

The centroid paradigm: Quantifying feature-based attention in terms of attention filters

Peng Sun^{1,2} · Charles Chubb¹ · Charles E. Wright¹ · George Sperling¹

Published online: 9 November 2015
© The Psychonomic Society, Inc. 2015

Abstract This paper elaborates a recent conceptualization of feature-based attention in terms of attention filters (Drew et al., *Journal of Vision*, 10(10:20), 1–16, 2010) into a general purpose centroid-estimation paradigm for studying feature-based attention. An attention filter is a brain process, initiated by a participant in the context of a task requiring feature-based attention, which operates broadly across space to modulate the relative effectiveness with which different features in the retinal input influence performance. This paper describes an empirical method for quantitatively measuring attention filters. The method uses a “statistical summary representation” (SSR) task in which the participant strives to mouse-click the centroid of a briefly flashed cloud composed of items of different types (e.g., dots of different luminances or sizes), weighting some types of items more strongly than others. In different attention conditions, the target weights for different item types in the centroid task are varied. The actual weights exerted on the participant’s responses by different item types in any given attention condition

are derived by simple linear regression. Because, on each trial, the centroid paradigm obtains information about the relative effectiveness of all the features in the display, both target and distractor features, and because the participant’s response is a continuous variable in each of two dimensions (versus a simple binary choice as in most previous paradigms), it is remarkably powerful. The number of trials required to estimate an attention filter is an order of magnitude fewer than the number required to investigate much simpler concepts in typical psychophysical attention paradigms.

Keywords Selective attention · Methodology · Centroid estimation · Statistical summary representations (SSRs) · Feature-based attention · Attention filter

Introduction

When looking for the vine that contains the most ripe red berries among vines that contain less ripe ones, we are selecting a color feature distributed broadly across space and aggregating that information to come to a useful conclusion. This ability to carry out tasks that prioritize information carried by a particular visual feature has enormous evolutionary value, and a great deal of empirical evidence confirms our intuition that we indeed can do this (Ball & Sekuler, 1981; Baldassi & Verghese, 2005; Davis & Graham, 1981; Haenny et al., 1988; Ho et al., 2012; Lankheet & Verstraten, 1995; Ling et al., 2009; Liu et al., 2007; Martinez-Trujillo & Treue, 2004; Maunsell et al., 1991; Muller et al., 2006; Saenz et al., 2003; Serences & Boynton, 2007; Shih & Sperling, 1996; Treue & Martinez-Trujillo, 1999). Many studies support the specific claim that such “feature-based attention” (FBA) does indeed modulate sensitivity broadly across space (Felisberti & Zanker, 2005;

✉ Peng Sun
peng.sun@nyu.edu

Charles Chubb
c.chubb@uci.edu

Charles E. Wright
cewright@uci.edu

George Sperling
sperling@uci.edu

¹ Department of Cognitive Sciences, University of California, Irvine, CA 92697-5100, USA

² Department of Psychology, New York University, New York, NY 10003, USA

Liu et al., 2007; Rossi & Paradiso, 1995; Saenz et al., 2003; Hayden & Gallant, 2005; McAdams & Maunsell, 2000; Motter, 1994; Saenz et al., 2002; Seidemann & Newsome, 1999; Treue, 2001), exerting influence even at locations that are irrelevant to the participant's task (Arman et al., 2006; Liu & Mance, 2011; Serences & Boynton, 2007; White & Carrasco, 2011; Zhang & Luck, 2009). In addition, imaging studies have shown that attention deployed broadly across space for a particular feature modulates the gain of cortical regions selective for the attended feature (Kamitani & Tong, 2005; Liu et al., 2003; Liu & Mance, 2011; O'Craven et al., 1997; Muller et al., 2006; Schoenfeld et al., 2007; Serences & Boynton, 2007; Serences et al., 2009). Ling et al. (2009) measured Threshold-vs.-Noise curves (Lu and Doshier, 1998) in a global motion task to provide evidence that FBA increases sensitivity to a target motion direction both by increasing the gain and also by sharpening the tuning of the MT neuron population response for motion direction. These studies confirm the existence of FBA as a human capability and sketch in bold strokes how FBA operates to control behavior. In addition, they provide important insights into where in the brain FBA is implemented and how the brain might operate to accomplish FBA.

The current paper introduces methods for addressing the next conceptual plane of questions concerning the behavioral goals achieved by FBA. Any given FBA-deployment aims to heighten sensitivity to a specific body of target information (e.g., information carried by the red elements of a scene), and much of the research cited above confirms that a given FBA-deployment can indeed alter sensitivity broadly across space to information carried by different features in the visual input. These observations, however, leave open many important questions central among which are:

1. How effective is the FBA-deployment in sensitizing the participant to the target information?
2. Is the FBA-deployment sensitive to information other than the target information? If so,
 - (a) which non-target features influence the FBA-deployment? and
 - (b) how exactly do they influence it?

These questions led us to conceptualize FBA in terms of *attention filters* (Drew et al., 2010). An attention filter is a process, initiated by a participant in the context of a task requiring FBA, which operates broadly across space to modulate the relative effectiveness with which different features in the retinal input influence performance.

The main purpose of the current paper is to describe the development of the centroid method for measuring the attention filter achieved by a particular deployment of FBA. The paradigm enables one to describe precisely (1) the relative amounts by which the attention filter passes each to-be-attended feature and rejects each to-be-ignored feature, and (2) the attention filter's overall sensitivity to the

information in the stimulus relative to the noise compromising performance. For a given set of stimulus items, by varying the instructions to the participant to induce different FBA-deployments in different attention conditions, one can iterate the centroid method to discover the structure of the entire space of attention filters achievable by human vision for that set of stimulus items.

Summary statistics and the centroid paradigm

Much recent work has focused on the ability of human participants to extract "summary statistics" from brief displays of ensembles of items. For example, substantial research now supports the claim that human participants are adept at extracting the mean size of an ensemble of disks (Ariely 2001; Chong & Treisman 2003, 2005a, b). Other work has focused on the effectiveness with which human participants can estimate the mean orientation of an ensemble of items (Dakin, 2001; Solomon, 2010).

As emphasized by Alvarez and Oliva (2008), the centroid is another summary statistic that human participants are adept at extracting from an ensemble of items. This paper shows how to analyze the attention filters that a human participant can interpose between a briefly-presented ensemble of items and the computation that he/she uses to extract the centroid.

The centroid paradigm, first used by Drew et al. (2010) and substantially refined here, offers a number of important advantages over previous methods used to measure attention filters (Chubb and Nam, 2000; Nam & Chubb, 2000; Chubb & Talevich, 2002), including the following:

1. It is much more efficient than these previous psychophysical choice paradigms, requiring many fewer trials to estimate the attention filter deployed by a participant in a given attention condition.
2. Fitting the data is surprisingly simple.
3. Refinements in training methods and in stimulus constraints simplify the summary computations and make the centroid paradigm more precise and resistant to artifacts.

Matlab code for analyzing the data from centroid experiments is provided in Appendix 4.

The centroid paradigm—overview

Imagine a large, flat, weightless piece of hard plastic upon which are placed a number of different stacks of pennies of different heights at different locations. The *centroid* (or center of gravity) of such a spatial array of penny-piles is the average location of all the pennies in the array. If a fulcrum were placed directly under the centroid of the penny-pile

array, the plastic sheet and the penny-piles on top of it would balance perfectly.

The paradigm described in this paper enables one to measure the attention filters that participants can achieve in estimating the centroids of clouds of items drawn from a given set $Types$ of item types. The first example application, which is described in the following two sections to illustrate the formal descriptions of the method, uses the set of vertically oriented Gabors shown in the upper row of Fig. 1; the second example application (the Dots experiment), uses eight dots with Weber contrasts $-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$ shown in the bottom row of Fig. 1. In other potential applications, the set $Types$ might contain (1) dots of different colors, (2) Gabors of different orientations, (3) Gabors of different spatial frequencies, (4) Gabors varying in both spatial frequency and orientation, (5) line segments of different lengths and orientations, (6) small objects of different shapes, etc. There is no requirement that the items in $Types$ be ordered (or even related) in any way; however, many applications of interest use items equally spaced along a single continuum (as in the example experiments).

To appreciate the basic idea behind the method, imagine a simulated experiment in which, on each trial

1. A stimulus is presented consisting of a spatially random array comprising several items of each of the item types in $Types$, and
2. A response is produced as follows:
 - (a) An unknown filter f , constant from trial to trial, is applied to the stimulus to create a map in which each item i of a given type $\tau_i \in Types$ in the stimulus field is replaced by a pile of pennies of size $f(\tau_i)$, and finally
 - (b) The centroid of the filtered stimulus is extracted.

Although this may not be obvious, the unknown filter f can easily be derived from the data from such an experiment; the section entitled “Analyzing the data: Estimating f_ϕ ” explains how.

In the experiments described below, the participant is asked in different attention conditions to try to weight the different item types in accordance with various different “target filters,” ϕ . Each of these different attention conditions requires an experiment analogous to the simulated experiment described above to measure the attention filter f_ϕ that the participant actually manages to achieve.

The Gabor pattern experiment

To make the presentation of the paradigm for estimating attention filters more concrete, we illustrate it with an example experiment based on the Gabor patterns shown in the top

row of Fig. 1. These eight Gabor patterns were the $Types$ in this example experiment. They were identical in form but differed in contrast. Each was 25×25 pixels. The space constant of the Gaussian window was 5 pixels in each of the horizontal and vertical dimensions, and the windowed sinusoid was vertical, had phase $-\frac{\pi}{2}$ relative to the center of the envelope, and had a period of 13 pixels. A single Gabor pattern in the stimulus subtended 0.51 deg. The principal spatial frequency of the Gabors was 3.51/cpd. Gabor contrasts were $\frac{k}{8}$ for $k = 1, 2, \dots, 8$. The 512×512 pixel region in which the stimuli were displayed subtended 10.51 deg. of visual angle at the viewing distance of 1m. The luminance of the homogeneous background was 52.1 cd/m^2 .

Participants were the first two authors plus two naive participants who had never before participated in any psychophysical experiments. The methods used in all experiments were approved by the UC Irvine Institutional Review Board, and the participants provided signed consent forms.

Typically, a participant will be tested in a number of different attention conditions. Each attention condition is defined by a *target filter* ϕ that assigns nonnegative weights to different item types; the target filter ϕ is used to give feedback in that condition. In this experiment, five attention conditions were investigated: in the *Uniform* attention condition, the target filter gave equal weight to all eight of the Gabor patterns; in the *Graded* attention condition, the target filter weighted each Gabor pattern in proportion to its contrast; in the *Inverse-graded* attention condition, the target filter weighted each Gabor pattern in inverse proportion to its contrast; in the *Lowest-only (Highest-only)* attention condition, the filter gave all its weight to the minimum (maximum) contrast Gabor pattern. As will be seen later, some attention conditions are more difficult than others.

In an attention condition with a target filter ϕ , the participant strives to weight different item types in accordance with ϕ ; usually, however, he/she is unable to do so perfectly. He/she gives too much weight to some item types and too little to others. The function that gives the weights exerted on the participant’s responses by different item types in $Types$ is called the participant’s *attention filter* f_ϕ ; the subscript ϕ keeps track of the target filter that yielded this particular attention filter.

An experimental trial

Defining the attention-weighted centroid

A particular stimulus (e.g., Fig. 2b and g) consists of N_{stim} items. Each item $i = 1, 2, \dots, N_{stim}$ is of a particular type $\tau_i \in Types$, and occurs at a location (x_i, y_i) . Note that there are N_{stim} items i in a stimulus, and N_{types} different item types in $Types$. Typically, but not necessarily,

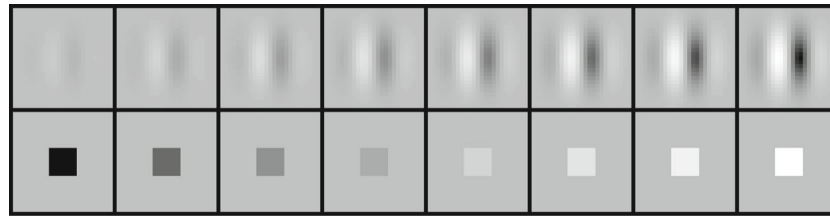


Fig. 1 The sets T_{types} of item types used in the two example experiments: The Gabor patterns experiment and The dots experiment. *Top row:* Gabor patterns with contrasts $\frac{1}{8}, \frac{1}{4}, \dots, 1$. *Bottom row:* Enlarged images of small, square dots with Weber contrasts $-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$

$N_{stim} \geq N_{types}$, and different items in a display may well be of the same type.

The target attention filter ϕ assigns a weight $\phi(\tau_i)$ to item i . Thus the spatial coordinates of the target centroid $(x_{correct}, y_{correct})$ are

$$\begin{aligned}
 x_{correct} &= \frac{1}{S} \sum_{i=1}^{N_{stim}} \phi(\tau_i)x_i \quad \text{and} \\
 y_{correct} &= \frac{1}{S} \sum_{i=1}^{N_{stim}} \phi(\tau_i)y_i \quad \text{for} \\
 S &= \sum_{i=1}^{N_{stim}} \phi(\tau_i). \tag{1}
 \end{aligned}$$

By convention, ϕ is scaled so that the sum over item Types is 1.0; i.e., $\sum_{k=1}^{N_{types}} \phi(k) = 1.0$.

Stimuli

In the Uniform, Graded and Inverse-graded attention conditions, every stimulus cloud included two instances of each of the eight Gabor patterns in the top row of Fig. 1. In the Lowest-only and Highest-only attention conditions, every stimulus cloud included three instances of each of these Gabor patterns. Therefore, $N_{stim} = 16$ for the Uniform, Graded and Inverse-graded attention conditions, and $N_{stim} = 24$ for the Lowest-only and the Highest-only attention conditions.

For any given target filter ϕ , the participant’s task was to try to mouse-click the ϕ -weighted centroid $(x_{correct}, y_{correct})$ of the stimulus cloud.

Sequence of trial events

The events that occurred in a trial in the Gabor experiment are shown in panels (a) through (f) of Fig. 2:

1. The participant initiated a trial by pressing the space bar. A blank screen of mean luminance was then presented for 1000 ms. In this display, a thin black line framed the region in which the stimulus cloud would be displayed (Fig. 2a).

2. The stimulus (Fig. 2b) was presented for 100 ms after which it was replaced by a blank stimulus field identical to Fig. 2a for 50 ms. The locations of the Gabor patterns were drawn from a bivariate Gaussian density constrained (as described below in “Generating full-set stimulus clouds”) to keep cloud size constant across trials.
3. 150 ms after the stimulus onset, a post-stimulus mask (Sperling, 1963) was presented of the sort shown in Fig. 2c; this mask stayed on for 100 ms.
4. The mask was then replaced by a blank stimulus field (Fig. 2d) with a cross-shaped mouse cursor in the middle.
5. The participant used the mouse to move the cursor (as indicated in Fig. 2e) to click on what he/she judged to be the correct location.
6. Then the participant was presented with feedback consisting of
 - (a) the stimulus,
 - (b) the mouse cursor located at the participant’s response, and
 - (c) a bullseye centered at the location of the correct response $(x_{correct}, y_{correct})$ (Eq. 1).

(The feedback panel in Fig. 2f shows that on this trial, the participant’s response was slightly below and to the right of the correct response.) The feedback stayed on the screen until the participant pressed the space bar to initiate the next trial.

Recorded on every trial:

1. the x - and y -coordinates and types of all items presented;
2. the x - and y -coordinates of the response location clicked on by the participant.

Generating full-set stimulus clouds

Of central interest in the attention condition with a given target filter ϕ are the responses produced by the participant on *full-set* trials. Each full-set stimulus cloud contains at least one of each type of item, and the number n_k of items of type

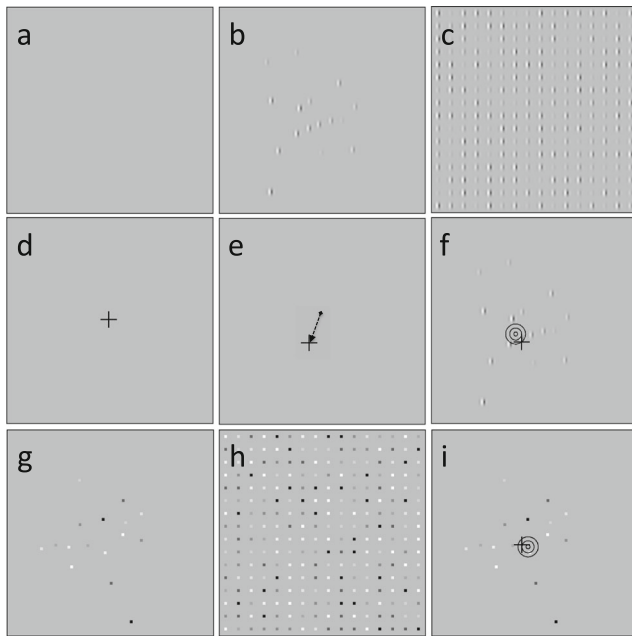


Fig. 2 The sequence of displays that occur during a trial. Panels (a) through (f) show the displays in a trial from the Gabor experiment: The participant initiates a trial by pressing the space bar and is then presented with a blank screen (a) with a thin, square, black frame surrounding the region where the stimulus will be displayed (1 s). Then, the stimulus (b) is presented for 100ms; it is followed by a display identical to (a) for 50ms. Then, a post-stimulus mask (c) is presented for 100 ms. After the mask, the participant is presented with (d), a blank screen with a mouse cursor in the middle. The arrow in (e) indicates the participant's movement of the mouse cursor to click on a response location, (R_x, R_y) . Then, feedback (f) is presented consisting of *i.* the stimulus, *ii.* the mouse cursor located at (R_x, R_y) , and *iii.* a bullseye centered at the location of the correct response $(x_{correct}, y_{correct})$ given by Eq. 1. This display remains on until the participant presses the space bar to initiate the next trial. In the Dots experiment, the sequence of events is identical; however, Panels (g), (h), and (i) replace panels (b), (c), and (f) respectively

k in *Types* is the same on every full-set trial. However, there is no requirement that the n_k 's be equal.¹

Every full-set cloud in the attention condition with a given target filter ϕ contains the same number N_{stim} of items; what should the spatial distribution of these items be? For reasons that will become apparent later, it is useful (1) to fix the expectation of the center of the cloud at the center of the stimulus field, and (2) select the standard deviation of the cloud distribution to insure that the clouds we present are contained within the stimulus field. To avoid unwittingly imposing any additional constraints, the natural choice for

¹As will become clear, fixing the n_k 's across trials allows the use a very simple and efficient linear regression procedure to estimate the attention filter f_ϕ achieved by the participant in the attention condition with any given target filter ϕ . It also makes it easy to (1) define the "Filter-fidelity" statistic (Eq. 8) that is used to measure the deviation of f_ϕ from ϕ and (2) to compute the " p_{miss} " statistic that is used to measure the corruption of the participant's responses by random error.

the distribution of item locations is a bivariate Gaussian density, which is the maximum entropy distribution for a fixed mean and variance.

There is, however, a problem with this simple strategy. Specifically, when the x - and y -coordinates of item locations are independent random variables, full-set stimulus clouds vary randomly (and strongly) across trials in how far items are spread out around the centroid. In the centroid task, it is empirically observed that responses tend to be more accurate on the trials in which items happen to bunch closely around the centroid than they do on trials in which the items are dispersed more broadly. In subsequent analyses, it will be critical to separate response variability due to trial-to-trial variability in the stimulus centroid location $(x_{correct}, y_{correct})$ from variability due to other stimulus factors. This can be accomplished much more easily when item clouds are created that do not vary in size, i.e., in dispersion.

Dispersion

To deal with the problem of varying cloud size, it helps to define the *dispersion* of a cloud of items. Let the vectors of x - and y -coordinates of the item locations in a given cloud be $\mathbf{x} = (x_1, x_2, \dots, x_{N_{stim}})$ and $\mathbf{y} = (y_1, y_2, \dots, y_{N_{stim}})$. Then the Dispersion(\mathbf{x}, \mathbf{y}) of the stimulus cloud composed of \mathbf{x}, \mathbf{y} is

$$\text{Dispersion}(\mathbf{x}, \mathbf{y}) = \left[\frac{1}{2N_{stim} - 2} \sum_{i=1}^{N_{stim}} (x_i - \bar{X})^2 + (y_i - \bar{Y})^2 \right]^{\frac{1}{2}}, \quad (2)$$

where \bar{X}, \bar{Y} are the means of the vectors \mathbf{x}, \mathbf{y} . Note: Dispersion is proportional to the root-mean-square (RMS) distance of the display items from their mean location; the proportionality constant is chosen so that Dispersion(\mathbf{x}, \mathbf{y}) is an unbiased estimator of the standard deviation used to generate the cloud.

To keep the value of dispersion constant at some value D for all full-set stimulus clouds used in a given experiment:

1. draw independent standard normal random variables \tilde{x}_i and $\tilde{y}_i, i = 1, 2, \dots, N_{stim}$.
2. and then produce the x - and y -coordinates of the actual item locations by setting

$$\begin{aligned} x_i &= \frac{D\tilde{x}_i}{\text{Dispersion}(\tilde{x}, \tilde{y})} \quad \text{and} \\ y_i &= \frac{D\tilde{y}_i}{\text{Dispersion}(\tilde{x}, \tilde{y})} \quad \text{for} \\ i &= 1, 2, \dots, N_{stim} \end{aligned} \quad (3)$$

This process starts with a cloud of points $(\tilde{x}_i, \tilde{y}_i)$; then the location of each dot gets shifted relative to the center of the

screen by factor $\frac{D}{Dispersion(\bar{x}, \bar{y})}$. The choice of D must strike a compromise between:

1. maximizing the information derived from each trial by making D as large as possible, and
2. insuring that all items in the stimulus cloud fit within the stimulus field.

A procedure that works well is to choose D in a given experiment so that approximately 95 % of the full-set stimulus clouds produced are contained within the stimulus field. When a given cloud produced using this D has one or more items that fall outside the stimulus field, that cloud sample is discarded, and a new cloud sample is produced.

For example, in the Uniform, Graded and Inverse-graded attention conditions in the Gabor experiment, each stimulus cloud comprised two Gabor patterns of each contrast value. The expectation of the (unweighted) centroid of each stimulus cloud was the center of the stimulus region. The dispersion (Eq. 2) of each stimulus cloud was 80 pixels (1.65 deg. of visual angle). This value was chosen because it led to discarding roughly 5 % of the stimulus clouds produced due to one or more item locations falling outside the stimulus region. An additional constraint was that the center-to-center distance between items (each of which subtended 25×25 pixels) was constrained to be at least 26 pixels to prevent items from overlapping.

General training in the centroid task

It will generally be useful to start an experiment by training the participant (with trial-by-trial feedback) to extract centroids of clouds. The number of items included in the displays used in this phase of training is typically equal to the number that will ultimately be used in data collection; however, the items are identical, even though the stimulus items will vary in the actual experiment. If the data collection phase mixes trials that include different numbers of items (as in the Dots experiment described below), the training trials in this phase may similarly vary the numbers of items occurring in clouds. Also, the post-stimulus mask used in this phase has an SOA identical to the masking SOA used in the data-collection phase.

The purpose of this training is to minimize idiosyncratic differences in the centroid computations used by different participants. As noted by Drew et al. (2010), in the absence of general training in the centroid task, different participants show significant individual differences in the centroid computations they use. In particular, some participants tend to overweight the contributions of peripheral items relative to items near the center of the cloud whereas other participants show the opposite tendency. These effects can be quite strong. Typically, a participant should remain in

this phase of the experiment until his/her performance (as reflected by mean response error) has stabilized. Only then should he/she be introduced to displays composed of items of different item types.

Data collection

After the participant has completed general training, he/she participates in several different attention conditions. In each attention condition, he/she is asked to use a new target filter ϕ to weight display items. It is natural to expect that performance in the task with each new target filter ϕ will require practice. Accordingly, for each new ϕ , it is important to begin by training the participant to perform the task with this target filter, collecting test data only after performance has stabilized.

ϕ -specific training

Using clouds in which items of different types are mixed, the participant is trained (with trial-by-trial feedback) to mouse-click centroids of clouds with items weighted by the target filter ϕ . (On a given trial, the correct response is given by Eq. 1.)

Standard training

The nature of the training used in a given attention condition is likely to depend on the target filter ϕ . We call a target filter ϕ *binary* if ϕ assigns equal weight to some subset of “target” items in *Types* and weight 0 to the remaining “distractor” items. For a binary target filter (and sometimes for other target filters), ϕ -specific training typically uses the same mix of item-cloud conditions as will be used in the data collection phase (see “Data collection with target filter ϕ ”). In these instances, the participant typically is tested in blocks of 100 trials. After each block, the participant is shown the attention filter he/she achieved in that block as well as several summary measures of accuracy. This feedback is provided to enable the participant to adjust his/her strategy to optimize performance. Because the procedures in ϕ -specific training blocks are identical to procedures in data-collection blocks, it is typically necessary only to retain as data the results from the first two blocks in which performance shows no improvement.

Pretraining

For non-binary target filters ϕ , however, it will sometimes be useful to include an initial phase of ϕ -specific training that uses “simplified” item-clouds that comprise fewer items than will be used in the data-collection phase. This

strategy is likely to be appropriate if ϕ assigns a range of different weights to the different elements of *Types*. In particular, this pretraining phase might include (often exclusively) clouds that comprise just two items. For a two-item cloud whose items are of types i and j in *Types*, the correct response lies $\frac{\phi(j)}{\phi(i)+\phi(j)}$ of the way from the location of the item of type i to the location of the item of type j . When the participant can produce appropriate responses to all such two-item clouds, this suggests that he/she understands the task at a rudimentary level. Following this pretraining, the participant progresses to standard training and data collection.

Data collection in the Gabor example

Blocks in the Gabor experiment comprised 100 full-set trials. On each full-set trial in each of the Uniform, Graded and Inverse-graded (Lowest-only and Highest-only) attention conditions, 16 (24) Gabor patterns were presented, two (3) of each contrast. The number of display items was increased in the Lowest-only and Highest-only attention conditions to insure that a given stimulus display contained at least three target items.

Procedure with naive participants

S3 completed 400 trials of general training (~20–27 min), S4 completed 200 trials (~10–13 min). In the final block of general training trials, each of participants S3 and S4 achieved a mean response error comparable to the mean response errors typically achieved by practiced participants, and general training was terminated.

Each of participants S3 and S4 completed 200 trials in each attention condition following a number of ϕ -specific training trials that varied around a mean of 333 depending on the particular target filter ϕ .²

Neither S3 nor S4 was tested in either of the Lowest-only or Highest-only attention conditions.

²Performance was poorer in the Inverse-graded attention condition than in either the Uniform or Graded attention conditions. Accordingly, a large number of ϕ -specific training trials were devoted to this attention condition (600) for each of participants S3 and S4 in order to insure that performance had stabilized. In addition, S3 required a greater number (400) of ϕ -specific training trials in the Graded attention condition than did S4 (100). S3 completed the different attention conditions in the following order: Inverse-graded (600 ϕ -specific training trials, and 200 data-collection trials over 2 days)→Uniform (100 ϕ -specific training trials, and 200 data-collection trials in 1 day)→Graded (400 ϕ -specific training trials, and 200 data-collection trials over 2 days); and S4 was tested in the reverse order from S3: Graded (100 ϕ -specific training trials, and 200 data-collection trials in 1 day)→Uniform (100 ϕ -specific training trials, and 200 data-collection trials in 1 day)→Inverse-graded (600 ϕ -specific training trials, and 200 data-collection trials over 2 days).

Procedure with experienced participants

Each of participants S1 and S2 had extensive previous experience in the centroid task. In addition, each had previous experience in variants of the centroid task using target filters similar to those tested in the current experiment. Accordingly, general training was omitted for each. Each of S1 and S2 performed 100 ϕ -specific training trials followed by 200 data-collection trials in each attention condition. Attention conditions were tested in the following order: Uniform → Graded → Inverse-graded → Lowest-only → Highest-only.

Modeling

Some attention conditions in the centroid task are likely to be harder than others. The difficulty encountered by a participant in an attention condition using a given target filter ϕ is likely to show up in two main ways:

1. The target filter f_ϕ achieved by the participant may deviate from the target filter.
2. Response accuracy may be compromised by
 - (a) random errors,
 - (b) a bias toward a fixed response location.

In analyzing the data from a given attention condition, it will be important to measure the strengths of these effects.

Model assumptions

The primary variable of interest is the attention filter f_ϕ achieved by the participant across just the full-set trials in the attention condition using a given target filter ϕ .³ However, the model includes four other parameters to account for specific sources of error that may influence responses: a default location ($x_{default}$, $y_{default}$), Data-drivenness (V) that describes a participant's reliance on the present stimulus versus the default location, and a noise parameter σ . These parameters are defined below.

To model the process by which the x -coordinate $R_x(j)$ and y -coordinate $R_y(j)$ of the participant's response are produced on a full-set trial j we define the following:

1. $\tau_i(j)$, $x_i(j)$ and $y_i(j)$ are the type, and x - and y -coordinates of the i^{th} item in the stimulus cloud presented on trial j .
2. $Q_x(j)$ and $Q_y(j)$ are independent, normally distributed random variables, each with mean 0 and variance σ^2 that represent random response error.

³In an experiment that also includes target-only trials (see "Target-only clouds"), the same method can be used to estimate the attention filter achieved across the target-only trials (provided there are multiple simultaneous targets).

3. V , the *Data-drivenness* parameter, reflects the proportion to which the participant’s response is determined by the stimulus presented on each trial as opposed to the fixed point $(x_{default}, y_{default})$ to which the participant’s response is assumed to tend (with weight $(1 - V)$) on each trial.
4. f_ϕ is the attention filter achieved by the participant across the full-set trials in the attention condition defined by the target filter ϕ .
5. S is the sum of the weights assigned to all the different items in any given full-set display. That is, on any trial j ,

$$S = \sum_{i=1}^{N_{stim}} f_\phi(\tau_i(j)) = \sum_{k=1}^{N_{types}} n_k f_\phi(k) \tag{4}$$

where for $k = 1, 2, \dots, N_{types}$, n_k is the number of items of type k in a full-set display. We assume without loss of generality that S is positive.

Then, the x - and y -coordinates of the predicted response on a given trial j are

$$R_x(j) = \frac{V}{S} \sum_{i=1}^{N_{stim}} f_\phi(\tau_i(j))x_i(j) + (1 - V)x_{default} + Q_x(j), \quad \text{and} \tag{5}$$

$$R_y(j) = \frac{V}{S} \sum_{i=1}^{N_{stim}} f_\phi(\tau_i(j))y_i(j) + (1 - V)y_{default} + Q_y(j). \tag{6}$$

The methods used to estimate the model parameters V , $x_{default}$, $y_{default}$, and f_ϕ are described in “Appendix 1. Estimating model parameters.” Methods for computing 95 % confidence intervals for V (Data-drivenness) as well as for the $f_\phi(k)$, $k = 1, 2, \dots, N_{types}$, are described in “Appendix 2. Estimating confidence intervals for model parameters.” The Matlab code that is used to perform these computations is given in “Appendix 4. Matlab code for fitting the centroid model.”

Results from the Gabor example experiment

Figure 3 shows the results for all participants for the Uniform, Graded, and Inverse-graded attention conditions. The attention filters f_ϕ achieved by all four participants in the Graded attention condition match the target filter ϕ fairly well. In addition, the attention filter achieved by participant S3 in the Uniform attention condition matches the corresponding target filter ϕ remarkably well. However, systematic deviations of f_ϕ from ϕ are evident for participants S1, S2 and S4 in the Uniform attention condition

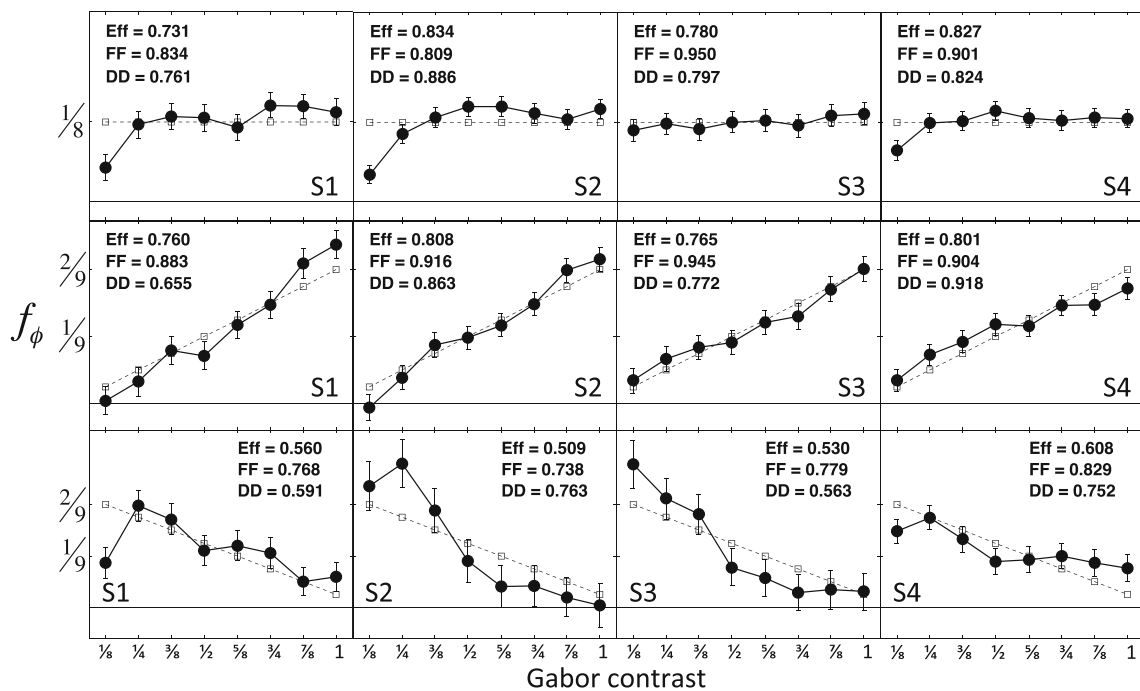


Fig. 3 Results for the Uniform, Graded and Inverse-graded attention conditions in the Gabor experiment for four participants. The four columns from left to right correspond to participants S1, S2, S3, and S4. Top, middle, and bottom panels show results for the Uniform, Graded and Inverse-graded attention conditions, respectively. In each panel, the dashed line with small open squares gives the target filter ϕ , and the solid line with filled circles gives the attention filter f_ϕ achieved by the participant. Error bars give 95 % confidence intervals derived using the methods described in “Appendix 2. Estimating

confidence intervals for model parameters.” In addition, the values of Efficiency (Eff), Filter-fidelity (FF), and Data-drivenness (DD) achieved in each attention condition are shown in the corresponding panel. Note that each participant achieves strikingly different attention filters in the Uniform and Graded attention conditions, in each case with high Efficiency. However, values of Efficiency, Filter-fidelity and Data-drivenness are lower in the Inverse-graded attention condition compared to the other two attention conditions

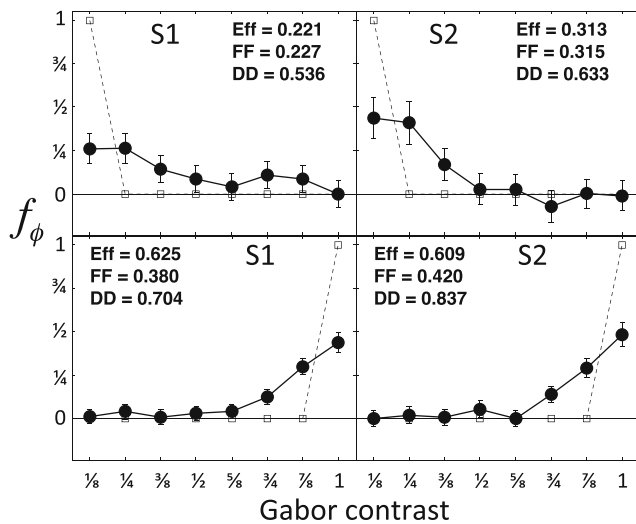


Fig. 4 Results for the Lowest-only and Highest-only attention conditions in the Gabor experiment. Results for participant S1 (S2) are shown in the left (right) two panels. Results for the Lowest-only (Highest-only) attention condition are shown in the top (bottom) two panels. In each panel, the dashed line with small open squares gives the target filter ϕ , and the solid line with filled circles gives the attention filter f_ϕ achieved by the participant. Error bars give 95 % confidence intervals computed as described in “Appendix 2. Estimating confidence intervals for model parameters.” Note that values of Efficiency (Eff), Filter-fidelity (FF), and Data-drivenness (DD) are low in comparison to the values observed in Fig. 3 for participants S1 and S2 in the Uniform and Graded attention conditions

and for all four participants in the Inverse-graded attention condition.

Figure 4 shows the results for S1 and S2 in the Lowest-only and Highest-only attention conditions. The attention filter f_ϕ achieved by each of participants S1 and S2 in each of these two attention conditions deviates strongly from the target filter ϕ . In each case, although the target filter ϕ assigns nonzero weight to Gabor patterns of a single contrast (contrast 0.125 (1.0) in the Lowest-only (Highest-only) attention condition), the attention filter f_ϕ achieved by each participant gives nonzero weight to Gabors ranging broadly in contrast.

In addition to plotting the target filter ϕ and the attention filter f_ϕ achieved by the participant, each of the panels in Figs. 3 and 4 is annotated with three additional statistics: “Data-drivenness,” “Filter-fidelity” and “Efficiency.” Data-drivenness is parameter V in Eqs. 5 and 6. Along with Data-drivenness, Filter-fidelity and Efficiency reflect the overall skill of the participant in the given attention condition. They are explained in the next section.

Analyzing the data: response error

Potential sources of response error

The participant’s responses can deviate from the target responses for various reasons. These include:

1. corruption of responses by random error, sources of which include
 - (a) early perceptual noise, including
 - i. misregistration of the locations of items in the display
 - ii. misregistration of the types of different items
 - iii. failure to register (i.e. missing) some items
 - (b) late noise, including instability across trials in
 - i. the attention filter being deployed
 - ii. the centroid computation
 - iii. motor response execution
2. corruption of responses by nonrandom error, sources of which include
 - (a) mismatch between the attention filter f_ϕ versus the target filter ϕ ,
 - (b) Data-drivenness less than 1, implying a tendency to produce responses biased toward a fixed default location,
 - (c) model failure (i.e., the computation used by the participant to produce responses deviates from the description provided by Eqs. 5 and 6)⁴

Measuring the quality of the participant’s attention filter: filter-fidelity

We use a statistic called *Filter-fidelity* to measure the effectiveness with which f_ϕ approximates the target filter ϕ for purposes of performing the centroid task on the item clouds used in a given type of trial (e.g., full-set trials or target-only trials). Filter-fidelity ranges in value from 0, if the attention filter f_ϕ achieved by the participant for this class of item clouds is the worst possible filter, to 1 if $f_\phi = \phi$. In this context, “worst” means that the variance of the difference between the x -coordinates (or the y -coordinates) of the centroids derived by using f_ϕ vs. ϕ is maximal.

⁴An important type of model failure can result from the use of a centroid computation that systematically overweights or underweights peripheral display items vs. items near the center of the stimulus cloud. Such spatially inhomogeneous weighting of display items was extensively analyzed by Drew et al. (2010) who found considerable variation across participants. It should be noted, however, that the participants of Drew et al. (2010) received no general training in the centroid task. The purpose of general training is precisely to minimize such idiosyncratic differences between the centroid computations used by different participants. Moreover, the model parameters introduced by Drew et al. (2010) to accommodate variations in the centroid computations used by different participants did not significantly influence the estimated attention filters. For these reasons, individual differences in the centroid computations used by different participants are not further considered here.

A worst possible attention filter $f_{\phi, worst}$ (there may be more than one worst attention filter) is derived by putting all the filter weight on a single item-type which should (given the task demands) exert minimal influence on the participant’s response. Specifically, a worst possible attention filter can be obtained by choosing an item-type j for which $n_j \phi(j)$ is minimal across $j = 1, 2, \dots, N_{types}$ and setting

$$f_{\phi, worst}(k) = \begin{cases} 1 & \text{if } k = j, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

for all types $k = 1, 2, \dots, N_{types}$.

Then Filter-fidelity is defined as

$$\text{Filter-fidelity} = 1 - \frac{\|\tilde{f}_\phi - \tilde{\phi}\|}{\|\tilde{f}_{\phi, worst} - \tilde{\phi}\|}, \tag{8}$$

where, for any function $f : Types \rightarrow \mathbb{R}$ (for which the denominator in Eq. 9 is nonzero),

$$\tilde{f} = \frac{f}{\sum_{k=1}^{N_{types}} n_k f(k)}. \tag{9}$$

This normalization insures that the x - and y -coordinates of the centroid of a full-set display derived using f are the following weighted sums:

$$\begin{aligned} x\text{-coordinate} &= \sum_{\substack{\text{all items } i \text{ in} \\ \text{full-set cloud}}} \tilde{f}(\tau_i) x_i \quad \text{and} \\ y\text{-coordinate} &= \sum_{\substack{\text{all items } i \text{ in} \\ \text{full-set cloud}}} \tilde{f}(\tau_i) y_i \end{aligned} \tag{10}$$

where x_i and y_i are the x - and y -locations of the i^{th} item in the display, and τ_i is its type.

Thus Filter-fidelity is the ratio of the Euclidean distance (in N_{types} -dimensional space) of \tilde{f}_ϕ from $\tilde{\phi}$ to the Euclidean distance of the worst possible filter (i.e., $\tilde{f}_{\phi, worst}$) from $\tilde{\phi}$.

The Filter-fidelity values, computed according to Eq. 8, are displayed in all panels in Figs. 3 and 4. All Filter-fidelity values for the Uniform and Graded attention conditions are quite large (> 0.83) reflecting the skill displayed by all four participants in matching the target functions. By contrast, for all participants, Filter-fidelity values are from 8 % to 23 % lower for the Inverse-graded attention condition.

The Lowest-only and Highest-only attention conditions are especially difficult. Across the Uniform, Graded and Inverse-graded conditions, the lowest Filter-fidelity values were achieved in the Inverse-graded condition. For S1 (S2) this Filter-fidelity value was 0.765 (0.732). By contrast, in the Highest-only condition S1 (S2) achieved Filter-fidelity 0.381 (0.421) (Fig. 4). In the Lowest-only condition, Filter-fidelity is even worse, indicating that although participants attempt to selectively attend to Lowest-only and Highest-only targets, they cannot successfully do so.

Measuring resistance to residual error: Efficiency

All of the error in the data unaccounted for by the model of Eqs. 5 and 6 is captured in $SS_{Residual}$ (Eq. 24). This quantity includes both

1. random error from various early and late sources in the response-production process as well as
2. error due to model failure.

The statistic that is typically used to quantify random error in the context of a regression analysis is

$$\hat{\sigma} = \sqrt{\frac{SS_{Residual}}{df}}, \tag{11}$$

where

$$\begin{aligned} df &= 2N_{trials} - (\text{number of free model parameters}) \\ &= 2N_{trials} - (N_{types} + 2). \end{aligned} \tag{12}$$

Although the model has parameters V , f_ϕ (which is of length N_{types}), $x_{default}$, $y_{default}$, the number of free parameters is equal to $N_{types} + 2$ because f_ϕ is constrained to sum to 1 (therefore it absorbs only $N_{types} - 1$ degrees of freedom). (In the Gabor experiment, $df = 10$ because $N_{types} = 8$.) The statistic $\hat{\sigma}$ is an unbiased estimate of the standard deviation of each of the random variables Q_x and Q_y (Eqs. 5 and 6). In itself, however, $\hat{\sigma}$ is difficult to interpret.

An alternative measure that facilitates comparison across experiments is the statistic p_{miss} . To get a clear sense of what p_{miss} reflects, imagine a centroid task in which

1. $Types$ contains only two items A and B ,
2. the stimulus cloud on any given trial comprises 10 items of type A and 10 of the B , and
3. the task is to click on the centroid of the locations of all items of type A and ignore all items of type B .

If it is difficult to achieve an attention filter that is selective for items of type A vs. type B , then the participant may adopt the strategy of picking out a single item of type A on each trial and simply clicking on the location of that one item.

Under this strategy, on each trial, the participant’s response is determined exclusively by items (actually, by only one item) of type A ; items of type B exert no systematic influence whatsoever. Thus, the attention filter achieved by the participant will match the target filter nearly perfectly. Nonetheless, performance will be very poor because the participant ignores nearly all of the relevant information in the display in producing his/her response on each trial. For this reason, even though Filter-fidelity is likely to be very close to 1, $\hat{\sigma}$ (Eq. 11) will be large. In the experiment imagined here, if the participant were always able to

find exactly one item of type A and to click with perfect accuracy on its location, this strategy will yield a value of $p_{miss} = 0.90$ because the participant is failing to include nine of the ten requested items in the display.

More generally, p_{miss} is the answer to the following question: Given

1. the attention filter f_ϕ achieved by the participant, and
2. the observed value of $\hat{\sigma}$,

what is the maximum possible proportion of display items that the participant could be failing, trial by trial, to include in his/her centroid computation?

Another way of thinking about p_{miss} is in terms of an ideal detector operating on a reduced stimulus. Suppose a computer performs the centroid task as follows: On each trial, the computer (1) discards proportion p of items from the stimulus (where the specific items discarded are chosen randomly), then (2) applies attention filter f_ϕ to the remaining items in the display, and (3) extracts (without any additional error) the centroid of the decimated and filtered display. The centroid derived through this procedure will vary randomly in each of the x - and y -coordinate values with some standard deviation σ_p that will increase with p . The statistic p_{miss} is the value that p must take in order for σ_p to be equal to $\hat{\sigma}$.

Although p_{miss} is useful as a summary of performance, it should not be taken seriously as an estimate of the proportion of items actually missed by the participant. It assigns all of the residual error to missed stimulus items; however, any credible process model must admit possible contributions to $\hat{\sigma}$ from both model failure as well as from all of the noise sources outlined above. Thus, p_{miss} should be viewed as an upper bound on the proportion of display items missed by the participant in his/her centroid computation. Following conventional nomenclature, we refer to $1 - p_{miss}$ as *Efficiency* (which is a lower bound on the proportion of display items included by the participant in his/her centroid computation).

As described below, and illustrated in the Dots Experiment, the lower bound on the proportion of display items included by the participant in his/her centroid computation can be lowered further by including in the experiment “singleton” trials in which a single target item is presented. The variance of the errors produced on these trials is due to other sources than failing to include display items in the centroid computation. It can therefore be subtracted from $\hat{\sigma}^2$ for purposes of computing p_{miss} . Whenever the Efficiency value reported from a particular experiment reflects such a correction, we refer to it as “singleton-corrected Efficiency.”

An algorithm to compute p_{miss} is described in “Appendix 3. Computing Efficiency,” and the Matlab code implementing this algorithm is given in “Appendix 4. Matlab code for fitting the centroid model.” (see specifically

“GetPMiss.m (Called by FitCentroidModel)” and the functions called by GetPMiss.m.)

The Efficiency values, computed as described in “Appendix 3. Computing Efficiency” are displayed in all the panels in Figs. 3 and 4. High Efficiency in a given attention condition indicates that the participant can indeed deploy the attention filter he/she has achieved broadly across space with high sensitivity. The current results support this conclusion for all participants in the Uniform and Graded attention conditions. However, Efficiency values for the Inverse-graded, Lowest-only and Highest-only attention conditions are much lower. As previously observed, Filter-fidelity values also tend to be suppressed for these attention conditions. It appears that human vision does not have the capability to produce attention filters matched to these attention conditions, at least not with the number of training trials provided here.

The relation between Data-drivenness, Filter-Fidelity and Efficiency

There are several relationships between Efficiency, Data-drivenness and Filter-fidelity that should be noted.

1. It is eminently possible for a participant to achieve very high values of both Filter-fidelity as well as Data-drivenness in a given attention condition even though his/her Efficiency is very low. Efficiency thus emerges as a key measure of performance. If Efficiency is low, then even if Filter-fidelity and Data-drivenness are both high, we must conclude either that (1) the attention filter achieved by the participant cannot be effectively deployed broadly across space or else (2) the items disclosed by the attention filter are too poorly localized in the output image of the filter to enable an accurate estimate of the centroid.
2. On the other hand, if Data-drivenness is low, then Efficiency is likely to be low as well. In particular, note that if Data-drivenness is 0, then the participant’s responses do not depend at all on the locations and types of items in the display. In this case, removing items from the stimulus display has no effect on the deviation of the participant’s responses from the responses predicted by the model. This means that if Data-drivenness is 0, then Efficiency will be undefined. More generally, for any fixed value of $\hat{\sigma}$ (Eq. 11), Efficiency will shrink with shrinking Data-drivenness V and will be undefined if $\frac{\hat{\sigma}}{V}$ is greater than the dispersion D used to control the size of stimulus clouds in a given experiment.

Simulations exploring interactions between Data-drivenness, Filter-fidelity and singleton-corrected Efficiency Both as a way to test the accuracy of the FitCentroidModel program (Appendix 4) and to understand better

the dependencies among the estimates it produces, we carried out a series of Monte-Carlo simulations. The attention filter of the simulated observer was similar to that achieved by Subject 3 in the Dark-only condition of the Dots experiment (Fig. 6, top row, third panel from left). Simulated data were produced for 60 variants of this observer. These variations were derived from the factorial combination of three factors: five levels of stimulus decimation proportion p (i.e., on each simulated trial, the observer failed to incorporate a randomly selected subset comprising proportion p of the display items into the centroid computation, with p ranging from 0 up to 0.625), four levels of Data-drivenness V , ranging $V = 1$ down to $V = 0.4$, and three levels of late error, σ_L (i.e., independent, normal random variables with standard deviation σ_L were added to the x - and y -coordinates of the response location on each simulated trial), which ranged from 10 to 40 % of the dispersion of the stimulus cloud (see Eq. 2). In addition, in each condition singleton-corrected Efficiency was estimated using two different levels of singleton standard deviation, $\sigma_{singleton}$. Five hundred simulated runs of a 100-trial block were generated and analyzed for each of these 120 variations.

Estimates of the attention filter weights, $f_{\phi,full-set}$, were generally quite accurate, as were the estimates of Data-drivenness. Specifically, when the decimation level was zero, these estimates were unbiased. In the extreme case, in which the simulated observer registered only 37.5 % of the display items, the mean estimate of an attention filter weight with a simulated value of 0.270, was reduced to 0.252; a distractor component of the attention filter, with a simulated weight of 0.080, was increased to 0.090. Although the simulated value of σ_L did not influence the bias of these estimates, it did affect their variability.

The results for singleton-corrected Efficiency are more complicated because this measure is decreased by any manipulation that reduces response accuracy (other than misspecification of the attention filter). In the special case in which Data-drivenness $V = 1.0$ and the value of $\sigma_{singleton}$ used for the analysis exactly matches the σ_L of the simulated observer, the estimated values of singleton-corrected Efficiency closely match $1 - p$, where p is the decimation proportion. However, the estimated singleton-corrected Efficiency was also reduced when V was less than 1.0 or $\sigma_L > \sigma_{singleton}$. Because singleton-corrected Efficiency is constrained to lie between zero and one, the effects of these three sources of judgment error (V , σ_L , $\sigma_{singleton}$) on estimated singleton-corrected Efficiency are subadditive.

Testing whether two attention filters are significantly different

In various circumstances, it may be of interest to assess whether the attention filters estimated from two sets of data

are significantly different. Matlab code to perform an F -test of this hypothesis is provided in “Appendix 5. Matlab code for nested model comparison.” See specifically “Main program: FTestForEqualityOfFilters.m.” The test compares the fits provided by two models:

1. The *full* model allows all model parameters, including the two attention filters to take different, arbitrary values for the two data sets.
2. The *nested* model assumes that the two data sets resulted from the use of single, shared attention filter (with the other model parameters, V , $x_{default}$, and $y_{default}$, allowed to take different values for the two data sets).

Let $N_{trials,1}$ ($N_{trials,2}$) be the number of trials in the first (second) data set. Then the numbers of degrees of freedom in the full and nested models are

$$df_{full} = 2(N_{trials,1} + N_{trials,2}) - 2(N_{types} - 1) - 6 \quad \text{and} \quad (13)$$

$$df_{nested} = 2(N_{trials,1} + N_{trials,2}) - (N_{types} - 1) - 6, \quad (14)$$

and for SS_{full} (SS_{nested}) the sum of squared residual errors between observed and predicted response locations under the full (nested) model, if the nested model captures the true state of the world, then the ratio

$$Q = \frac{\left(\frac{SS_{nested} - SS_{full}}{df_{numerator}} \right)}{\left(\frac{SS_{full}}{df_{denominator}} \right)} \quad (15)$$

has an F distribution with degrees of freedom

$$df_{numerator} = df_{nested} - df_{full} = N_{types} - 1, \quad (16)$$

and

$$df_{denominator} = df_{full}. \quad (17)$$

A test of the null hypothesis that the attention filters achieved by participants S1 and S2 in the Lowest-only attention condition (top two panels of Fig. 4) are identical in form, yields $F_{7,780} = 1.7052$, $p = 0.1044$. By contrast, a test of the null hypothesis that the attention filters achieved by participant S1 in the Highest-only and Lowest-only attention conditions (the left-hand panels of Fig. 4) are identical in form, yields $F_{7,780} = 30.9317$, $p \approx 0$.

Gabor experiment: summary and caveat

The prominent features of the Gabor experiment results are:

- Participants can flexibly deploy very different attention filters in response to altered task demands. Stimuli are identically distributed in the Uniform and the Graded attention conditions, yet participants achieve attention filters that differ strongly in form, in each case with high Filter-fidelity, high Efficiency and high Data-drivenness. The stimulus exposure duration was only 100 ms, and the stimulus is followed after 50 ms by

a pattern mask; thus, the accurate attention filters and high Efficiency were achieved with only 150 ms of available stimulus information.

- Participants are limited in the attention filters they can achieve. While they can achieve high values of Filter-fidelity, Efficiency and Data-drivenness when using Uniform and Graded attention filters, the values of these measures are suppressed in the Inverse-graded, Lowest-only and Highest-only attention conditions.
- *Caveat.* What is here designated as an attention filter may involve other processes than merely attention. For example, Fig. 3 shows that in the uniform and inverse attention conditions, participants S1, S2, S4 clearly have some difficulty in giving adequate weight to the lowest-contrast Gabor patch. On the other hand, in the graded attention condition, participants are required to give very little or zero weight to this lowest-contrast patch, and this they do quite well, especially when they cannot see it. These participants may be attending to the lowest-contrast patch as much as to the other patches, but fail to weight it properly because of deficient discrimination, not deficient attention. If one developed a measure of discriminability, and incorporated it into the attention filter computation, one might arrive at a “discriminability-corrected attention filter.” The choice we make here is to use the term “attention filter” for the simple but possibly impure concept, and to allow for the subsequent development of more complex, purer measures of attention.

The rest of this section focuses in more careful detail on the inferences enabled by the results of the Gabor experiment.

The filter mixture model

A basic framework for interpreting the results of centroid method experiments is the “filter mixture model.” This model proposes that participants possess a limited set of basic attention filters, i.e., a basis set. The large number of different observed attention filters is assumed to result from combinations of the basic attention filters. As a starting point, the basic attention filters are assumed to be “basic” in the sense that parametric variations in their properties are disallowed. This is formalized as follows. For a given set *Types*:

1. The participant possesses a basis set of attention filters f_j , $j = 1, 2, \dots, N$, which confer sensitivity to the item types in *Types* broadly across space, where
 - (a) each filter f_j is implemented by a retinotopically organized array of neurons in early vision, and
 - (b) $f_j(k)$ gives the activation produced in this neural array by items of type k .

2. The participant can produce attention filters

$$f = \sum_{j=1}^N A_j f_j \quad (18)$$

where the A_j 's are constrained

- (a) to be nonnegative (implying that the participant cannot reverse the sign of the pattern of sensitivity of a given basic attention filter) and
 - (b) to sum to a value no greater than 1 (imposing a bound on the sensitivity that the participant can achieve).
3. In the attention condition with target filter ϕ , the participant strives to choose the weights A_j in Eq. 18 to produce the attention filter $f_\phi = f$ that will minimize response error (the difference between judged and correct centroids) when f_ϕ is used in Eqs. 5 and 6.

Note that only if the participant possesses one or more basic filters that correlate strongly and positively with the target filter ϕ , will he/she be able to use Eq. 18 to produce an attention filter f sufficiently high in amplitude to robustly estimate the response location in spite of the noise in Eqs. 5 and 6.

It is beyond the scope of the current paper to discuss how one might estimate the basic filters f_j from the data from a centroid experiment or submit the filter mixture model to an empirical test. The model nonetheless provides several inferential principles that are useful. Under the filter mixture model, the attention filter f_ϕ that a participant achieves in the attention condition with target filter ϕ is a weighted sum of the basic filters f_j , $j = 1, 2, \dots, N$. If Efficiency is high in this attention condition, this suggests that

1. some of the basic filters f_j , $j = 1, 2, \dots, N$ correlate strongly and positively with ϕ , and
2. these useful filters are given large weights in the sum that yields the attention filter f_ϕ .

Under the filter mixture model, the results of the Gabor experiment suggest that the number N of basic filters in human vision with sensitivity to the eight Gabor patterns used in our stimuli is at least two. This follows from the finding that participants are able to achieve clearly distinct attention filters in the Uniform vs. Graded attention conditions, in each case with high Efficiency. The attention filters achieved in the Inverse-Graded, Lowest-only and Highest-only attention conditions also differ in form from each other as well as from the attention filters achieved in the Uniform and Graded attention conditions. However, the low Efficiencies observed in the latter three attention conditions imply that each of the available basic filters f_j , $j = 1, 2, \dots, N$, correlates either negatively or near 0 with the target filters used in these attention conditions; it would be precarious to

infer from these data that human vision possesses basis filters that correlate positively with the target filters in these attention conditions.

Elaborating and fine-tuning the centroid method

Choosing an appropriate stimulus onset asynchrony

As stimulus onset asynchrony (SOA: the time between stimulus onset and the onset of the post-stimulus mask) is increased, response error decreases to some asymptotic level. The decrease of response error with an increase in SOA reflects increased effectiveness of visual input relative to early noise in the processing stream. The asymptote of response error at long stimulus durations reflects random perturbations of the response process that are invariant with respect to the strength of the input signal. For example, all of the following could contribute to asymptotic response error level: (1) trial-to-trial instability in the centroid computation, (2) error in localizing the to-be-clicked-on location, and (3) motor error in registering the response.

Typical applications of the centroid method should use an SOA that is brief enough to preclude eye movements and/or spatial shifts of attention yet long enough to insure that response error has descended to its asymptotic level. Often a brief pilot experiment is required to select an appropriate SOA. The Dots Experiment (below) includes an example of such a pilot experiment.

More on stimulus clouds

Typically, to measure an attention filter, it is convenient to use full-set trials. Some applications require other sorts of trials. This section addresses the question of how best to construct the stimulus clouds used in some useful non-full-set trials.

Singleton trials

A “singleton” trial is a trial in which only a single item (whose type is typically fixed and selected to be highly salient) is presented. Singleton trials can provide a useful lower-bound on response error.

In an experiment using singleton trials, how should the locations of singletons be distributed? It is tempting to distribute singletons identically to individual items occurring in full-set clouds. This strategy, however, defeats the main purpose of including singleton trials: to derive a lower bound on response noise. If singletons are distributed identically to individual items occurring in full-set

clouds, then the participant will need to produce much more variable responses on singleton trials than he/she does on full-set trials (because the centroid of a full-set cloud has lower variance than the individual items in the cloud). Empirically, we have observed that response error increases with the variability of the target location. To equalize the variability of target response locations on singleton trials and full-set trials, on each singleton trial,

1. derive a full-set cloud using the method described in “Generating full-set stimulus clouds.” Then
2. place the singleton at the ϕ -weighted centroid of that cloud (Eq. 1).

Target-only clouds

In many attention conditions of interest the target filter ϕ assigns equal nonzero value to all items in a particular “target” subset T and 0 to the remaining “distractor” types D . We call such target filters *binary*. In the special case in which the target filter is binary, it is useful to mix three sorts of trials during both condition-specific training and data collection. The first two sorts of trials are the “full-set” and “singleton” trials discussed above. For binary target filters, it is also useful to include “target-only” trials in which the stimulus cloud contains the same mix of target items as on full-set trials but contains no distractor items. The target-only trials are equivalent to providing the participant with a perfect attention filter. The point of including target-only trials in a binary attention condition is to two-fold:

1. to compare performance (as reflected by the attention filter, Filter-fidelity, Efficiency and Data-drivenness) achieved in the presence of distractors with performance achieved with a perfect filter, i.e., with no distractors (e.g., Sperling et al 1992).
2. to enable the participant to refine his/her attention filter by experiencing stimulus clouds unpolled by distractors.

Both aims are achieved by using target-only clouds in which item locations are distributed identically to the locations of target items on full-set trials.

The dots experiment

The Dots Experiment is an example application that uses binary target filters. This experiment illustrates the utility of including a small number of (1) singleton trials and (2) target-only trials interspersed among full-set trials in experimental blocks.

Methods

Item types

In this experiment, *Types* included the eight square dots of different gray levels shown in the bottom row of Fig. 1. Dots were 7×7 pixels, subtending 0.144 deg. of visual angle at the viewing distance of 1 m. The Weber contrasts of the eight dots (relative to the uniform gray background) were approximately -1.0 , -0.75 , -0.5 , -0.25 , 0.25 , 0.5 , 0.75 and 1.0 .

Displays

As in the Gabor experiment, the stimulus region surrounded by the thin black frame (Fig. 2a) comprised 512×512 pixels. At the viewing distance of 1 m, this region subtended 8 deg. of visual angle. The luminance of the homogeneous background was 77 cd/m^2 . Stimulus clouds were constructed as described in “Generating full-set stimulus clouds.” The expectation of the centroid of each stimulus cloud was the center of the stimulus region. Each full-set stimulus cloud comprised two dots of each of the eight Weber contrasts. The dispersion (Eq. 2) of each full-set stimulus cloud was 80 pixels (1.65 deg. of visual angle). This value led us to discard roughly 5 % of the stimulus clouds produced due to one or more item locations falling outside the stimulus region. All dots were constrained to be separated from each other by at least two pixels. Each target-only trial comprised two dots of each target type and no distractor dots; the dots presented were distributed in the stimulus field exactly as they would have been in a full-set trial. Each singleton trial contained a single black dot (Weber contrast -1.0); the location of this dot was distributed identically to the correct response on a full-set trial.

Individual trials

The sequence of events that occurred on an experimental trial precisely paralleled the sequence that occurred in a trial of the Gabor patterns experiment, except that the display items were the dots shown in Fig. 1 rather than the Gabor patterns. Display durations were identical to those used in the Gabor experiment (see “The Gabor pattern experiment”). The sequence of displays in a full-set trial is illustrated in Fig. 2.

Attention conditions

Participants were tested in two, complementary full-set attention conditions using binary target filters ϕ . Stimulus clouds were composed of two dots of each of the

eight contrasts. In the first “Dark-only” attention condition the four dot types darker than the background were the target items, and the four dot types lighter than the background were the distractor items. That is, the target filter assigned equal weight to the eight target dots darker than the background and weight 0 to the eight distractor dots brighter than the background. In the “Light-only” attention condition, the roles of dark and light dots were reversed; light dots became the targets and dark dots became the distractors.

Participants

The participants were the same four who participated in the Gabor experiment.

Pilot experiment: selection of the target-to-mask SOA

SOA (stimulus onset asynchrony) refers to the interval from the onset of the stimulus to the onset of the post-stimulus mask. Collection of data in the main experiment was preceded by a brief pilot experiment to choose the SOA to be used in the main experiment. And prior even to collecting data in the pilot experiment, each participant ran separate blocks of 50 trials at each of SOA = 48 ms (36 ms stimulus exposure followed by 12-ms blank frame prior to mask), 82 ms (48-ms stimulus, 34-ms blank), 150 ms (100-ms stimulus, 50-ms blank) and 300 ms (100-ms stimulus, 200-ms blank) using homogeneous dot clouds comprising eight black dots. After this practice, a pilot experiment was conducted using exclusively the Dark-only attention condition. Each participant performed a separate block of 140 trials for each of the four SOAs (560 trials total). A block of 140 trials comprised 100 full-set trials, 20 target-only trials and 20 singleton trials.

Results of the pilot experiment are shown in Fig. 5. For SOA = 48, 82, 150 and 300 ms, the four graphs plot (for participants S1, S2, S3 and S4), response error (the mean Euclidean distance of the participant’s responses from the correct responses across the 100 full-set trials at the given SOA). For all participants, response error has declined to its asymptote by 150 ms, so this was the SOA used in the main experiment.

Main experiment: procedure

As in the Gabor experiment, general training was used for S3 and S4 but omitted for S1 and S2. Additionally, standard ϕ -specific training was used for all participants in each attention condition. As in the pilot study, experimental blocks comprised 100 full-set trials, 20 target-only trials, and 20 singleton trials in a mixed list, i.e., trials within a block were randomly sequenced.

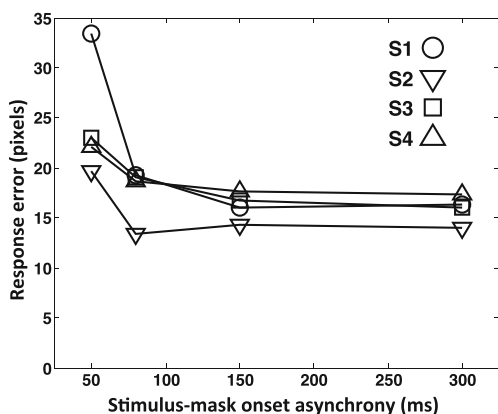


Fig. 5 Stimulus availability requirements for attention-filtered centroid judgments: Response error as a function of SOA in the Dots experiment. For each SOA (48, 82, 150, and 300 ms), each of the four participants performed a separate block of 140 trials comprising 100 full-set trials, 20 target-only trials, and 20 singleton trials in an attention condition in which the target filter assigned equal weight to all dots darker than the background (targets) and weight 0 to all dots brighter than the background (distractors). Response error is the mean Euclidean distance of the participant’s responses from the correct responses across the 100 full-set trials at the given SOA

For each participant, performance in each task showed no significant improvement from block 2 to block 3. Therefore, each participant performed just three blocks of trials in each attention condition. The first block was discarded as practice; the second and third were retained as data.

Results and discussion

The results for participants S1, S2, S3, and S4 are plotted in columns 1, 2, 3, and 4 of Fig. 6. All participants achieve strikingly different attention filters in the Dark-only versus the Light-only attention conditions. Specifically, the *F*-test described in the section entitled “Testing whether two attention filters are significantly different” yields for S1, $F_{7,780} = 140.96, p \approx 0$; for S2, $F_{7,780} = 271.01, p \approx 0$; for S3, $F_{7,780} = 111.78, p \approx 0$; for S4, $F_{7,780} = 108.85, p \approx 0$. The attention filters achieved in the Dark-only attention condition give dramatically higher weight to dots with negative Weber contrasts than they do to dots with positive Weber contrasts (on average, $5.61 \times$ greater), and the reverse is true in the Light-only attention condition ($5.78 \times$ greater). Note, however, that the Filter-fidelity values (inset in panels) tend to be smaller than the Filter-fidelity values in the Uniform and Graded attention conditions of the Gabor experiment. This reflects the fact that the attention filters achieved by all participants deviate strongly and systematically from the target filters. In particular,

1. participants are unable to completely ignore distractors (distractor weights tend to deviate positively and, in

most cases, significantly from 0 in the attention filters achieved), and

2. participants are unable to give equal weight to all target items (target items with higher absolute Weber contrasts tend to receive higher weight).

By comparison, notice that in most cases, the attention filter achieved for target-only trials tends to be more uniform across the four target items than does $f_{\phi, \text{full-set}}$. (The exceptions are S3 and S4 in the Light-only attention condition.) This suggests that one of the costs incurred in ignoring dots of the distractor polarity may be loss of sensitivity to low-salience dots of the target polarity.⁵

Each panel contains the singleton-corrected Efficiency, $\text{scEff}_{\text{full-set}}$ ($\text{scEff}_{\text{target-only}}$), which gives the proportion of dots the participant would need to include in his/her centroid computation to account for the random error in his/her responses on full-set (target-only) trials if the only source of error were missed dots. In all cases, singleton-corrected Efficiency values are high indicating that the participant can indeed deploy the attention filter he/she has achieved broadly across space with high sensitivity. In each panel $\text{scEff}_{\text{target-only}}$ is slightly higher than $\text{scEff}_{\text{full-set}}$ indicating that one cost of filtering out distractors is to inject noise into the response-production process.

Averaging across participants and the two attention conditions, response error on singleton trials was 46 % (50 %) as large as $\hat{\sigma}$ estimated from full-set (target-only) trials. This suggests that approximately $\frac{1}{4}$ of the variance in random response error on full-set and target-only trials results from error in localizing and moving the mouse to click on the selected response location; by the same token, approximately $\frac{3}{4}$ of the variance in random response error on full-set and target-only trials results from processing the stimulus and computing the centroid.

Additional performance measures

Attention filter selectivity Let f_{ϕ} be the attention filter achieved by a participant in an attention condition with a binary target filter ϕ . In this case, it is useful to define “filter selectivity” as the average of $f_{\phi}(t)$ across all target items $t \in \text{Types}$ divided by the average of $|f_{\phi}(d)|$ for distractor items $d \in \text{Types}$. For example, filter selectivities of 10 or

⁵A prominent theory of cognitive aging proposes that as people age they become less able to inhibit irrelevant information at different processing stages (Hasher and Zacks, 1988; Lustig et al., 2007). In particular, evidence is accumulating to support the claim that older adults may be specifically impaired in deploying feature-based attention (Quigley et al., 2010; Quigley & Muller, 2014). This suggests that comparison of the attention filters and Efficiencies achieved in full-set vs. target-only trials in centroid experiments such as the Dots experiment reported here may provide a useful measure of cognitive aging.

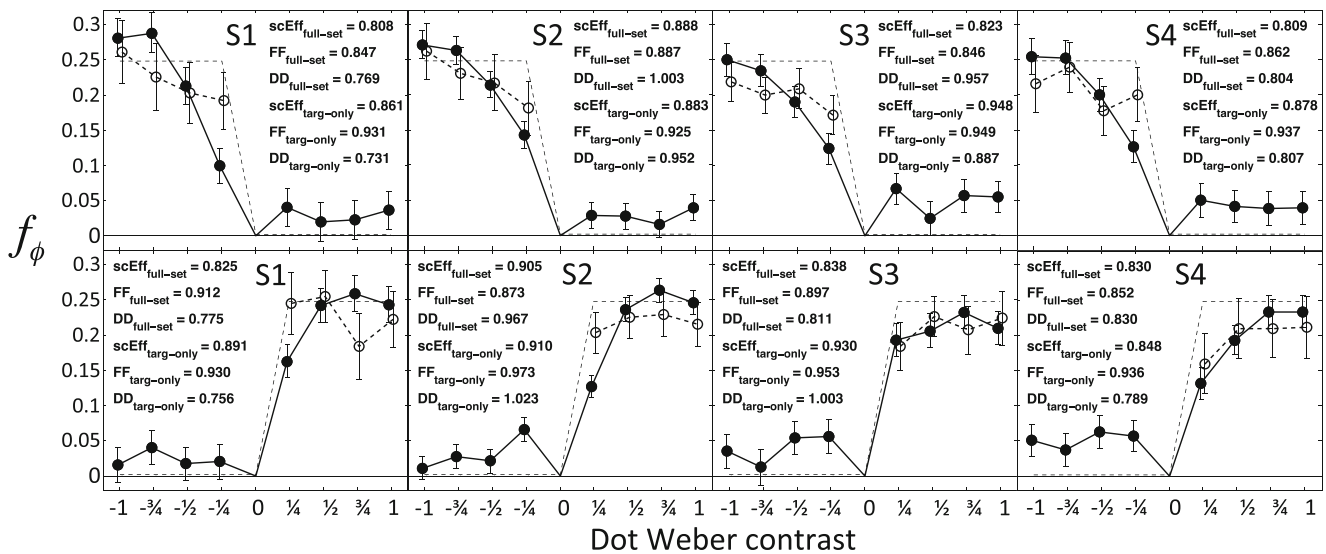


Fig. 6 Results of the Dots experiment. Top (Bottom) panels give results from the Dark-only (Light-only) attention conditions for participants S1, S2, S3, and S4 from left to right. In each panel, the thin dashed line gives the target filter ϕ , the solid line marked with filled circles gives the attention filter $f_{\phi,full-set}$ achieved by the participant in the full-set trials, and the dashed line marked with open circles gives the attention filter $f_{\phi,target-only}$ estimated from the target-only trials; because only target items are presented on target-only trials, $f_{\phi,target-only}$ is defined only across the targets in a given attention condition. To facilitate comparison, $f_{\phi,target-only}$ has been rescaled so that the sum of its values is equal to the sum of $f_{\phi,full-set}$ across just the

target items. Error bars give 95 % confidence intervals computed as described in Appendix 2. Note that $f_{\phi,full-set}$ is drawn with the solid line running through the point (0, 0). This convention reflects the fact that had invisible dots (of Weber contrast 0) been included in the stimulus cloud, their influence on the centroid would have been 0. The values of singleton-corrected Efficiency (scEff), Filter-fidelity (FF) and Data-drivenness (DD) achieved by each participant in each attention condition on full-set and target-only trials are given in each panel. All participants achieve strikingly different attention filters in the two different attention conditions. However, the attention filters achieved differ in important ways from the target filters

higher are commonly observed in attending to black versus white items (Inverso et al., [In press](#)) or red vs. green items or large vs. small items (Blair et al., 2015), and these represent highly selective attention filters. On the other hand, in the Dots experiment, filter selectivities achieved by participants S1, S2, S3 and S4 in the Dark-only (Light-only) attention condition were all lower than 10: 7.29, 7.97, 3.91 and 4.89 (9.55, 6.88, 5.28, 3.78).

The productivity function Another potentially useful descriptor (whose usefulness is not limited to attention conditions with binary target filters) is the “productivity function,” $P_{\phi}(k) = scEff_{\phi} \times f_{\phi}(k)$, $k \in Types$, where f_{ϕ} is the attention filter achieved by a given participant in the condition with target filter ϕ and $scEff_{\phi}$ is the singleton-corrected Efficiency achieved in that condition. For any item type k , $P_{\phi}(k)$ provides an estimate of the overall effectiveness with which items of type k influence responses in the centroid task in the attention condition with target filter ϕ . Insofar as f_{ϕ} characterizes the perceptual limits on information that is available to brain processes subsequent to the early centroid computation, and insofar as Efficiency $scEff_{\phi}$ characterizes cognitive limits, the productivity function P_{ϕ} is an estimate of that portion of the stimulus information that is available to subsequent brain processes.

General discussion

An attention filter is a process, initiated by a participant in the context of a task requiring feature-based attention, which operates broadly across space to modulate the relative effectiveness with which different features in the retinal input influence task performance. As we have shown, the specific task of extracting the centroid of a cloud of items can form the core of a method for deriving precise, quantitative measurements of attention filters.

The feature-similarity gain model and the centroid task

A prominent theory of FBA is the “feature-similarity gain model” (Treue & Martinez-Trujillo 1999; Martinez-Trujillo & Treue 2004). Under this theory, “...the up- or down-regulation of the gain of a sensory neuron reflects the similarity of the features of the currently behaviorally relevant target and the sensory selectivity of the neuron along all target dimensions.” Treue and Martinez-Trujillo (1999). In other words (this theory proposes), FBA will operate to amplify the responses of neurons sensitive to the attended feature and attenuate the responses of neurons insensitive to the attended feature.

The feature-similarity gain model is intended first and foremost to apply to deployments of FBA in which the participant strives to heighten the salience of a singular feature—e.g., a specific direction of motion or a specific color. It might be argued that the Highest-only and Lowest-only attention conditions in the Gabor experiment aim at FBA deployments of this sort; however, participants perform poorly at each of these tasks suggesting that human vision is devoid of neurons selective for the target feature in either of these two tasks. In the Uniform, Graded, and Inverse-graded attention conditions of the Gabor experiment, as well as the Dark-only or Light-only attention conditions in the Dots experiment, participants strive to deploy FBA in ways that heighten the salience of a range of feature values rather than a single, specific feature value.

The feature-similarity gain model might be generalized to handle FBA deployments of this sort by assuming that the gain of a given class of neurons is set in a given attention condition according to the degree to which the differential sensitivity of that neuron to the items in the display “matches the target filter” (i.e., the function ϕ in Eq. 1 used to give feedback in a particular attention condition).

There are various possible interpretations that might be given to the phrase, “matches the target filter”; however, under all such definitions, the proposed generalization of the feature-similarity gain model is likely to produce sub-optimal performance in the centroid task. In particular, consider the case in which human vision comprises (1) a particular class C_{ideal} of neurons whose sensitivity to the different items in *Types* precisely matches the target filter in a given attention condition as well as (2) other classes C_1, C_2, \dots of neurons whose sensitivity to the different items in *Types* matches the target filter less well. The generalized feature-similarity gain model would assign gain to the neurons in a given class C_k in proportion to the degree to which the activation produced in neurons in class C_k by the items in *Types* “matches the target filter.” Typically, then, various classes C_k of neurons would be likely to receive non-zero gain under the generalized feature-similarity gain model. However, under nearly all definitions of the phrase, “matches the target filter,” performance in the centroid task will be optimized by assigning full gain to neuron class C_{ideal} and zero gain to all other classes of neurons.

It is an empirical question whether the generalized feature-similarity model holds. We submit, however, that it would be surprising if human vision were committed to a general strategy so likely to produce suboptimal behavior given the available neural resources.

Beyond the centroid task

It is important to realize that the attention filters a participant can achieve for one task may differ from those

he/she can achieve for another task. A promising direction for future work is to compare the attention filters for various different sets of *Types* across different tasks. The centroid task is especially appealing because of the remarkable statistical power it confers in estimating attention filters. Nonetheless, experiments to measure attention filters using other tasks are straightforward. For example, it is easy to imagine paradigms generalizing the sorts of experiments that have been used to investigate the extraction of summary statistics from ensembles of items (Alvarez & Oliva, 2008; Alvarez 2011; Ariely, 2001; Chong & Treisman, 2003, 2005a, b).

One family of such experiments might use dots of the sort used in the Dots experiment. On each trial the stimulus is a cloud of dots (with the counts of different dot types varying across trials). In a given attention condition the participant is asked to attempt to apply a target attention filter ϕ to the cloud of dots in the display and sum the filter output across space to extract a summary statistic; if this statistic is greater than some fixed criterion, then the correct response is “A,” otherwise “B.” For example, in an attention condition analogous to the Dark-only condition in the Dots experiment, ϕ assigns the value 1 (0) to all dots of negative (positive) Weber contrast, and the criterion might be set at 9.5. In this case, the correct response would be “A” if the number of dots darker than the background was 10 or more and “B” otherwise. This rule would be used to give trial-by-trial feedback. The data would consist of (1) the matrix M whose j^{th} row contained the counts of different dot types in the stimulus on trial j , and (2) the vector R whose j^{th} entry was 1 (0) if the participant responded “A” (“B”) on trial j . A simple probit model (i.e., a general linear model with a Gaussian linking function) that uses the counts of different dot types as variables to predict the participant’s trial-by-trial classifications of stimulus clouds suffices to estimate the attention filter achieved by the participant in a given attention condition.

Will this classification task yield attention filters identical to the centroid task? The answer to this question is likely to have important consequences for our understanding of the functional architecture of human vision.

Final remarks

The centroid method described here provides guidelines developed (often through painful experience) for efficiently measuring attention filters. The two example experiments illustrate various aspects of the method. The method is distinguished by its power and simplicity:

1. *Statistical power:* The attention filter plotted in each of the 12 panels of Fig. 3 is derived from only 200 trials (~10–13 min.) The same is true for each of the full-set attention filters in Fig. 6. To achieve comparable results

using a standard psychophysical choice paradigm (e.g., Nam and Chubb (2000), Chubb and Nam (2000), and Chubb and Talevich (2002) would require 3000 or more trials). The target-only attention filters plotted in each of the four panels of Fig. 6 are based on only 40 trials; although their confidence intervals are larger, these curves are nonetheless very informative.

2. *Modeling simplicity*: To estimate an attention filter (as well as the other model parameters, $x_{default}$, $y_{default}$, V and σ) from centroid task data is easy: A simple linear model is used to predict the x - and y -coordinates of the participant’s response from the locations of different types of dots across all of the trials observed in the data set.

Acknowledgments This work was supported by NSF Award BCS-0843897.

Appendix 1: Estimating model parameters

The section, “Appendix 4. Matlab code for fitting the centroid model” includes Matlab code that implements all of the computations described in Appendices 1–3.

It is possible to rewrite Eqs. 5 and 6 as follows. For each trial j ,

$$R_x(j) = \sum_{k=1}^{N_{types}+2} W_k X_k(j) + Q_x(j) \quad \text{and} \quad (19)$$

$$R_y(j) = \sum_{k=0}^{N_{types}+2} W_k Y_k(j) + Q_y(j), \quad (20)$$

where

1. for $k = 1, 2, \dots, N_{types}$, $X_k(j)$ is the sum of the x -locations of all items of type k in the cloud on trial j , and $Y_k(j)$ is the corresponding sum of y -locations,
2. $X_{N_{types}+1}(j) = 1$, $X_{N_{types}+2}(j) = 0$,
3. $Y_{N_{types}+1}(j) = 0$, $Y_{N_{types}+2}(j) = 1$, and
- 4.

$$W_k = \begin{cases} \frac{V}{S} f_\phi(k) & \text{for } k = 1, 2, \dots, N_{types}, \\ (1 - V)x_{default} & \text{if } k = N_{types} + 1, \\ (1 - V)y_{default} & \text{if } k = N_{types} + 2. \end{cases} \quad (21)$$

Each of Eqs. 19 and 20 describes a basic linear model for which standard regression is the appropriate analysis. Moreover, both equations use the same weights W_k , $k = 0, 1, \dots, N_{types}$. Accordingly, standard matrix methods can be used to estimate the model parameters $x_{default}$, $y_{default}$, f_ϕ , V , and σ .

Let X be the $N_{trials} \times (N_{types} + 2)$ matrix whose $(j, k)^{th}$ entry is $X_k(j)$, and let Y be the corresponding matrix generated from the $Y_k(j)$ ’s. Then

1. form the full stimulus matrix M by appending the matrix Y to the bottom of the matrix X : i.e., for $j = 1, 2, \dots, 2N_{trials}$, and $k = 1, 2, \dots, N_{types} + 2$,

$$M(j, k) = \begin{cases} X_k(j) & j \leq N_{trials} \\ Y_k(j - N_{trials}) & j > N_{trials} \end{cases}, \quad \text{and} \quad (22)$$

2. form the full response vector R by appending R_y to R_x : i.e., for $j = 1, 2, \dots, 2N_{trials}$,

$$R(j) = \begin{cases} R_x(j) & j \leq N_{trials} \\ R_y(j - N_{trials}) & j > N_{trials} \end{cases}. \quad (23)$$

Then derive the weight vector \widehat{W} minimizing

$$SS_{Residual} = \|M\widehat{W} - R\|^2, \quad (24)$$

i.e., find the weight vector \widehat{W} that acts on the stimulus matrix M to best predict the observed response vector R . In fact, \widehat{W} can be derived via linear regression.

Specifically, for M the matrix given by Eq. 22 and R the vector given by Eq. 23, the linear regression model assumes that

$$R = MW + Q \quad (25)$$

where Q is a $2N_{trials} \times 1$ vector whose entries are jointly independent Gaussian random variables all with mean 0 and standard deviation σ . Under these assumptions, the maximum likelihood estimate of W is given by

$$\widehat{W} = M^\dagger R = M^\dagger (MW + Q) = W + M^\dagger Q \quad (26)$$

where M^\dagger denotes the (left-side) pseudoinverse of M . I.e., M^\dagger satisfies $M^\dagger M = I$ (where I is the $(N_{types} + 2) \times (N_{types} + 2)$ identity matrix).

Given \widehat{W} , estimates of V and f_ϕ , $x_{default}$ and $y_{default}$ are obtained as follows:

$$\widehat{V} = \sum_{k=1}^{N_{types}} \widehat{W}_k n_k \quad (27)$$

(where n_k is the number of items of type k occurring in each display). It is easy to check that

$$E[\widehat{V}] = \sum_{k=1}^{N_{types}} E[\widehat{W}_k] n_k = \sum_{i=1}^{N_{types}} \frac{V}{S} f_\phi(k) n_k = V, \quad (28)$$

where S is given by Eq. 4.

To estimate f_ϕ , we take

$$\widehat{f}_\phi(k) = \frac{\widehat{W}_k}{\sum_{j=1}^{N_{types}} \widehat{W}_j} \quad \text{for } k = 1, 2, \dots, N_{types}, \quad (29)$$

to insure (for plotting purposes) that \widehat{f}_ϕ sums to 1. Note that there is no aspect of the modeling procedure that excludes negative attention filter values $\widehat{f}_\phi(k)$ for some item types k .

Finally, the following statistics are used to estimate $x_{default}$ and $y_{default}$:

$$\hat{x}_{default} = \frac{\widehat{W}_{N_{types}+1}}{(1 - \widehat{V})} \quad \text{and} \quad \hat{y}_{default} = \frac{\widehat{W}_{N_{types}+2}}{(1 - \widehat{V})}. \quad (30)$$

Appendix 2: Estimating confidence intervals for model parameters

Throughout this section, it will be convenient to write N_W for $N_{types} + 2$ (the number of weights W_k in Eqs. 19 and 20).

Point estimates of the model parameters V , f_ϕ , and σ are given by Eqs. 27, 29, and 11, and estimates of both $x_{default}$, $y_{default}$ are given by Eq. 30. This section of the appendix shows how to derive confidence intervals for V and for $f_\phi(k)$, $k = 1, 2, \dots, N_{types}$. The parameters $x_{default}$ and $y_{default}$ are included in the model primarily to enable estimation of V and are of little interest in themselves; moreover, for high values of V , the estimates of $x_{default}$ and $y_{default}$ become highly volatile. For these reasons, confidence intervals are not supplied for these two parameters.

The model used to derive these estimates assumes Eq. 25 and applies Eq. 26 to derive estimates \widehat{W}_k , of the regression weights W_k , for $k = 1, 2, \dots, N_W$.

By singular value decomposition, we can express M^\dagger as

$$CHL^T = M^\dagger \quad (31)$$

where C is the $N_W \times N_W$ matrix of “principal components,” H is $N_W \times N_W$ diagonal matrix of corresponding eigenvalues, and L is the column orthonormal matrix whose k^{th} row gives the “loadings” of the different principal components contributing to the k^{th} column of matrix M^\dagger .

The matrix

$$\Theta = E \left[M^\dagger Q (M^\dagger Q)^T \right] = \sigma^2 C H^2 C^T \quad (32)$$

is the covariance matrix that characterizes the ellipsoidal dispersion of the multivariate Gaussian noise that perturbs the estimate \widehat{W} away from its expectation W .

The random variables \widehat{W}_i , $i = 1, 2, \dots, N_W$ are jointly normal with means W_i and covariance matrix Θ . A fundamental result (see, e.g., Graybill (1961), Theorem 3.6, p. 56) is captured in the following

One-variable linear combination theorem

Let random variables X_1, X_2, \dots, X_N be jointly normal with expectations $\mu_1, \mu_2, \dots, \mu_N$ and covariance matrix

Γ . Then for any vector v of length N , the random variable

$$G_v = \sum_{i=1}^N v_i X_i \quad (33)$$

is normally distributed with

$$E[G_v] = \sum_{k=1}^N v_k \mu_k \quad (34)$$

and variance

$$\text{var}[G_v] = \sum_{j=1}^N \sum_{k=1}^N \Gamma_{jk} v_j v_k = v^T \Gamma v. \quad (35)$$

This theorem is actually a special case of the following

Two-variable linear combination theorem

Let X_1, X_2, \dots, X_N be jointly normal with expectations $\mu_1, \mu_2, \dots, \mu_N$ and covariance matrix Γ . For any vectors v and w of length N , the random variables

$$G_v = \sum_{i=1}^N v_i X_i \quad \text{and} \quad G_w = \sum_{i=1}^N w_i X_i \quad (36)$$

are jointly normal with

$$E[G_v] = \sum_{k=1}^N v_k \mu_k \quad \text{and} \quad E[G_w] = \sum_{k=1}^N w_k \mu_k \quad (37)$$

and 2×2 covariance matrix Υ with entries

$$\Upsilon_{11} = \text{var}[G_v] = \sum_{j=1}^N \sum_{k=1}^N \Gamma_{jk} v_j v_k = v^T \Gamma v, \quad (38)$$

$$\Upsilon_{22} = \text{var}[G_w] = \sum_{j=1}^N \sum_{k=1}^N \Gamma_{jk} w_j w_k = w^T \Gamma w, \quad \text{and} \quad (39)$$

$$\Upsilon_{12} = \Upsilon_{21} = \text{cov}[G_v, G_w] = \sum_{j=1}^N \sum_{k=1}^N \Gamma_{jk} v_j w_k = v^T \Gamma w. \quad (40)$$

The confidence interval for V

The linear combination theorem implies that our estimate \widehat{V} of Data-drivenness (Eq. 27) is normally distributed with mean V (as shown in Eq. 28) and variance

$$\text{var}_V = \sigma^2 \eta^T \Theta \eta \quad (41)$$

where η is the $N_W \times 1$ vector

$$\eta_k = \begin{cases} n_k & k = 1, 2, \dots, N_{types} \\ 0 & k > N_{types} \end{cases} \quad (42)$$

for n_k the number of items of type k occurring in each full-set stimulus cloud. Thus, the random variable

$$Z_V = \frac{\widehat{V} - V}{(\sigma^2 \eta^T \Theta \eta)^{\frac{1}{2}}} \tag{43}$$

has a standard normal distribution. However, the true value of σ is unknown. If instead, the unbiased estimator $\widehat{\sigma}$ (given by Eq. 11) is used in the denominator, we obtain a statistic

$$T_V = \frac{\widehat{V} - V}{(\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}}} \tag{44}$$

that has a t -distribution with $r = 2N_{trials} - N_{types} - 2$ degrees of freedom. Writing F_r^{-1} for the inverse of the t -cdf with r degrees of freedom, we note that the critical value of T_V is given by

$$t_{crit} = F_r^{-1} \left(\frac{p_{crit} + 1}{2} \right) \tag{45}$$

That is,

$$\mathbb{P}[-t_{crit} < T_V < t_{crit}] = p_{crit}, \tag{46}$$

from which it follows that

$$\mathbb{P} \left[-(\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}} t_{crit} - \widehat{V} < -V < (\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}} t_{crit} - \widehat{V} \right] = p_{crit}, \tag{47}$$

and hence that

$$\mathbb{P} \left[\widehat{V} - (\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}} t_{crit} < V < \widehat{V} + (\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}} t_{crit} \right] = p_{crit}, \tag{48}$$

yielding a p_{crit} confidence interval of $\widehat{V} \pm (\widehat{\sigma}^2 \eta^T \Theta \eta)^{\frac{1}{2}} t_{crit}$ for V .

Confidence intervals for f_ϕ

The problem of computing confidence intervals for f_ϕ is complicated by the fact that \widehat{f}_ϕ is constrained to sum to 1 by Eq. 29. The following result (Fieller, 1954) is directly applicable.

The ratio theorem

Let X and Y be jointly normal random variables with expectations μ_X and μ_Y , variances $v_{11}\sigma^2$, $v_{22}\sigma^2$ and covariance $v_{12}\sigma^2$. Then, if v_{11} , v_{22} and v_{12} are all known, and if μ_Y is

positive and large in comparison to σ , a $1 - \alpha$ confidence interval for the ratio $\frac{\mu_X}{\mu_Y}$ is given by

$$(m_L, m_U) = \frac{1}{1-g} \left[\frac{X}{Y} - \frac{g v_{12}}{v_{22}} \pm \frac{t_{r,\alpha}}{Y} \sqrt{v_{11} - 2 \frac{X}{Y} v_{12} + \frac{X^2}{Y^2} v_{22} - g \left(v_{11} - \frac{v_{12}^2}{v_{22}} \right)} \right] \tag{49}$$

where

$$g = \frac{t_{r,\alpha}^2 \widehat{\sigma}^2 v_{22}}{Y^2} \tag{50}$$

for $\widehat{\sigma}$ the unbiased estimator of σ and $t_{r,\alpha} = F_r^{-1} \left(\frac{1+\alpha}{2} \right)$ (where, as above, F_r^{-1} denotes the inverse of the t -cdf with r degrees of freedom).

To derive confidence intervals for $f_\phi(k)$, we note that for $k = 1, 2, \dots, N_{types}$, the antecedent conditions of the ratio theorem are satisfied by taking the random variables

$$X = \widehat{W}_k = \sum_{j=1}^{N_W} \widehat{W}_j \xi_j^k, \tag{51}$$

and

$$Y = \sum_{j=1}^{N_{types}} \widehat{W}_j = \sum_{j=1}^{N_W} \widehat{W}_j U_j, \tag{52}$$

where ξ^k and U are the $N_W \times 1$ vectors whose j^{th} components are given by

$$\xi_j^k = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad U_j = \begin{cases} 1 & j = 1, 2, \dots, N_{types} \\ 0 & \text{otherwise} \end{cases}. \tag{53}$$

Then the one-variable linear combination theorem implies, for Θ given by Eq. 32, that

$$\eta_{11} = \frac{1}{\sigma^2} (\xi^k)^T \Theta \xi^k = (\xi^k)^T C H^2 C^T \xi^k \quad \text{and} \tag{54}$$

$$\eta_{22} = U^T C H^2 C^T U,$$

and the two-variable linear combination theorem implies that

$$\eta_{12} = (\xi^k)^T C H^2 C^T U. \tag{55}$$

In the current application of the ratio theorem, the unbiased estimator $\widehat{\sigma}$ of σ is given by Eq. 11, and the degrees of freedom r is equal to $2 \times N_{trials} - (N_{types} + 2)$. Thus, all the values required to evaluate Eq. 49 are in hand.

Appendix 3: Computing Efficiency

This Appendix shows how to compute $\text{Efficiency} = 1 - p_{miss}$ (See “Measuring resistance to residual error: Efficiency.”). Roughly speaking, p_{miss} is the proportion of items by which stimulus clouds would need to be decimated in order to yield the observed value of $\hat{\sigma}$ (as given by Eq. 11) if this decimation were the only source of random error in the participant’s centroid computation. (In the case in which Efficiency is “singleton-corrected,” $\sigma_{SC} = \sqrt{\hat{\sigma}^2 - \hat{\sigma}_{singleton}^2}$ is used instead of $\hat{\sigma}$, where $\hat{\sigma}_{singleton}$ is the unbiased estimator of the standard deviation of the response errors (in each of x and y) observed on singleton trials.) p_{miss} could of course be estimated by simulation. Here we present an algorithm to compute p_{miss} exactly.

Notation

For $j = 1, 2, \dots, N_{types}$, let n_j be the number of items of type j in each stimulus cloud. Let N be the total number of items in each stimulus display, and for $i = 1, 2, \dots, N$, let τ_i be the type of the i^{th} item in the stimulus display, and let (X_i, Y_i) be its location. We write \tilde{N} for the set of integers $1, 2, \dots, N$. For any of the integers $L = 0, 1, \dots, N - 1$, we write α_L for a random subset of \tilde{N} of size $N - L$ (i.e., α_L is derived by removing a random set of L items from \tilde{N}). For any integer $L = 0, 1, \dots, N - 1$, we define P_L , the proportion of L ost items from a stimulus of N items, as

$$P_L = \frac{L}{N}. \tag{56}$$

Assumptions about stimulus cloud locations

The algorithm to compute p_{miss} makes several assumptions about the locations of items in a stimulus cloud. All of the x - and y -coordinates, $X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_N$, of the stimulus cloud locations are assumed to be identically distributed random variables with mean 0 and variance η^2 . The X_i ’s (Y_i ’s) are also assumed to be pairwise uncorrelated. That is, the expected value of the product of the x -coordinates (and of the y -coordinates) of two stimulus items is

$$E[X_i X_j] = \begin{cases} \eta^2 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \quad \text{and} \quad E[Y_i Y_j] = \begin{cases} \eta^2 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

Note in particular that these assumptions are satisfied by stimulus clouds

1. whose locations are drawn independently from a circular, bivariate Gaussian density whose horizontal and vertical standard deviations are both η .
2. whose locations are produced as described in “Generating full-set stimulus clouds” so as to have fixed dispersion η .

The hypothetical response process used to derive p_{miss}

The response process used to derive p_{miss} mirrors the model of Eqs. 5 and 6. However, it assumes that all random response error is due to random removal of items from the stimulus prior to computing the centroid. Let $x_{default}$ and $y_{default}$ be real numbers, let V be a Data-drivenness value (i.e., $0 < V < 1$), let $\ddot{f} : Types \rightarrow \mathbb{R}$ be an attention filter, and set

$$f = \frac{\ddot{f}}{\sum_{i \in \tilde{N}} \ddot{f}(\tau_i)}. \tag{58}$$

Because \ddot{f} is an attention filter, it is constrained (as a matter of convention) to satisfy $\sum_{i=1}^{N_{types}} \ddot{f}(\tau_i) = 1$. The function f will not generally satisfy this condition. However, f has the convenient property that the x - and y -coordinates of the response to an undecimated cloud (i.e., a cloud from which 0 items have been removed) in our hypothetical experiment are given by

$$\begin{aligned} x\text{-coordinate} &= \sum_{i \in \tilde{N}} f(\tau_i) X_i \quad \text{and} \\ y\text{-coordinate} &= \sum_{i \in \tilde{N}} f(\tau_i) Y_i. \end{aligned} \tag{59}$$

For any subset A of \tilde{N} let

$$S_A = \sum_{i \in A} f(\tau_i). \tag{60}$$

Then the random variables

$$R_X(L) = \frac{V}{S_{\alpha_L}} \sum_{i \in \alpha_L} f(\tau_i) X_i + (1 - V)x_{default} \quad \text{and} \tag{61}$$

$$R_Y(L) = \frac{V}{S_{\alpha_L}} \sum_{i \in \alpha_L} f(\tau_i) Y_i + (1 - V)y_{default} \tag{62}$$

represent the x - and y -coordinates of a response that is produced by

1. decimating the stimulus by L items,
2. replacing each of the remaining items i in the display (i.e., each $i \in \alpha_L$) by a “penny-pile” of size $f(\tau_i)$,
3. deriving the centroid (C_X, C_Y) of this filtered, decimated cloud, and
4. selecting the point that lies proportion V of the way from $(x_{default}, y_{default})$ to (C_X, C_Y) .

This process is very similar to the process described in Eqs. 5 and 6 by which the participant is assumed to produce his/her response on a given trial j . The key differences are

1. the summation in each of Eqs. 5 and 6 is over all items in the display whereas the summation in each of Eqs. 61 and 62 is over a random subset of $N - L$ items from the original stimulus display, and
2. Equations 5, 6 include the additive noise term $Q_x(j)$ ($Q_y(j)$). These terms are lacking from Eqs. 61 and 62.

The definition of p_{miss}

For $L = 0, 1, \dots, N - 1$, define

$$\begin{aligned} \text{Var}_{miss}(P_L) &= \text{var}[R_X(L) - R_X(0)] \\ &= \text{var}[R_Y(L) - R_Y(0)]. \end{aligned} \tag{63}$$

That is, $\text{Var}_{miss}(P_L)$ is the variance of the difference (in either the x - or the y -coordinate) between the response produced using a stimulus cloud with L randomly chosen items removed vs. the response produced using a full stimulus cloud.

As is evident, $\text{Var}_{miss}(P_L)$ is an increasing function of the proportions P_L corresponding to $L = 0, 1, \dots, N - 1$. We extend this function to any $0 < p \leq P_{N-1}$ by linearly interpolating between the values $\text{Var}_{miss}(P_L)$ as follows. For any probability $p < P_{N-1}$, let L_p be the greatest integer less than or equal to Np and define

$$\begin{aligned} \text{Var}_{miss}(p) &= \text{Var}_{miss}(P_{L_p}) \\ &+ (Np - L_p)[\text{Var}_{miss}(P_{L_p+1}) - \text{Var}_{miss}(P_{L_p})]. \end{aligned} \tag{64}$$

It is not hard to show that this definition is equivalent to setting

$$\text{Var}_{miss}(p) = \text{var}[R_X(\lambda_p) - R_X(0)] \tag{65}$$

where λ_p is the integer-valued random variable defined as follows:

$$\lambda_p = \begin{cases} L_p & \text{with probability } 1 - \beta_p, \\ L_p + 1 & \text{with probability } \beta_p, \end{cases} \tag{66}$$

for

$$\beta_p = Np - L_p. \tag{67}$$

The function $\text{Var}_{miss}(p)$ is strictly increasing over the set of all p such that $0 < p \leq P_{N-1}$. Therefore, the inverse function Var_{miss}^{-1} is well defined. Accordingly, for any given value of the model parameter $\sigma^2 < \text{Var}_{miss}(P_{N-1})$ characterizing the random variables $Q_x(j)$ and $Q_y(j)$ in Eqs. 5 and 6, we now define

$$P_{miss}(\sigma) = \text{Var}_{miss}^{-1}(\sigma^2). \tag{68}$$

Specifically, Eq. 64 implies that

$$P_{miss}(\sigma) = \frac{1}{N} \left(L_\sigma + \frac{\sigma^2 - \text{Var}_{miss}(P_{L_\sigma})}{\text{Var}_{miss}(P_{L_\sigma+1}) - \text{Var}_{miss}(P_{L_\sigma})} \right) \tag{69}$$

where L_σ is the greatest integer such that $\text{Var}_{miss}(P_{L_\sigma}) < \sigma^2$. To reiterate, $P_{miss}(\sigma)$ is the proportion of items by which the stimulus cloud must be decimated in order for the variance of the difference between the x -coordinate (the same goes for the y -coordinate) of the response produced using a decimated stimulus cloud vs. the response produced using a full stimulus cloud to be equal to σ^2 .

Finally, we define the statistic p_{miss} characterizing the performance of a participant in a given centroid task condition by

$$p_{miss} = P_{miss}(\hat{\sigma}) \tag{70}$$

for $\hat{\sigma}$ given by Eq. 11.

Computing p_{miss}

As is clear from Eqs. 69 and 70, if we can compute $\text{Var}_{miss}(P_L)$ for any $L = 0, 1, \dots, N - 1$, then we can

1. find the greatest integer $L_{\hat{\sigma}}$ such that $\text{Var}_{miss}(P_{L_{\hat{\sigma}}}) \leq \hat{\sigma}^2$, and
2. compute $p_{miss} = P_{miss}(\hat{\sigma})$ using Eq. 69.

Accordingly, the remainder of this section develops an algorithm to compute $\text{Var}_{miss}(P_L)$.

For $L = 0, 1, \dots, N - 1$, note that

$$\begin{aligned} R_X(L) - R_X(0) &= \frac{V}{S_{\alpha_L}} \sum_{i \in \alpha_L} f(\tau_i) X_i + (1 - V)x_{default} \\ &- \left(V \sum_{i \in \tilde{N}} f(\tau_i) X_i + (1 - V)x_{default} \right) \\ &= V Z_X(L) \end{aligned} \tag{71}$$

for

$$Z_X(L) = \frac{1}{S_{\alpha_L}} \sum_{i \in \alpha_L} f(\tau_i) X_i - \sum_{i \in \tilde{N}} f(\tau_i) X_i. \tag{73}$$

(Of course, there exists a corresponding random variable $Z_Y(L)$ identically distributed to $Z_X(L)$.) Thus, $\text{Var}_{miss}(P_L)$ is given by

$$\text{Var}_{miss}(P_L) = V^2 \text{var}(Z_X(L)), \tag{74}$$

which follows from Eq. 72.

Our solution to the problem of computing $\text{Var}_{miss}(P_L)$ depends crucially on computing the variance of $Z_X(L)$. Toward this end, writing $\tilde{\alpha}_L$ for the complement of α_L , note that $Z_X(L)$ can be rewritten as follows to aggregate all the

X_i 's for $i \in \alpha_L$ in one sum and all the X_i 's for $i \in \bar{\alpha}_L$ in the other:

$$Z_X(L) = \frac{1 - S_{\alpha_L}}{S_{\alpha_L}} \sum_{i \in \alpha_L} f(\tau_i) X_i - \sum_{i \in \bar{\alpha}_L} f(\tau_i) X_i = \sum_{i=1}^N W_i X_i \tag{75}$$

for

$$W_i = \begin{cases} \left(\frac{1 - S_{\alpha_L}}{S_{\alpha_L}}\right) f(\tau_i) & i \in \alpha_L \\ -f(\tau_i) & i \in \bar{\alpha}_L \end{cases} \tag{76}$$

Moreover, for $i = 1, 2, \dots, N$, $E[X_i] = 0$, and X_i and W_i are independent; hence $E[Z_X(L)] = 0$, from which it follows that

$$\begin{aligned} \text{var}(Z_X(L)) &= E \left[\left(\sum_{i \in \tilde{N}} W_i X_i \right)^2 \right] \\ &= \sum_{i \in \tilde{N}} \sum_{j \in \tilde{N}} E[W_i W_j] E[X_i X_j]. \end{aligned} \tag{77}$$

But for all $i, j \in \tilde{N}$, if $i \neq j$, then $E[X_i X_j] = 0$, and otherwise $E[X_i^2] = \eta^2$. Hence,

$$\begin{aligned} \text{var}(Z_X(L)) &= \eta^2 \sum_{i=1}^N E[W_i^2] \\ &= \eta^2 E \left[\sum_{i \in \alpha_L} \left(\frac{1 - S_{\alpha_L}}{S_{\alpha_L}} f(\tau_i) \right)^2 + \sum_{i \in \bar{\alpha}_L} f^2(\tau_i) \right]. \end{aligned} \tag{78}$$

It follows that

$$\text{var}(Z_X(L)) = \eta^2 E[\Gamma(\alpha_L)] \tag{79}$$

where the function $\Gamma(A)$ is defined as follows for any $A \subset \tilde{N}$:

$$\Gamma(A) = \frac{1}{S_A} \left(\frac{1}{S_A} - 2 \right) \sum_{i \in A} f^2(\tau_i) + \sum_{i \in \bar{A}} f^2(\tau_i). \tag{80}$$

Thus, if we can compute $E[\Gamma(\alpha_L)]$, we will be able to compute $\text{Var}_{miss}(L)$. Toward this end, call any vector $\vec{k} = (k_1, k_2, \dots, k_{N_{types}})$ an L -count vector if $k_j \in \{0, 1, \dots, n_j\}$ for $j = 1, 2, \dots, N_{types}$, and

$$\sum_{j=1}^{N_{types}} k_j = N - L. \tag{81}$$

That is, an L -count vector describes the numbers of different item types remaining after L items have been removed. For any subset $A \subset \tilde{N}$, if k_j gives the number of integers $h \in A$ for which $\tau_h = j$, then \vec{k} is called the *count vector* of A . Note that in this case

$$S_A = \sum_{j=1}^{N_{types}} k_j f(j) \tag{82}$$

and

$$\Gamma(A) = \frac{1}{S_A} \left(\frac{1}{S_A} - 2 \right) \sum_{j=1}^{N_{types}} k_j f^2(j) + \sum_{j=1}^{N_{types}} n_j f^2(j). \tag{83}$$

As is easy to check from Eqs. 82 and 83, if subsets $A, A' \subset \tilde{N}$, share the same count vector, then $\Gamma(A) = \Gamma(A')$. Accordingly, set $\gamma(\vec{k}) = \Gamma(A)$ for $A \subset \tilde{N}$ with L -count vector \vec{k} and observe that

$$E[\Gamma(\alpha_L)] = \sum_{\substack{\text{all } L\text{-count} \\ \text{vectors } \vec{k}}} \gamma(\vec{k}) \mathbf{P}(\vec{k}) \tag{84}$$

where $\mathbf{P}(\vec{k})$ is the probability that a subset α_L derived by randomly sampling $N - L$ integers from \tilde{N} without replacement will have L -count vector \vec{k} . This probability is given by

$$\mathbf{P}(\vec{k}) = \frac{\prod_{j=1}^{N_{types}} \binom{n_j}{k_j}}{\binom{N}{N-L}}. \tag{85}$$

To see this, note first that the denominator gives the total number of subsets of \tilde{N} of size $N - L$, and the numerator gives the total number of those subsets that have, for $j = 1, 2, \dots, N_{types}$, exactly k_j of the n_j integers $i \in \tilde{N}$ for which $\tau_i = j$.

Computing $\text{Var}_{miss}(P_L)$

To compute $\text{Var}_{miss}(P_L)$ for $L = 0, 1, \dots, N - 1$, we

1. Use the Matlab program called `GetAllLCountVectors.m` (included in ‘‘Appendix 4. Matlab code for fitting the centroid model’’) that takes as input arguments the integer L , and the vector $\vec{n} = (n_1, n_2, \dots, n_{N_{types}})$ whose entries give the numbers of different types of items in a given stimulus cloud and returns a matrix K whose columns are all the L -count vectors consistent with \vec{n} .
2. For each of the vectors $\vec{k} = K(:, j)$ in this array
 - (a) use Eqs. 82 and 83 to compute $\gamma(\vec{k}) = \Gamma(A)$ for some $A \subset \tilde{N}$ with L -count vector \vec{k} ,
 - (b) use Eq. 85 to derive $\mathbf{P}(\vec{k})$.
3. Use Eq. 84 to compute $E[\Gamma(\alpha_L)]$.
4. Derive $\text{var}(Z_X(L))$ using Eq. 79, and
5. compute $\text{Var}_{miss}(P_L)$ using Eq. 74.

The algebraic derivation of p_{miss} described above has been confirmed by Monte Carlo simulation.

Limitations of the Efficiency computations

The algorithm for computing Efficiency ($= 1 - p_{miss}$) has the following limitations:

1. For a given stimulus cloud size N_{stim} , the minimum Efficiency that can be estimated is $\frac{1}{N_{stim}}$. This is a consequence of the fact that a decimated display must contain at least a single item for a centroid to be computable. In the case of an experimental design that uses stimulus clouds with fixed dispersion D , an Efficiency of $\frac{1}{N_{stim}}$ corresponds to a value of $\hat{\sigma} = D$ (the standard deviation of the x - or y -coordinate of a single display item). Two consequences of this observation are that
 - (a) if $\hat{\sigma} > D$, it is impossible to compute Efficiency. In this case, our software assigns to Efficiency the value NaN (which stands for “not a number” in Matlab).
 - (b) if $N_{stim} = 1$, it is impossible to compute Efficiency.
2. Efficiency may become practically impossible to compute if displays become too complicated in the sense that
 - (a) the number of different types of items contained in a display becomes too large, and/or

- (b) the total number of items contained in a display becomes too large.

Specifically, a bottleneck can arise in computing $E[\Gamma(\alpha)]$ (Eq. 84). In particular, the number of L -count vectors can explode for values of L near half the total number of items in a display. For example, if full-set displays contain 3 each of 12 different types of items, then the number of different L -count vectors is 12 for $L = 1$ (and $L = 35$); 78 for $L = 2$ (and $L = 34$); 364 for $L = 3$ (and $L = 33$); \dots ; 766,272 for $L = 13$ (and $L = 23$); 1,024,464 for $L = 14$ (and $L = 22$); \dots , 1,650,792 for $L = 17$ (and $L = 19$); and 1,703,636 for $L = 18$. This means that for complicated displays, it may still be possible to algebraically compute high and low Efficiencies but impossible to compute Efficiencies near $\frac{1}{2}$.

When displays become too complicated to allow an exact computation of Efficiency, one can still estimate Efficiency by using Monte Carlo simulation.

Appendix 4: Matlab code for fitting the centroid model

The main program: FitCentroidModel.m

```
function Results = FitCentroidModel(ItemTypes,ItemLocsX,ItemLocsY,...
    ResponseX,ResponseY,TargetFilter,SingleDotStdDev)

% Results = FitCentroidModel(ItemTypes,ItemLocsX,ItemLocsY,...
%                               ResponseX,ResponseY,TargetFilter,SingleDotStdDev)
%
% Results is a structure with fields
%
% Results.Types -- A list of the unique values in ItemTypes
% Results.TargetFilter -- Filter used to give feedback in this condition
% Results.ItemTypes -- See below
% Results.ItemLocsX -- See below
% Results.ItemLocsY -- See below
% Results.ResponseX -- See below
% Results.ResponseY -- See below
% Results.SingleDotStdDev -- See below
% Results.AttentionFilter -- The estimated attention filter
% Results.AttentionFilter95pctCI -- Corresponding 95% confidence intervals
% Results.DataDrivenness -- The estimated value of Data-drivenness
% Results.DataDrivenness95pctCI -- Corresponding 95% confidence intervals
% Results.x_default -- The estimated value of model parameter x_default
% Results.y_default -- The estimated value of model parameter y_default
% Results.SigmaHat -- The estimated value of model parameter Sigma
% Results.Efficiency -- The estimated value of Efficiency
```

```

% Results.FilterFidelity -- The estimated value of Filter-fidelity
% Results.Residual -- The vector of differences between predicted and
%                   observed responses. The first NTrials entries of this
%                   2*NTrials by 1 column vector give the differences between
%                   predicted and observed x-coordinates; the last NTrials
%                   entries give the differences between predicted and observed
%                   y-coordinates.

% Let NStim be the number of items presented in a stimulus cloud, and let
% NTrials be the number of trials in the data set.
%
% ItemTypes is an NStim by NTrials matrix. ItemTypes(r,c) gives the type
% of the rth item in the cloud presented on trial c. ItemLocsX and
% ItemLocsY are both NStim by NTrials matrices. ItemLocsX(r,c) and
% ItemLocsY(r,c) give the x- and y-coordinates of the location of the rth
% item in the cloud presented on trial c. ResponseX(c) and ResponseY(c)
% give the x- and y-coordinates of the participant's response on trial c.
%
% TargetFilter is an optional argument. If provided, it is used to compute
% the FilterFidelity, which is a measure of how closely the attention
% filter achieved by the participant matches the TargetFilter.
%
% SingleDotStdDev is an optional argument. If it is provided, it is
% assumed to be an estimate of the response error standard deviation (in
% each of the x- and y-coordinates) derived from trials in which the
% stimulus comprised a single dot whose location is distributed identically
% to the location of the correct response on a full-set trial.
% Specifically, for SDRX and SDRY, the vectors of the x- and y-coordinates
% of the participant's responses to single dot stimuli and SDX and SDY the
% actual locations of the single-dot stimuli, it is assumed that
%
% SingleDotStdDev = sqrt(mean([SDRX(:)-SDX(:); SDRY(:)-SDY(:)].^2));
%
% SingleDotStdDev^2 is subtracted from the full-set trial response error
% variance prior to estimating pmiss in order to derive an estimate of
% Efficiency that discounts motor response error.

%% Set up for analysis
if nargin < 7 || isempty(SingleDotStdDev)
    SingleDotStdDev = 0;
end
if nargin < 6
    TargetFilter = [];
end
Crit = 0.95;    % We will derive 95% confidence intervals for Attention
                % filter values as well as Data-drivenness.

[NStim,NTrials] = size(ItemTypes); % NStim is the number of items in a

```

```

% stimulus cloud; NTrials is the
% number of trials in the data set.

Types = unique(ItemTypes(:));
NTypes = length(Types);
ItemCountsPerCloud = NaN(size(Types));
for k=1:NTypes
    ItemCountsPerCloud(k) = sum(ItemTypes(:,1)==Types(k));
end

R = [ResponseX(:);ResponseY(:)]; % R as defined by Eq. 23 of the paper

X = NaN(NTrials,NTypes); % As specified in "Appendix 1: Estimating model
% parameters," for k=1,2,...,NTypes, X(j,k) will
% hold the sum of the x-coordinates of all items
% of type Types(k) in the stimulus cloud
% presented on trial j.

Y = NaN(NTrials,NTypes); % and Y(j,k) will hold the sum of the
% y-coordinates of all items of type Types(k) in
% the stimulus cloud presented on trial j.

for h=1:NTypes
    X(:,h) = sum(ItemLocsX.*(ItemTypes==Types(h)))';
    Y(:,h) = sum(ItemLocsY.*(ItemTypes==Types(h)))';
end

Ones = ones(NTrials,1);
Zeros = zeros(NTrials,1);
X = [X Ones Zeros]; % X is then extended to include a column vector
% of ones followed by a column vector of zeros,
Y = [Y Zeros Ones]; % and Y is extended to include a column vector
% of zeros followed by a column vector of ones,

M = [X;Y]; % M as defined by Eq. 22 of the paper

%% Perform regression analysis and estimate model parameters
PinvM = pinv(M);
WWhat = PinvM*R; % Eq. 26 of the paper
DataDrivenness = WWhat(1:NTypes)'*ItemCountsPerCloud(:); % Paper Eq. 27
AttentionFilter = WWhat(1:NTypes)/sum(WWhat(1:NTypes)); % Paper Eq. 29
x_default = WWhat(NTypes+1)/(1-DataDrivenness); % Paper Eq. 30
y_default = WWhat(NTypes+2)/(1-DataDrivenness); % Paper Eq. 30

Residual = R - M*WWhat;
NFreeParams = NTypes + 2; % Parameters other than the coordinate values of
% AttentionFilter are x_default, y_default and
% DataDrivenness; however, the AttentionFilter
% is normalized to sum to 1; thus it consumes
% only NTypes-1 degrees of freedom.

df = 2*NTrials-NFreeParams;

```

```

UnbiasedVarEstimate = sum(Residual.^2)/df;
SigmaHat = sqrt(UnbiasedVarEstimate);

%% Derive confidence intervals for DataDrivenness and AttentionFilter
[C,H,L]=svd(PinvM,'econ');      % Paper Eq. 31
CovMat = (C*H)*(C*H)';
CovMatFilterOnly = CovMat(1:NTypes,1:NTypes);
VarDataDrivenness = UnbiasedVarEstimate*ItemCountsPerCloud(:)*...
    CovMatFilterOnly*ItemCountsPerCloud(:); % Paper Eq. 41
StdError = sqrt(VarDataDrivenness);
CI = tinv((Crit+1)/2,df)*StdError;          % Paper Eq. 45
DataDrivennessCIs = [DataDrivenness - CI, DataDrivenness + CI];
                                                    % Paper Eq. 48

AttentionFilterCIs = NaN(NTypes,2);
for k=1:NTypes
    [LowerCI,UpperCI] = GetAttentionFilterCIs(k,WHat,... % Implements
        CovMatFilterOnly,Crit,df,SigmaHat); % Paper Eq. 49
    AttentionFilterCIs(k,1) = LowerCI;
    AttentionFilterCIs(k,2) = UpperCI;
end

%% Compute Mismatch (= 1-FilterFidelity)
if ~isempty(TargetFilter)
    Mismatch = GetMismatch(ItemCountsPerCloud,...
        AttentionFilter, TargetFilter); % Yields 1-FilterFidelity
        % (See paper Eq. 8)
else
    Mismatch = [];
end

%% Compute PMiss (= 1-Efficiency)
if SingleDotStdDev < SigmaHat
    DDSqueezed = min(max(DataDrivenness,0),1); % Insure that DataDrivenness
        % is between 0 and 1
    AdjustedSigmaHat = sqrt(SigmaHat^2 - SingleDotStdDev^2)/DDSqueezed;
else
    AdjustedSigmaHat = 0;
end
cX = mean(ItemLocsX);
cY = mean(ItemLocsY);
MeanDispersion = mean((sum([(ItemLocsX - ones(NStim,1)*cX);...
    (ItemLocsY - ones(NStim,1)*cY)].^2)/(2*NStim-2)).^(1/2));
    % Computes the mean value of paper
    % Eq. 2 across all trials in the
    % data set.

if AdjustedSigmaHat > MeanDispersion % then pmiss is undefined because it
    % is impossible to decimate stimuli

```

```

% to make response error as large as
% the residual error displayed by
% the subject.
    pmiss = NaN;
else
    PosFilter = AttentionFilter;
    PosFilter(PosFilter < 0)=0.001; % If Efficiency is low, then errors
    % estimating AttentionFilter can
    % produce large, spurious negative
    % values that lead (for reasons that
    % remain obscure) to systematic
    % overestimates of Efficiency. These
    % lines fix this problem.
    pmiss = GetPMiss(AdjustedSigmaHat,ItemCountsPerCloud,...
                    PosFilter,MeanDispersion);
end

%% Store analysis results to be returned
Results = struct;
% The first fields record the data upon which the analysis was performed
Results.Types = Types;
Results.TargetFilter = TargetFilter;
Results.ItemTypes = ItemTypes;
Results.ItemLocsX = ItemLocsX;
Results.ItemLocsY = ItemLocsY;
Results.ResponseX = ResponseX;
Results.ResponseY = ResponseY;
% The rest of the fields record the analysis results
Results.AttentionFilter = AttentionFilter;
Results.AttentionFilter95pcntCI = AttentionFilterCIs;
Results.DataDrivenness = DataDrivenness;
Results.DataDrivenness95pcntCI = DataDrivennessCIs;
Results.x_default = x_default;
Results.y_default = y_default;
Results.SigmaHat = SigmaHat;
Results.Efficiency = 1-pmiss;
if ~isempty(Mismatch)
    Results.FilterFidelity = 1-Mismatch;
else
    Results.FilterFidelity = [];
end
Results.SingleDotStdDev = SingleDotStdDev;
Results.Residual = Residual;
end

```

**GetAttentionFilterCIs.m (Called
by FitCentroidModel.m)**

```
function [mL,mU] = GetAttentionFilterCIs(WhichK,WHat,...
                                       CovMatFilterOnly,Crit,df,SigmaHat)

% GetAttentionFilterCIs uses a theorem of Fieller, 1954, that gives the
% confidence intervals for a ratio of normal random variables, provided
% the denominator is highly likely to be nonnegative. The reference is
% Fieller, EC. (1954). "Some problems in interval estimation.". Journal of
% the Royal Statistical Society, Series B 16 (2): 175-185.

NTypes = size(CovMatFilterOnly,1);
b = sum(WHat(1:NTypes));
a = WHat(WhichK);
bvec = ones(NTypes,1);
avec = zeros(NTypes,1); avec(WhichK) = 1;
M = [avec bvec];
CovMat = M'*CovMatFilterOnly*M;
nu11 = CovMat(1,1);
nu22 = CovMat(2,2);
nu12 = CovMat(1,2);
t = tinv((1+Crit)/2,df);
g = (nu22*(t*SigmaHat)^2)/(b^2); % Paper Eq. 50
R = a/b;
mL = (R - (g*nu12)/nu22 - ((SigmaHat*t)/b)*... % Paper Eq. 49
      sqrt(nu11 - 2*R*nu12+(R^2)*nu22-g*(nu11 - (nu12^2)/nu22)))/(1-g);
mU = (R - (g*nu12)/nu22 + ((SigmaHat*t)/b)*... % Paper Eq. 49
      sqrt(nu11 - 2*R*nu12+(R^2)*nu22-g*(nu11 - (nu12^2)/nu22)))/(1-g);
end
```

GetMismatch.m (Called by FitCentroidModel.m)

```

function Mismatch = GetMismatch(ItemCountsPerCloud, AttentionFilter, ...
                                TargetFilter)

% Mismatch = GetMismatch(ItemCountsPerCloud, AttentionFilter, TargetFilter)
%
% Mismatch = 1 - FilterFidelity (See paper Eq. 8).

NItemsPerCloud = sum(ItemCountsPerCloud);
ItemWeights = ItemCountsPerCloud(:).*TargetFilter(:);
LowestItemWeight = min(ItemWeights);
tmp = find(ItemWeights == LowestItemWeight);
fworstIndex = tmp(1);

VecWorst = zeros(1,NItemsPerCloud);
VecTarg = NaN(1,NItemsPerCloud);
VecAF = NaN(1,NItemsPerCloud);
for k=1:length(TargetFilter)
    if k==1
        st = 1;
        nd = ItemCountsPerCloud(1);
    else
        st = nd+1;
        nd = nd+ItemCountsPerCloud(k);
    end
    VecTarg(st:nd) = TargetFilter(k);
    VecAF(st:nd) = AttentionFilter(k);
    if k == fworstIndex
        VecWorst(st:nd) = 1;
    end
end
VecWorst = VecWorst/sum(VecWorst);
VecAF = VecAF/sum(VecAF);
VecTarg = VecTarg/sum(VecTarg);
Mismatch = norm(VecAF(:)-VecTarg(:))/norm(VecWorst(:)-VecTarg(:));
end

```

GetPMiss.m (Called by FitCentroidModel.m)

```
function pmiss = GetPMiss(SigmaHat,ItemCountsPerCloud,AttentionFilter,...
                        Dispersion)
% pmiss = GetPMiss(SigmaHat,ItemCountsPerCloud,AttentionFilter,Dispersion)
%
% Computes paper Eq. 69

CloudSize = sum(ItemCountsPerCloud);
varLower = 0;
DecimationNum = CloudSize;
for k=1:CloudSize-1
    crntVar = GetVarCorrespondingToDecimationNum(ItemCountsPerCloud,...
                                                AttentionFilter,Dispersion,k);
    if SigmaHat^2 < crntVar
        DecimationNum = k-1 + (SigmaHat^2 - varLower)/(crntVar-varLower);
        break
    else
        varLower = crntVar;
    end
end
pmiss = DecimationNum/CloudSize;
end
```


GetVarCorrespondingToDecimationNum.m (Called by GetPMiss.m)

```

function varCorrespondingToDecNum = GetVarCorrespondingToDecimationNum(...
    ItemCountsPerCloud,AttentionFilter,Dispersion,DecimationNum)

% varCorrespondingToDecNum = GetVarCorrespondingToDecimationNum(...
%     ItemCountsPerCloud,AttentionFilter,Dispersion,DecimationNum)

AttentionFilter = AttentionFilter(:);
ItemCountsPerCloud = ItemCountsPerCloud(:);
NTypes = length(ItemCountsPerCloud);
Types = 1:NTypes;
AttentionFilter = AttentionFilter/(ItemCountsPerCloud'*AttentionFilter);
TypeWeights = AttentionFilter(Types);
TypeWeightsSq = TypeWeights.^2;
AllKVecs = GetAllCountVectors(ItemCountsPerCloud,DecimationNum);
S_A = TypeWeights(:)'*AllKVecs; % Paper Eq. 82
KdotFSquared = TypeWeightsSq(:)'*AllKVecs;
NdotFSquared = TypeWeightsSq(:)'*ItemCountsPerCloud;
Gamma = (S_A.^(-2) - 2*S_A.^(-1)).*KdotFSquared + NdotFSquared;
% Paper Eq. 83
Probs = GetProbs(ItemCountsPerCloud,AllKVecs); % Implements paper Eq. 85
varCorrespondingToDecNum = (Dispersion^2)*(Probs(:)'*Gamma(:));
% Paper Eq. 79 making use of Eq. 84
end

%% GetProbs
function Probs = GetProbs(ItemCountsPerCloud,AllKVecs)

% Computes paper Eq. 85

CloudSize = sum(ItemCountsPerCloud);
KVecSums = sum(AllKVecs);
NKVecs = size(AllKVecs,2);
Probs = exp(sum(logchoose(ItemCountsPerCloud*ones(1,NKVecs),AllKVecs)) ...
    -logchoose(CloudSize*ones(1,NKVecs),KVecSums));
end

%% logchoose
function log_n_choose_k = logchoose(n,k)

% log_n_choose_k = logchoose(n,k)

if size(n) ~= size(k)
    error('n and k must have the same size')
end

log_n_choose_k = NaN(size(n));
z = (n ~= round(n) | k ~= round(k) | n < 0 | k < 0 | n < k);
log_n_choose_k(~z) = (gammaln(n(~z)+1)...
    - (gammaln(k(~z)+1)+gammaln(n(~z)-k(~z)+1)));
end

```

**GetAllCountVectors.m (Called
by GetVarCorrespondingToDecimationNum.m)**

```

function AllCountVecs = GetAllCountVectors(nvec,L)

% AllCountVecs = GetAllCountVectors(nvec,L)
%
% Input variables: nvec is a vector of nonnegative integers whose sum is
% N. L is a nonnegative integer less than or equal to N.
%
% Output variable: AllCountVecs is a matrix whose columns comprise the complete
% set of vectors K such that (1) each component of K is a nonnegative
% integer less than or equal to the corresponding component of nvec, and
% (2) the sum of the component values in K is N-L.

global tmpAllCountVecs cnt Ntypes

Ntypes=length(nvec);
N = sum(nvec);
if N < L
    tmpAllCountVecs = [];
    return
end
NMinusL = N - L;
tmpAllCountVecs = NaN(Ntypes,round(exp(logchoose(N,L))));
    % Allocate more columns than we can possibly need. We do this
    % because it's faster to assign to preallocated space; however, we
    % don't know how to compute exactly how many columns we'll need ahead
    % of time. We'll discard the ones we don't use at the end.

cnt = 0;

kvec = zeros(Ntypes,1);

bottom = max([0 NMinusL-sum(nvec(2:Ntypes))]);
top = min([nvec(1) NMinusL]);

for h=bottom:top
    kvec(1) = h;
    branch(nvec,kvec,NMinusL,2);
end
AllCountVecs = tmpAllCountVecs(:,1:cnt);
end

%% branch %%
function branch(nvec,kvec,NMinusL,j)
global tmpAllCountVecs cnt Ntypes

```

```

if sum(kvec) == NMinusL
    cnt = cnt+1;
    tmpAllCountVecs(:,cnt) = kvec(:);
elseif j == Ntypes
    tmp = NMinusL-sum(kvec);
    if tmp <= nvec(j)
        kvec(j) = tmp;
        cnt = cnt+1;
        tmpAllCountVecs(:,cnt) = kvec(:);
        return
    end
elseif j > Ntypes
    return
else % j < Ntypes
    NumLeft = NMinusL-sum(kvec);
    bottom = max([0 NumLeft - sum(nvec(j+1:Ntypes))]);
    top = min([nvec(j) NumLeft]);

    for h=bottom:top
        tmp = kvec;
        tmp(j) = h;
        branch(nvec,tmp,NMinusL,j+1);
    end
end
end
end

```

Appendix 5: Matlab code for nested model comparison

Main program: FTestForEqualityOfFilters.m

```

function [F,dfNumerator,dfDenominator,p] = ...
    FTestForEqualityOfFilters(Data1,Data2)

% [F,dfNumerator,dfDenominator,p] = FTestForEqualityOfFilters(Data1,Data2)
%
% This function performs an F-test of the null hypothesis that the
% participant used the same attention filter in the condition whose data
% is collected in Data1 as he/she did in the condition whose data
% is collected in Data2. Model parameters other than the attention filter
% are allowed to take separate values in fitting the data from the two
% different conditions.
%
% Data1 and Data2 are both structures comprising the following fields:
%
% ItemTypes
% ItemLocsX

```

```

% ItemLocsY
% ResponseX
% ResponseY
%
% ItemTypes is an NItemsPerCloud by NumTrials matrix. ItemTypes(i,k)
% gives the type of the ith item in the cloud presented on trial k.
%
% ItemLocsX is an NItemsPerCloud by NumINumTrials matrix. ItemLocsX(i,k)
% gives the x-coordinate of the location of the ith item in the cloud
% presented on trial k. ItemLocsY(i,k) gives the y-coordinate of the
% location of the ith item in the cloud presented on trial k. ResponseX(k)
% and ResponseY(k) give the x- and y-coordinates of the participant's
% response on trial k.
%
% It is assumed that unique(Data1.ItemTypes(:)) is equal to
% unique(Data2.ItemTypes(:)). If not, the program exist with an error
% message. However, the number of trials contributing to Data1 need not be
% the same as the number of trials contributing to Data2. Nor does it need
% to be the case that the number of items of a given type occurring in the
% stimulus clouds in Data1 is the same as the number of that type occurring
% in the stimulus clouds in Data2.

global NTypes X1 X2 Y1 Y2 ItemCountsPerCloud1 ItemCountsPerCloud2 ...
                                                RX1 RY1 RX2 RY2

IT1 = Data1.ItemTypes;
IT2 = Data2.ItemTypes;
Types = unique(IT1(:));
Types2 = unique(IT2(:));
if any(Types ~= Types2)
    error('Item types in Data1 must be the same as item types in Data2')
else
    clear Types2
end

XLocs1 = Data1.ItemLocsX;
XLocs2 = Data2.ItemLocsX;
YLocs1 = Data1.ItemLocsY;
YLocs2 = Data2.ItemLocsY;
RX1 = Data1.ResponseX;
RX2 = Data2.ResponseX;
RY1 = Data1.ResponseY;
RY2 = Data2.ResponseY;
NTrials1 = length(RX1);
NTrials2 = length(RX2);

NTypes = length(Types);
ItemCountsPerCloud1 = NaN(size(Types));
ItemCountsPerCloud2 = NaN(size(Types));

```

```

X1 = NaN(NTrials1,NTypes);
Y1 = NaN(NTrials1,NTypes);
X2 = NaN(NTrials2,NTypes);
Y2 = NaN(NTrials2,NTypes);
for k=1:NTypes
    ItemCountsPerCloud1(k) = sum(IT1(:,1)==Types(k));
    ItemCountsPerCloud2(k) = sum(IT2(:,1)==Types(k));
    X1(:,k) = sum(XLocs1.*(IT1==Types(k)))';
    Y1(:,k) = sum(YLocs1.*(IT1==Types(k)))';
    X2(:,k) = sum(XLocs2.*(IT2==Types(k)))';
    Y2(:,k) = sum(YLocs2.*(IT2==Types(k)))';
end

Results1 = FitCentroidModelForFTest(Data1);
Results2 = FitCentroidModelForFTest(Data2);
RSSFull = sum(Results1.Residual.^2) + sum(Results2.Residual.^2);
ParamsFull = 2*(NTypes + 2);    % Each of Data1 and Data2 has been fit by a
                                % whose parameters are an attention filter
                                % that absorbs NTypes-1 degrees of freedom,
                                % plus the 3 additional parameters,
                                % DataDrivenness, x_default and y_default.
dfFull = 2*(NTrials1+NTrials2) - ParamsFull;

ResultsRestricted = FitCentroidModelForFTestRestricted(Results1,Results1);
ParamsRestricted = NTypes + 5; % We force the model fits of Data1 and
                                % Data2 to use the same attention filter.
                                % However, all the other parameters are
                                % free to vary separately; thus, we need
                                % parameters DataDrivenness1, x_default1
                                % and y_default1 as well as
                                % DataDrivenness2, x_default2
                                % and y_default2 in addition to the shared
                                % attention filter.
dfRestricted = 2*(NTrials1+NTrials2)-ParamsRestricted;
RSSRestricted = ResultsRestricted.ResidualSumSquares;
dfNumerator = dfRestricted-dfFull;
Numerator = (RSSRestricted-RSSFull)/dfNumerator;
dfDenominator = dfFull;
Denominator = RSSFull/dfDenominator;
F = Numerator/Denominator;
p = 1 - Fcdf(F,dfNumerator,dfDenominator);
end

```

**FitCentroidModelForFTest.m (Called
by FTestForEqualityOfFilters.m)**

```

function Results = FitCentroidModelForFTest(DataStruct)
% Results = FitCentroidModelForFTest(DataStruct)
%
% This function performs only those computations required for the F-test of
% the null hypothesis that two data sets can be modeled using the same
% attention filter. Specifically, it (1) derives estimates of
% DataDrivenness, the AttentionFilter, x_default, y_default and (2)
% computes residual error. It does not compute confidence intervals for
% model statistics, nor does it compute Efficiency or FilterFidelity.

%% Set up for analysis
ItemTypes = DataStruct.ItemTypes;
ItemLocsX = DataStruct.ItemLocsX;
ItemLocsY = DataStruct.ItemLocsY;
ResponseX = DataStruct.ResponseX;
ResponseY = DataStruct.ResponseY;
NTrials = length(ResponseX);
Types = unique(ItemTypes(:));
NTypes = length(Types);
ItemCountsPerCloud = NaN(size(Types));
for k=1:NTypes
    ItemCountsPerCloud(k) = sum(ItemTypes(:,1)==Types(k));
end
R = [ResponseX(:);ResponseY(:)];           % Eq. 23
X = NaN(NTrials,NTypes);
Y = NaN(NTrials,NTypes);
for k=1:NTrials
    for h=1:NTypes
        X(k,h) = sum(ItemLocsX(ItemTypes(:,k)==Types(h),k));
        Y(k,h) = sum(ItemLocsY(ItemTypes(:,k)==Types(h),k));
    end
end
Zeros = zeros(NTrials,1);
Ones = ones(NTrials,1);
M = [[X Ones Zeros];[Y Zeros Ones]];      % Eq. 22

%% Estimate model parameters
[What,~,Residual] = regress(R,M);
Results = struct;
Results.DataDrivenness = What(1:NTypes)'*ItemCountsPerCloud(:); % Eq. 27
Results.AttentionFilter = What(1:NTypes)/sum(What(1:NTypes)); % Eq. 29
Results.x_default = What(NTypes+1)/(1-Results.DataDrivenness); % Eq. 30
Results.y_default = What(NTypes+2)/(1-Results.DataDrivenness); % Eq. 30
Results.Residual = Residual;
end

```

**FitCentroidModelForFTestRestricted.m (Called
by FTestForEqualityOfFilters.m)**

```
function Results = FitCentroidModelForFTestRestricted(Results1,Results2)

% Results = FitCentroidModelForFTestRestricted(Results1,Results2)
%
% Results1 and Results2 are structures derived by calls
%
% Results1 = FitCentroidModelForFTest(Data1);
% Results2 = FitCentroidModelForFTest(Data2);
%
% where Data1 and Data2 are structures with fields as described in the main
% comment to FTestForEqualityOfFilters.m. The current function fits a
% model to the pooled data collected in Data1 and Data2 that uses the same
% attention filter to all the data but allows separate DataDrivenness,
% x_default and y_default parameters for the two data sets.

global NTypes

FilterGuess = Results1.AttentionFilter + Results1.AttentionFilter;
FilterGuess = FilterGuess/sum(FilterGuess);
guess = [FilterGuess(:)' ...
         Results1.DataDrivenness Results2.DataDrivenness ...
         Results1.x_default Results2.x_default ...
         Results1.y_default Results2.y_default];
options = optimset('MaxFunEvals',10000,'MaxIter',10000,'TolX',1,'TolFun',0.0001);
[Params, SSDevs] = fminsearch('SharedAttentionFilterSumSquaredDevs',guess,options);
Results = struct;
Results.SharedAttentionFilter = Params(1:NTypes);
Results.DataDrivenness1 = Params(NTypes+1);
Results.DataDrivenness2 = Params(NTypes+2);
Results.x_default1 = Params(NTypes+3);
Results.x_default2 = Params(NTypes+4);
Results.y_default1 = Params(NTypes+5);
Results.y_default2 = Params(NTypes+6);
Results.ResidualSumSquares = SSDevs;
end
```

**SharedAttentionFilterSumSquaredDevs.m (Called
by FitCentroidModelForFTestRestricted.m)**

```

function SSDevs = SharedAttentionFilterSumSquaredDevs(guess)

% SSDevs = SharedAttentionFilterSumSquaredDevs(guess)

global NTypes X1 X2 Y1 Y2 ItemCountsPerCloud1 ItemCountsPerCloud2 ...
                                                RX1 RY1 RX2 RY2

SharedF = guess(1:NTypes)';
DD1 = guess(NTypes+1);
DD2 = guess(NTypes+2);
x_d1 = guess(NTypes+3);
x_d2 = guess(NTypes+4);
y_d1 = guess(NTypes+5);
y_d2 = guess(NTypes+6);

S1 = ItemCountsPerCloud1(:)'*SharedF(:);
S2 = ItemCountsPerCloud2(:)'*SharedF(:);

PredictedRX1 = (DD1/S1)*X1*SharedF + (1-DD1)*x_d1;
PredictedRY1 = (DD1/S1)*Y1*SharedF + (1-DD1)*y_d1;
PredictedRX2 = (DD2/S2)*X2*SharedF + (1-DD2)*x_d2;
PredictedRY2 = (DD2/S2)*Y2*SharedF + (1-DD2)*y_d2;

SSDevs = sum((PredictedRX1-RX1).^2) + sum((PredictedRY1-RY1).^2)+...
        sum((PredictedRX2-RX2).^2) + sum((PredictedRY2-RY2).^2);
end

```


References

- Alvarez, G. A. (2011). Representing multiple objects as an ensemble enhances visual cognition. *Trends in Cognitive Sciences*, 15(3), 122–131.
- Alvarez, G. A., & Oliva, A. (2008). The representation of simple ensemble visual features outside the focus of attention. *Psychological Science*, 19(4), 392–398.
- Ariely, D. (2001). Seeing sets: Representation by statistical properties. *Psychological Science*, 12(2), 157–162.
- Arman, A. C., Ciaramitaro, V. M., & Boynton, G. M. (2006). Effects of feature-based attention on the motion aftereffect at remote locations. *Vision Research*, 46(18), 2968–2976.
- Baldassi, S., & Vergheze, P. (2005). Attention to locations and features: Different topdown modulation of detector weights. *Journal of Vision*, 5(6), 556–570.
- Ball, K., & Sekuler, R. (1981). Adaptive processing of visual motion. *Journal of Experimental Psychology: Human Perception and Performance*, 7(4), 780–794.
- Blair, G., Wright, C. E., Chubb, C., & Sperling, G. (2015). Disc size supports top-down, selective attention in a task requiring integration across multiple targets. *Journal of Vision*, 15(12), 897–897.
- Chong, S. C., & Treisman, A. M. (2003). Representation of statistical properties. *Vision Research*, 43(4), 393–404.
- Chong, S. C., & Treisman, A. M. (2005a). Attentional spread in the statistical processing of visual displays. *Perception and Psychophysics*, 67(1), 1–13.
- Chong, S. C., & Treisman, A. M. (2005b). Statistical processing: Computing the average size in perceptual groups. *Vision Research*, 45(7), 891–900.
- Chubb, C., & Nam, J.-H. (2000). The variance of high contrast texture is sensed using negative half-wave rectification. *Vision Research*, 40, 1695–1709.
- Chubb, C., & Talevich, J. (2002). Attentional control of texture orientation judgments. *Vision Research*, 42, 311–330.
- Dakin, S. C. (2001). Information limit on the spatial integration of local orientation signals. *Journal of the Optical Society of America A*, 18, 1016–1026.
- Davis, E. T., & Graham, N. (1981). Spatial frequency uncertainty effects in the detection of sinusoidal gratings. *Vision Research*, 21(5), 705–712.
- Drew, S., Chubb, C., & Sperling, G. (2010). Precise attention filters for Weber contrast derived from centroid estimations. *Journal of Vision*, 10(10:20), 1–16. <http://www.journalofvision.org/content/10/10/20>
- Felisberti, F. M., & Zanker, J. M. (2005). Attention modulates perception of transparent motion. *Vision Research*, 45(19), 2587–2599.
- Fieller, E. (1954). Some problems in interval estimation. *Journal of the Royal Statistical Society, Series B*, 16(2), 175–185.
- Graybill, F. (1961). *An introduction to linear statistical models, Volume 1*: McGraw-Hill.
- Haenny, P. E., Maunsell, J. H., & Schiller, P. H. (1988). State-dependent activity in monkey visual cortex. ii. Retinal and extraretinal factors in v4. *Experimental Brain Research*, 69(2), 245–259.
- Hasher, L., & Zacks, R. T. (1988). Working memory, comprehension, and aging: A review and a new view. In G. H. Bower (Ed.), *Psychology of Learning and Motivation* (Vol. 22 pp. 193–225). New York: Academic Press.
- Hayden, B. Y., & Gallant, J. L. (2005). Time course of attention reveals different mechanisms for spatial and feature-based attention in area v4. *Neuron*, 47(5), 637–643.
- Ho, T. C., Brown, S., Abuyo, N. A., Ku, E.-H. J., & Serences, J. T. (2012). Perceptual consequences of feature-based attentional enhancement and suppression. *Journal of Vision*, 12, 1–17. (8:15 (<http://www.journalofvision.org/content/12/8/15>), doi:10.1167/12.8.15))
- Inverso, M., Sun, P., Chubb, C., Wright, C. E., & Sperling, G. (In press). Evidence against global attention filters selective for absolute bar-orientation in human vision. *Attention, Perception and Psychophysics*.
- Kamitani, Y., & Tong, F. (2005). Decoding the visual and subjective contents of the human brain. *Nature and Neuroscience*, 8(5), 679–685.
- Lankheet, M. J., & Verstraten, F. A. (1995). Attentional modulation of adaptation to two-component transparent motion. *Vision Research*, 35(10), 1401–1412.
- Ling, S., Liu, T., & Carrasco, M. (2009). How spatial and feature-based attention affect the gain and tuning of population responses. *Vision Research*, 49(10), 1194–1204.
- Liu, T., Larsson, J., & Carrasco, M. (2007). Feature-based attention modulates orientation-selective responses in human visual cortex. *Neuron*, 55(2), 313–323.
- Liu, T., & Mance, I. (2011). Constant spread of feature-based attention across the visual field. *Vision Research*, 51(1), 26–33.
- Liu, T., Slotnick, S. D., Serences, J. T., & Yantis, S. (2003). Cortical mechanisms of feature-based attentional control. *Cerebral Cortex*, 13(12), 1334–1343.
- Lu, Z.-L., & Doshier, B. A. (1998). External noise distinguishes attention mechanisms. *Vision Research*, 38, 1183–1198.
- Lustig, C., Hasher, L., & Zacks, R. T. (2007). Inhibitory deficit theory: Recent developments in a “new view.” In D. S. Gorfein, & C. M. MacLeod (Eds.), *The place of inhibition in cognition* (pp. 145–162). Washington: American Psychological Association.
- Martinez-Trujillo, J. C., & Treue, S. (2004). Feature-based attention increases the selectivity of population responses in primate visual cortex. *Current Biology*, 14(9), 744–751.
- Maunsell, J. H., Sclar, G., Nealey, T. A., & DePriest, D. D. (1991). Extraretinal representations in area v4 in the macaque monkey. *Visual Neuroscience*, 7(6), 561–573.
- McAdams, C. J., & Maunsell, J. H. (2000). Attention to both space and feature modulates neuronal responses in macaque area v4. *Journal of Neurophysiology*, 83(3), 1751–1755.
- Motter, B. C. (1994). Neural correlates of attentive selection for color or luminance in extrastriate area v4. *The Journal of Neuroscience*, 14(4), 2178–2189.
- Muller, M. M., Anderson, S., Trujillo, N. J., Valdes-Sosa, P., & Hillyard, S. A. (2006). Feature-selective attention enhances color signals in early visual areas of the human brain. *Proceedings of the National Academy of Sciences, USA*, 103(38), 14250–14254.
- Nam, J.-H., & Chubb, C. (2000). Texture luminance judgments are approximately veridical. *Vision Research*, 40, 1677–1694.
- O’Craven, K. M., Rosen, B. R., Kwong, K. K., Treisman, A. M., & Savoy, R. L. (1997). Voluntary attention modulates fMRI activity in human MT-MST. *Neuron*, 18(4), 591–598.
- Quigley, C., Andersen, S. K., Schultz, L., Grunwald, M., & Muller, M. M. (2010). Feature-selective attention: Evidence for a decline in old age. *Neuroscience Letters*, 474(1), 5–8.
- Quigley, C., & Muller, M. M. (2014). Feature-selective attention in healthy old age: A selective decline in selective attention? *The Journal of Neuroscience*, 34(7), 2471–2476.
- Rossi, A. F., & Paradiso, M. A. (1995). Feature-specific effects of selective visual attention. *Vision Research*, 35(5), 621–634.
- Saenz, M., Buracas, G. T., & Boynton, G. M. (2002). Global effects of feature-based attention in human visual cortex. *Nature Neuroscience*, 5(7), 631–632.
- Saenz, M., Buracas, G. T., & Boynton, G. M. (2003). Global feature-based attention for motion and color. *Vision Research*, 43(6), 629–637.

- Schoenfeld, M. A., Hopf, J. M., Martinez, A., Mai, H. M., Sattler, C., Gasde, A., Heinze, H. J., & Hillyard, S. A. (2007). Spatio-temporal analysis of feature-based attention. *Cerebral Cortex*, *17*(10), 2468–2477.
- Seidemann, E., & Newsome, W. T. (1999). Effect of spatial attention on the responses of area MT neurons. *Journal of Neurophysiology*, *81*(4), 1783–1794.
- Serences, J. T., & Boynton, G. M. (2007). Feature-based attentional modulations in the absence of direct visual stimulation. *Neuron*, *55*(2), 301–312.
- Serences, J. T., Saproo, S., Scolari, M., Ho, T., & Muftuler, L. T. (2009). Estimating the influence of attention on population codes in human visual cortex using voxel-based tuning functions. *Neuroimage*, *44*(1), 223–231.
- Shih, S. I., & Sperling, G. (1996). Is there feature-based attentional selection in visual search? *Journal of Experimental Psychology: Human Perception and Performance*, *22*(3), 758–779.
- Solomon, J. A. (2010). Visual discrimination of orientation statistics in crowded and uncrowded arrays. *Journal of Vision*, *10*, 1–16. (14:19 <http://www.journalofvision.org/content/10/14/19>, doi:10.1167/10.14.19)
- Sperling, G. (1963). A model for visual memory tasks. *Human Factors*, *5*, 19–31.
- Sperling, G., Wurst, S. A., & Lu, Z.-L. (1992). Using repetition detection to define and localize the processes of selective attention. In D. E. Meyer, & S. Kornblum (Eds.), *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience—A silver jubilee, chapter 12* (pp. 265–298). Cambridge: MIT Press.
- Treue, S. (2001). Neural correlates of attention in primate visual cortex. *Trends in Neuroscience*, *24*(5), 295–300.
- Treue, S., & Martinez-Trujillo, J. C. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, *399*(6736), 575–579.
- White, A. L., & Carrasco, M. (2011). Feature-based attention involuntarily and simultaneously improves visual performance across locations. *Journal of Vision*, *11*, 1–10. (6:15(<http://www.journalofvision.org/content/11/6/15>), doi:10.1167/11.6.15))
- Zhang, W., & Luck, S. J. (2009). Feature-based attention modulates feedforward visual processing. *Nature Neuroscience*, *12*(1), 24–25.