α_2 we use 1 and .5.

5.3 V1 Edge Detection

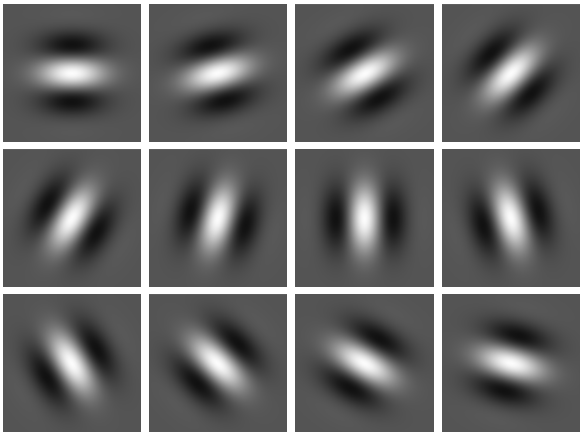

Fig. 6. The V1 Gabor kernels used by the model. Light areas indicate areas that produce an excitatory response, dark areas produce inhibitory responses.

Hypercolumns within V1 component of the model exhibit both edge detection and edge enhancement behavior. As in Li's model, we use 12 columns per hypercolumn, responding to orientations varying by 15° increments (Figure 6). Edge detection results from the pattern of synaptic connections from the retina, defined by:

$$V1_{i,j,\theta} = \sum_{x,y} Gabor_{x,y,\theta} R^{w-b}_{i+x,j+y} \quad (7)$$

We use a value of 7 for λ, which gives a spatial frequency of 8.9 cycles/degree for our medium scale resolution, corresponding to parafoveal processing in the 2° to 5° eccentricity range found by Foster [18]. At the high and low resolution scales this corresponds to 23.8 and 5.9 cycles/degree, respectively. We use σ = 2 to define the Gaussian envelope, which was chosen to encapsulate a single cycle of the sinusoid. x' and y' are found by rotating x and y by θ degrees:

$$[x', y'] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (8)$$

5.4 V1 Edge Enhancement

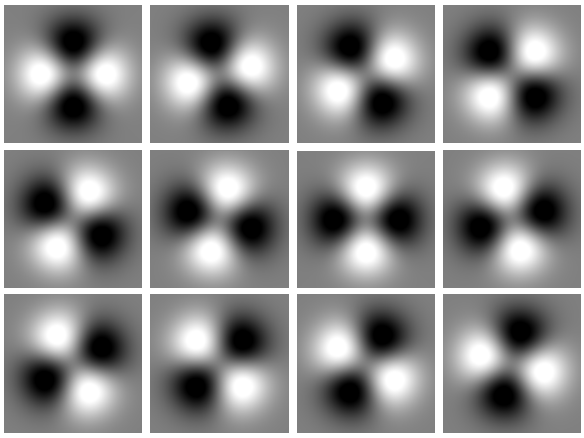

Fig. 7. The V1 edge enhancement kernels used by the model. Excited neurons in the light areas have an excitatory effect, excited neurons in the dark areas have an inhibitory effect.

Columns of V1 exhibit enhanced activity when they correspond to an edge that lies along a continuous contour. The pattern of synaptic connections (Figure 7) used in the model to produces this behavior is defined by:

$$E'_{x,y,\theta} = G_{x,y,\sigma}(x'^2 - y'^2) \quad (9)$$

Yielding the enhanced V1 column activity:

$$V1'_{i,j,\theta} = \sum_{x,y,\theta} E'_{x,y,\theta} V1_{i+x,j+y,\theta} \quad (10)$$

5.5 GPU model implementation

The model was implemented in nVidia's CUDA GPU programming environment, and run on an nVidia Quadro FX 3600M graphics processor. The model required approximately 200ms to run, and consumed nearly 300MB of graphics memory. The neuron layers were stored in GPU memory as arrays of single precision floating point numbers, and operated upon by graphics kernels.

6 Approach to Optimization

Visualizations encode information with visual variables that correspond to the feature dimensions detected by V1. For example, flow visualizations use the orientation visual variable to express vector flow orientation. Our

work begins with the premise that in effective visualizations, the perception of those visual variables closely matches the data being expressed. In visualization where visual variables are interpreted as information, it is clearly desirable that those visual variables are perceived in a manner that accurately reflects the data that is being conveyed. If a viewer's perception of orientation within a flow visualization does not correctly correspond to the flow direction, then the viewer's judgment of the flow direction will likely be impaired.

6.1 Graphical Primitives

A key issue is the choice of the graphical primitives used in the optimization and their parameterization. For example, we might choose to use a grid of arrows and simply optimize the line with and arrow density. However, because arrow grids are known to be poor (ref) even and optimized solution is likely to be more and minimally acceptable. A more sophisticated primitive can take into account prior work in data visualization, suggesting that a particular style of presentation may be more effective. For example, curved streaks that are tangential to the flow pattern and that have more distinct heads than tails are suggested by a number of sources (ref).

The parameterization determines the space of visualizations that will be searched by the optimization algorithm, and it is thus important that it be chosen carefully. An excessively liberal parameterization will result in a large search space, yet an overly restrictive parameterization will cause desirable solutions to fall outside the search space.

To better understand these issues, we chose to investigate our optimization approach with tow kinds of primitives, one very simple: black or white pixels, and a second that used streaklets.

6.1.1 Streaklet Primitives

In the present study, we parameterize the flow visualization in two ways. In the first method, we use a set of graphical streaklet primitives, with each streaklet defined by a center point, a length, and a color. The path of the streaklet is found by advecting from the center point a distance of half the length in both the upstream and downstream directions. Thus we drastically reduce the search space at the cost of not exploring visualizations with graphical elements non-tangential to the flow direction. The width of the streaklets correspond to the color, in our parameterization they vary in appearance from thin-blue to thick-yellow.

6.1.2 Pixel Primitives

In the second parameterization, we define the visualization with an 512x512 array of black and white pixels. This enlarges the search space considerably, allowing the algorithm to search dense flow visualization techniques. We chose to limit parameterization to black and white pixels, rather than greyscale or color, to limit the search space, which even limited to black and white pixels, has a size of 2^{512^2} visualizations.

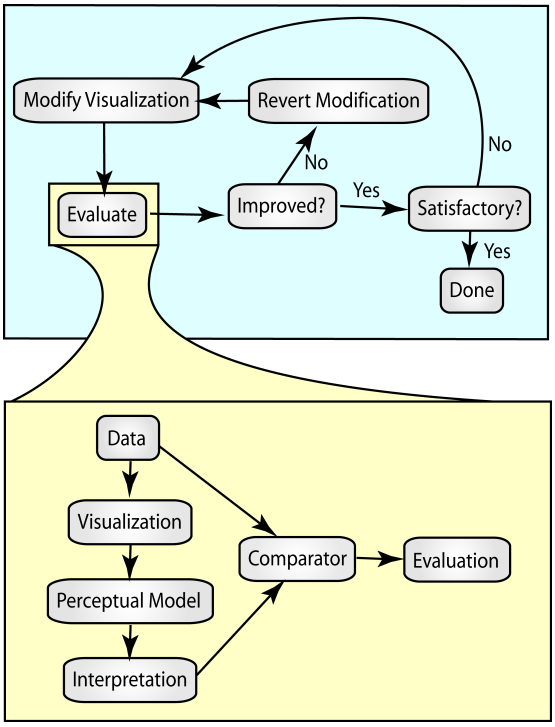

Fig. 8. The data visualization optimization block diagram, showing the hillclimbing algorithm (top) and the evaluation function (bottom).

6.2 The hill climbing algorithm

The visualization optimization is based on the hill climbing search technique, and is described in figure 8 and algorithm 1. The algorithm begins with an arbitrary and potentially poor solution. For the streaklet-based parameterization we chose to start with a blank visualization, and for the pixel-based parameterization we chose to start with random white noise. This choice primarily affected the convergence time of the algorithm, and had little effect on the resulting visualization. The visualization is repeatedly modified slightly, and the modifications are kept only if they result in an improved effectiveness. In applying the algorithm to flow visualization optimization using the streaklet primitives, we modify the flow visualization by randomly adding or removing a streaklet, or by slightly modifying the parameters of a streaklet. For the visualization using pixel primitives, we toggle a random pixel. The process of modifying and keeping improvements continues until some satisfactory solution is reached. While more efficient search strategies may exist, hill climbing's simplicity makes it appropriate for an initial search technique.

6.3 Evaluation metric

The evaluation metric is a measure of how accurately the data visualization conveys data to the viewer. This

Algorithm 1 Hill Climbing Visualization Optimization Algorithm

$Visualization \leftarrow Blank$ or $Random$
while $Visualization$ is not satisfactory **do**
  Randomly add, remove, or modify an element of $Visualization$
  $Effectiveness \leftarrow$ effectiveness of $Visualization$ (Algorithm 2)
  if $Effectiveness > BestEffectiveness$ **then**
    $BestEffectiveness \leftarrow Effectiveness$
  else
    Revert changes to $Visualization$
  end if
end while

is calculated by comparing the data to the perception of that data as predicted by the model. For example, flow visualizations convey flow direction by using perceptual edges tangential to the flow direction. However, it may be possible for non-tangential edges to be perceived in the visualization. This can result, for instance, from the shapes of arrowheads, or by the positioning of nearby streaklets such that they are aligned perpendicularly to the flow. By comparing the vector data with the predicted perception of edges, we can estimate how accurately the data is being perceived by the viewer. The overall evaluation is calculated as:

$$Eval = \alpha OrientationEval + (1-\alpha)SpeedEval \quad (11)$$

The value of α specifies the relative weight of the perception of orientation and speed in the final evaluation. By setting the value of α to one, the evaluation of speed perception can be optionally ignored. Likewise, the evaluation of orientation perception can be ignored by setting α to zero. The $OrientationEval$ and $SpeedEval$ terms are described below.

6.4 High-level perception

Understanding of data visualizations does not occur in the early visual system. The visual patterns detected here are processed by higher-level areas of cognition to yield a conscious understanding of the visualization. In fact, our work rests on the reasonable assumption that the brain produces this high-level understanding from the activity of low-level neurons, and that that erroneous low-level perception has a degrading effect on high-level understanding.

There is a cognitive mapping from low-level perception to high-level understanding. This mapping depends on top-down factors, such as the context and task. In a flow visualization, the meaning of an edge pattern is natural. For example, a horizontal edge is understood as a left-right flow direction. For color, there is no natural mapping, color can indicate any variable or value. These must be inferred from context, such as presence of a color key or previous experience. In our data visualization evaluation, we use the natural mapping for orientation, and chose an arbitrary mapping for color: blue-yellow mapping to speed. It is assumed that this cognitive mapping of color is provided to the viewer via a color key.

6.5 Orientation perception

To allow the analysis of orientation selective neurons in our model, we translate the neural activity into a vector representation. Such a vector representation allows the overall orientation of a set of neurons to be calculated via a vector sum. While an orientation can be naturally represented as a vector aligned with the orientation, and is uniquely defined if constrained to the positive y direction, this creates complications when performing analysis that incorporates averages or sums of orientations, because such calculations result in a strong bias in the positive-y direction. Instead, we translate to a vector with double the angle between the orientation angle and the positive x-axis, producing angles that range from 0 to 2π. The orientation vector is calculated as:

$$\vec{O}_{i,j,\theta} = \begin{bmatrix} V1'_{i,j,\theta}\cos(2\theta) \\ V1'_{i,j,\theta}\sin(2\theta) \end{bmatrix} \quad (12)$$

From the activation of the edge sensitive neurons in the model, we predict the perception of orientation, $\vec{O}'_{x,y}$, produced by the data visualization. The predicted orientation at a point in visual space is calculated from the activity of edge selective neurons with receptive fields near the point being predicted. Nearby neural activities are converted to vectors in orientation space and summed, weighted by their distance to the visual point being predicted.

$$\vec{O'}_{i,j,\theta} = \sum_{x,y,\theta} G_{x,y}\vec{O}_{i+x,j+y,\theta} \quad (13)$$

To evaluate the predicted perceived orientation at a point in the visualization, we calculate the projection of the predicted orientation perception vector onto a normalized orientation vector calculated from the data. This yields values that are largest when the predicted and actual orientations are closely aligned, and smallest when unaligned. The overall orientation evaluation is found by summing the projection over all points.

$$OrientationEval = \sum_{i,j} \vec{O}'_{i,j} \cdot \frac{\vec{Actual}_{i,j}}{|\vec{Actual}_{i,j}|} \quad (14)$$

6.6 Speed perception

Data visualizations frequently use color to encode a data variable, in flow visualizations this is typically flow speed. In such cases, the perception of speed may be included by using a non-zero value of α in equation 11. Perceived colors must be interpreted as data values

with the aid of some mapping from colors to values. The perceived speed is calculated from the activity of blue-yellow neurons, weighted by distance of the receptive field to the point being predicted. We use a linear mapping, with saturation, for the interpretation from colors to speed:

$$S_{i,j} = min(max(\alpha \sum_{x,y} G_{x,y} R^{y-b}_{i+x,j+y} + \beta, 0), 1) \quad (15)$$

where α and β define the mapping between the colors and the data values. As such, these values are dependent on the data set used. To evaluate the speed, we sum the absolute difference between the predicted perceived speed and the actual speed:

$$SpeedEval = -\sum_{i,j} |S_{i,j} - |\vec{Actual}_{i,j}|| \quad (16)$$

This produces evaluations that are largest when the predicted speed closely matches the actual speed, and smallest when there is a large difference.

7 RESULTS

Figure 9 shows a streaklet-based visualization optimized by the algorithm for orientation perception ($\alpha = 1$, thus ignoring speed evaluation). The data was created from a 5x5 uniform grid of randomly orientated, normalized vectors. First, we notice that the streaklets are spaced out much like the Jobard & Lefer algorithm (figure 10). This occurs because as the spacing between streaklets is reduced, there is an increased likelihood of a edge non-tangential to the flow being perceived between nearby streaklets. A balance is thus reached between having streaklets spaced too close, resulting in perceived non-tangential edges, and spaced too sparsely, resulting in a visualization of insufficient coverage to adequately convey the data.

However, it's worth noting that unlike the Jobard & Lefer algorithm, this spacing is not explicitly specified. The spacing is instead produced as an emergent property because it is found by the algorithm to be perceptually optimal. One consequence of this is that the spacing is not a strict rule, and does not produced if it is not predicted to be perceptually optimal by the model. We see this in the figure in areas where the flow field diverges or converges. The spacing of the streaklets that develop elsewhere in the visualization do not emerge in these areas, producing a clear display these converging and diverging areas.

It has been conjectured that the effectiveness of aligning streaklets in a head-to-tail fashion, as in the Jobard & Lefer algorithm, is a result of the early perceptual processing of the visual system. To find such streaklet alignment produced as an emergent property of our perceptual optimization method lends support to this conjecture.

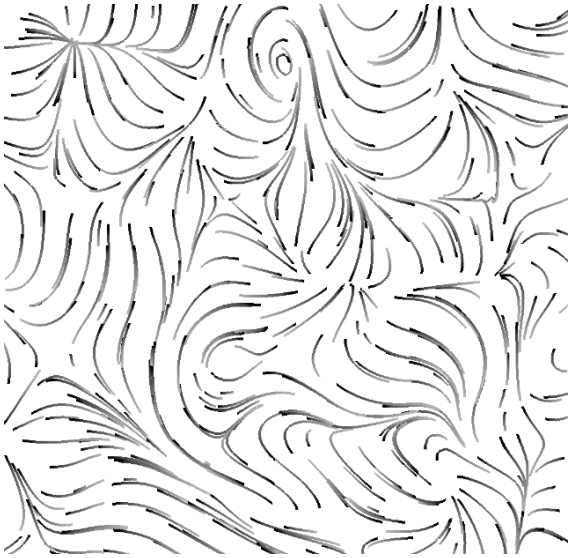

Fig. 9. Flow visualization optimized by our algorithm using a streaklet parameterization, note the emergent head to tail alignment.

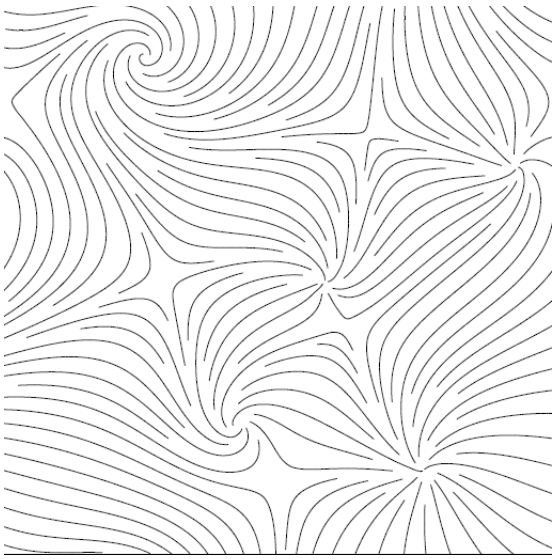

Fig. 10. A flow visualization created by the Jobard & Lefer algorithm (From Jobard & Lefer, 1997)

Figure 11 demonstrates the algorithm on ocean flow data from the U.S. Navy Operational Global Ocean Model (NCOM). The visualized region shows the Gulf Stream current passing between Florida and Cuba. Using $\alpha = .5$, the algorithm optimized for both orientation and speed perception. In this case, color-coding was used to represent flow speed. The resulting visualization clearly shows the strong gulf stream current flowing from the bottom left to the top right.

Using the pixel-based parameterization (Figure 12), we see convergence to a visualization solution that more closely resembles those generated by the line integral convolution (LIC) technique of Cabral and Ledom [1] (Figure 13). We do, however, notice some important

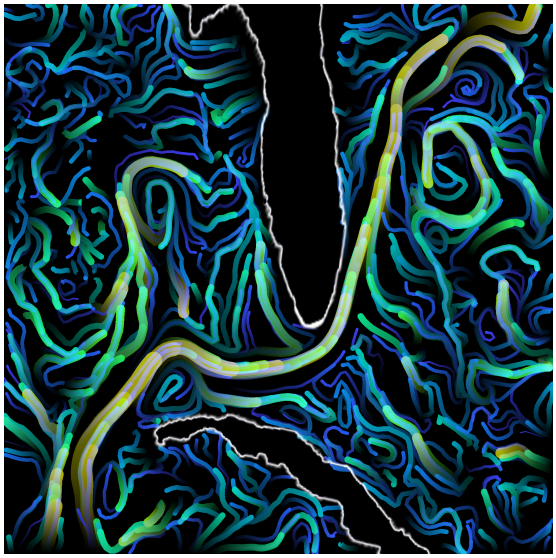

Fig. 11. A gulf stream visualization produced by our technique optimizing the perception of speed from color.

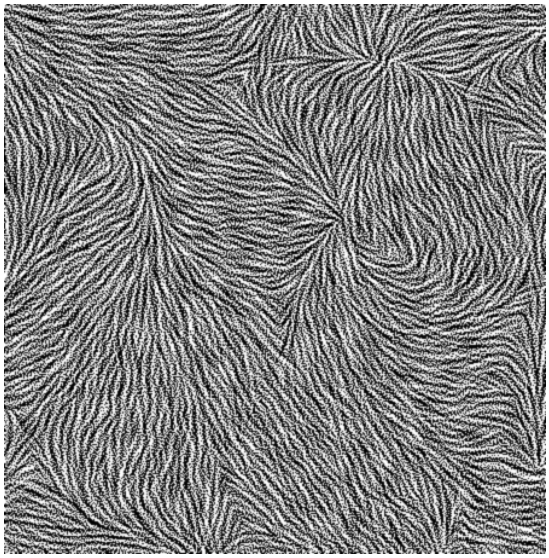

Fig. 12. The flow visualization produced by our technique using a pixel parameterization

differences. There is a clear difference in the spatial frequency of the texture in the direction perpendicular to the flow direction. The frequency of the LIC visualization results from the underlying noise texture being convoluted. In figure 13, the underlying figure is a white noise texture, which results in a frequency on the order of one pixel. This frequency can be adjusted by altering the convoluted texture, but bears no direct relation to the perceptual properties of vision. The perpendicular wavelength generated by our technique is approximately 7 pixels, produced from our choice of λ for the Gabor kernels of V1.

One notable limitation of our technique, with respect to LIC, is the ability to visualize areas of high curvature. In such areas we see the orderly arrangement of the visualization break down and produce sharp discontinuities result. We also see the effects of the large search space resulting from the pixel-based parameterization. In particular, the pixel-based parameterization did not produce extended contours or the spacing observed from the streaklet parameterization. This is due to the large number of pixels that must be toggled to move a pixel-based contour to a more optimal position. Moving a contour requires many steps, and the hill-climbing optimization requires that each intermediate visualization have an improved evaluation.

We found that for the streaklet-based parameterization the search required several minutes to converge on a satisfactory visualization, using an nVidia Quadro FX 3600M graphics card. The pixel-based parameterization required several hours.

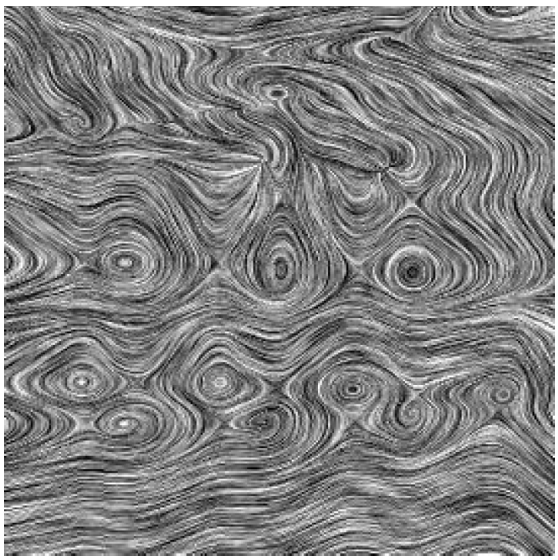

Fig. 13. A flow visualization created by the Line Integral Convolution method (From Cabral & Ledom 1993)

8 Discussion

We have shown how by using a computational approach, perceptual theory can be applied to data visualization to produce tangible results. In doing so, we address a fundamental question of data visualization: What does it mean to be a good data visualization? In the context of this work, we give our answer: A good data visualization is one that triggers neural activity that encodes the desired information in the brain.

In this context, one can view data visualization as a communications problem. In communications systems, source data is encoded, transmitted across a medium, and then decoded at the destination. The quality of such a system lies in its ability to quickly and faithfully transmit data from the source to the destination. Viewed from this perspective, visualization can be considered the encoding stage of this system. The quality of this system as a whole depends not only on this encoding, but on the decoding done by the human visual system.

While we would not argue that the visualizations we have generated thus far are the best that have been produced for this type of data, we believe that they are at least quite good and certainly better than the arrow grids that appear in many scientific publications and on the web sites of modeling groups. Furthermore, it is encouraging to consider that there is still significant room for improvement using this technique. As we better understand the functioning of the visual system produced by its neurobiology, and more accurately model human visual mechanisms, the perceptions predicted by the model will become more accurate. Because this prediction is used to search for the optimal data visualization, this will translate to an improvement in data visualizations produced by our method.

An advantage of this approach is that it has the potential to find solutions not considered by designers. For instance, the overriding the streaklet spacing in areas of high convergence or divergence seems obvious in retrospect, but it was not an expected result.

We find it significant that the optimization algorithm converged upon visualizations similar to the popular Jobard & Lefer algorithm. This supports our assumption that effective visualization requires an accurate perception of visual variables used to convey data, and the approach in general.

Obviously, the solutions that were produced were highly constrained by the graphical primitives we used. This suggests that the key to successful application lies in the adoption of primitives that reduce the dimensionality of the search space, but not so much that the best solutions are excluded. Clearly, there is scope for further investigations here. The method does not have to be applied using streaklet or pixel primitives, it could potentially be applied to any parameterized graphical primitives.

There is clear room for improvements in the search algorithm. We use as simple stochastic hill climbing technique, which randomly makes alterations to the visualization. Improvements could be made by making those alteration preferentially in areas where the model has calculated high perceptual error. Furthermore, a well known limitation of hill-climbing search algorithms is the propensity to find local maxima in the search space. While more sophisticated search algorithms may yield visualizations that are calculated to be even more perceptually accurate, we found that the simple hill-climbing approach produced satisfactory solutions in good time.

Human factors experiments are often used to evaluate different displays in order to determine if they can meet operational requirements [36]. However, a complex display can involve the use of dozens of different symbols, on backgrounds consisting of different textures and colors, together with linear features that use line styles and colors. Using human factors experiments to ensure that all symbols and lines are clearly visible against all possible background combinations is usually not possible since the number of conditions is an exponential function of the number of factors to be considered leading to an experiment that is too large to be carried out. Our system is capable of executing an experimental "trial" several times a second and it can be run 24 hours a day. It can easily execute a million trials in a single day.

Compare this to a human factors experiment where each trial lasts five seconds and a typical session is less than an hour. We may succeed in running 5 subjects a day giving 4000 trials a day although in our experience this is optimistic. Thus, the computer method is 250 times faster. This means that methods that would be infeasible using human subjects become practical.

We have demonstrated that the method can be used in a hill climbing optimization process. Other uses are quality control for complex displays. It may be possible to use our method to test a large set of symbols against a large set of backgrounds and, by using our evaluation technique, determine which are visually compatible. Due to the speed with which we can perform an evaluation, we can also imagine an interactive design assistant. Such a system could provide realtime feedback to a designer as he or she works to create a visualization.

REFERENCES

[1] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," in *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1993, pp. 263–270. [Online]. Available: http://dx.doi.org/10.1145/166117.166151

[2] B. Jobard and W. Lefer, "Creating evenly-spaced streamlines of arbitrary density," in *Eurographics Workshop*. Springer Verlag, 1997, pp. 43–56.

[3] G. Turk and D. Banks, "Image-guided streamline placement," in *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1996, pp. 453–460.

[4] J. J. van Wijk, "Spot noise texture synthesis for data visualization," in *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1991, pp. 309–318.

[5] D. H. Laidlaw, J. S. Davidson, T. S. Miller, M. da Silva, R. M. Kirby, W. H. Warren, and M. Tarr, "Quantitative comparative evaluation of 2d vector field visualization methods," in *VIS '01: Proceedings of the conference on Visualization '01*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 143–150.

[6] C. Ware, *Information Visualization: Perception for Design, 2nd Ed*. Morgan Kaufman, 2004.

[7] D. J. Field, A. Hayes, and R. F. Hess, "Contour integration by the human visual system: Evidence for a local "association field"," *Vision Research*, vol. 33, no. 2, pp. 173 – 193, 1993.

[8] C. Ware, "3d contour perception for flow visualization," in *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization*. New York, NY, USA: ACM, 2006, pp. 101–106.

[9] D. Pineo and C. Ware, "Neural modeling of flow rendering effectiveness," in *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization*. New York, NY, USA: ACM, 2008, pp. 171–178.

[10] H. Rushmeier, H. Barrett, P. Rheingans, S. Uselton, and A. Watson, "Perceptual measures for effective visualizations," in *Proceedings of the 8th conference on Visualization'97*. IEEE Computer Society Press, 1997, pp. 515–517.

[11] C. Healey, K. Booth, and J. Enns, "Harnessing preattentive processes for multivariate data visualization," pp. 107–107, 1993.

[12] President's Information Technology Advisory Committee, "Computational Science: Ensuring America's Competitiveness," *Computational Science Subcommittee (PITAC), National Coordination Office for Information Technology Research and Development*, p. 117, 2005.

[13] D. Pineo and C. Ware, "Neural modeling of flow rendering effectiveness," in *ACM Transactions on Applied Perception*, vol. 7, no. 3, 2010.
[14] N. Drasdo and C. Fowler, "Non-linear projection of the retinal image in a wide-angle schematic eye." *British Medical Journal*, vol. 58, no. 8, p. 709, 1974.
[15] K. Naka, "Neuronal circuitry in the catfish retina," *Invest. Ophthalmol*, vol. 15, no. 11, pp. 926–935, 1976.
[16] S. Polyak, *The retina*. Chicago: University of Chicago Press, 1941.
[17] S. Schein, "Anatomy of macaque fovea and spatial densities of neurons in foveal representation," *The Journal of Comparative Neurology*, vol. 269, no. 4, pp. 479–505, 1988.
[18] K. Foster, J. Gaska, M. Nagler, and D. Pollen, "Spatial and temporal frequency selectivity of neurones in visual cortical areas v1 and v2 of the macaque monkey." *The Journal of physiology*, vol. 365, no. 1, p. 331, 1985.
[19] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J Physiol*, vol. 160, no. 1, pp. 106–154, 1962. [Online]. Available: http://jp.physoc.org
[20] ——, "Receptive fields and functional architecture of monkey striate cortex," *J Physiol*, vol. 195, no. 1, pp. 215–243, 1968. [Online]. Available: http://jp.physoc.org/cgi/content/abstract/195/1/215
[21] V. Mountcastle, "The columnar organization of the neocortex," *Brain*, vol. 120, no. 4, pp. 701–722, 1997. [Online]. Available: http://brain.oxfordjournals.org/cgi/content/abstract/120/4/701
[22] W. H. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick, "Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex," *J. Neurosci.*, vol. 17, no. 6, pp. 2112–2127, 1997. [Online]. Available: http://www.jneurosci.org/cgi/content/abstract/17/6/2112
[23] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97 – 136, 1980. [Online]. Available: http://www.sciencedirect.com/science/article/B6WCR-4D6RJM2-46/2/b34aa05384b2a7702189c22840489174
[24] J. M. Wolfe, K. R. Cave, and S. L. Franzel, "Guided search: An alternative to the feature integration model for visual search," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 15, no. 3, pp. 419 – 433, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/B6X08-46X8WS6-1/2/501dfc15c55b7f804b0900725ef0d027
[25] L. Itti and C. Koch, "Computational modelling of visual attention." *Nat Rev Neurosci*, vol. 2, no. 3, pp. 194–203, March 2001. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/11256080
[26] S. Fabrikant and K. Goldsberry, "Thematic relevance and perceptual salience of dynamic geovisualization displays," in *Proceedings of 22nd ICA international cartographic conference: mapping approaches into a changing world*, July 2005.
[27] S. Garlandini and S. Fabrikant, "Evaluating the effectiveness and efficiency of visual variables for geographic information visualization," *Spatial Information Theory*, pp. 195–211.
[28] S. Fabrikant, S. Hespanha, and M. Hegarty, "Cognitively inspired and perceptually salient graphic displays for efficient spatial inference making," *Annals of the Association of American Geographers*, vol. 100, no. 1, pp. 13–29, January 2010.
[29] R. Rosenholtz, Y. Li, and L. Nakano, "Measuring visual clutter," *Journal of Vision*, vol. 7, no. 2, pp. 1–22, 2007. [Online]. Available: http://journalofvision.org/7/2/17/
[30] R. Rodieck, "Quantitative analysis of cat retinal ganglion cell response to visual stimuli," *Vision Research*, vol. 5, no. 12, pp. 583–601, 1965.
[31] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 2, no. 7, pp. 1160–1169, 1985.
[32] Z. Li, "A neural model of contour integration in the primary visual cortex," *Neural Comput.*, vol. 10, no. 4, pp. 903–940, 1998.
[33] S. Grossberg and J. Williamson, "A neural model of how horizontal and interlaminar connections of visual cortex develop into adult circuits that carry out perceptual grouping and learning," *Cerebral Cortex*, vol. 11, no. 1, p. 37, 2001.
[34] H. Wilson, D. McFarlane, and G. Phillips, "Spatial frequency tuning of orientation selective units estimated by oblique masking," *Vision Research*, vol. 23, no. 9, pp. 873–882, 1983.
[35] D. Marr, E. Hildreth, and T. Poggio, "Evidence for a fifth, smaller channel in early human vision," 1979.
[36] C. Wickens, J. Lee, Y. Liu, and S. Becker, *An introduction to human factors engineering*. Pearson Prentice Hall Upper Saddle River, NJ, 2004.