Utrecht University

BACHELOR THESIS

A Model for Numerosity Adaptation

15 ECTS

Author:
Virginia Meijer
5488745

Supervisors:
Andromachi Tsouli MSc
dr. Maarten van der Smagt

Bachelor Artificial Intelligence
Faculty of Humanities

July 13, 2018

Abstract

Virginia Meijer

A Model for Numerosity Adaptation

Research into numerosity perception has shown that the set size of a group of items can be perceived by human species as well as non-human species through a similar numerosity system. Further and more recent research has shown that this numerosity system is susceptible to adaptation. However, the question which remains unanswered is "how does numerosity adaptation occur".

This paper specifically investigates numerosity adaptation on a neural level. It proposes a model for numerosity adaptation similar to adaptation in the primary visual cortex (V1), but modified to the neural characteristics of numerosity neurons. Using a MATLAB implementation of the model, results for the numerosity adaptation model are computed through simulations of the implemented model. These results are then compared to earlier found psychophysical results of numerosity adaptation. Especially the results of two earlier studies (Aagten-Murphy & Burr, 2016; Tsouli et al., 2018) are compared to the model's results.

This paper found that earlier results could be explained fully by the modified V1 adaptation model, providing a working model for numerosity adaptation. Even so, more research should be done to ascertain if the model provided fully explains numerosity adaptation or only the part of it examined in this paper.

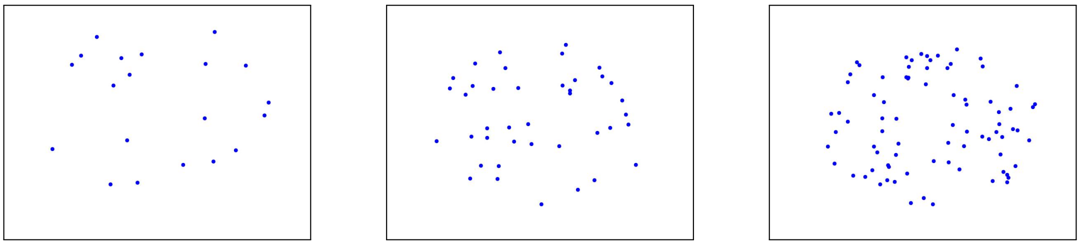
Keywords: numerosity perception, adaptation, modeling, tuning curves

Contents

1 Introduction & Theoretical background

1.1 Numerosity Perception

Numerosity refers to the set size of a group of items as shown in figure 1.1. Numerosity can be perceived by humans species (Xu & Spelke, 2000; Barth et al., 2005; Cantlon & Brannon, 2007), as well as by non-human species through a similar system (Agrillo et al., 2012; Jones et al., 2013). These findings show that numerosity can be defined as symbolic (numbers) as well as non-symbolic (for example dots).



(a) **Numerosity 20 dots** (b) **Numerosity 40 dots** (c) **Numerosity 80 dots**

FIGURE 1.1: THREE EXAMPLES OF NUMEROSITIES
Numerosities 20, 40 and 80 are respectively shown in the form of 20, 40 and 80 dots.

One theory for the way numerosity is perceived defines numerosity perception as a *number sense*. According to the *number sense* theory numerosity perception mirrors primary sensory perception and thus should behave as a primary perceptual attribute for which a dedicated perceptual system should exist (Anobile et al., 2016). Evidence for the *number sense* theory has been found in the human parietal cortex, in which neurons specifically tuned to the perception of certain numerosities along with a network of topographic maps for numerosity perception has been found (Harvey et al., 2013; Harvey & Dumoulin, 2017). Further evidence has been found through the discovery of a topographic map for object size perception, which did not fully coincide with the topographic numerosity map (Harvey et al., 2015). This indicates that a dedicated system for numerosity perception exists.

Moreover, a trait which signifies a dedicated perceptual system exists is the ability to adapt and can be found in perceptual systems such as vision (for example research by Thompson and Burr (2009) and King and Crowder (2018)) and audition (for example research by Parra and Pearlmutter (2007) and Schweinberger et al. (2008)). Following the theory that numerosity behaves as a perceptual system, adaptation should occur for this system as well. Indeed, earlier research shows that numerosity is susceptible to adaptation (Burr & Ross, 2008; Arrighi et al., 2014; Anobile et al., 2016; Castaldi et al., 2016; Tsouli et al., 2018).

Even though previous research has ascertained the attribute of adaptation for numerosity perception, the question that remains unanswered is "how does numerosity adaptation occur". That is to say, does a model exist which can explain the adaptive behavior of numerosity?

1.2 Neural Adaptation

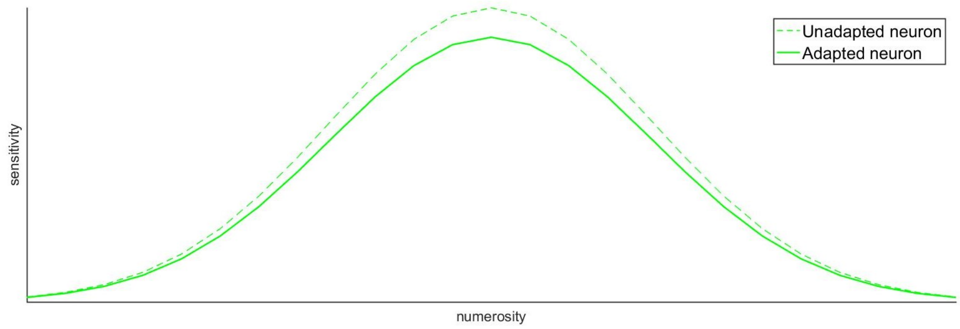
Neural adaptation refers to the recalibration on a neuron's sensitivity after perceiving a stimulus to optimize perception in a particular system (Thompson & Burr, 2009; Mease et al., 2013). The individual neurons adapt which in turns changes the neural population's reaction to a stimulus. In other words, the change in the neurons' sensitivities results in a change in perception of the stimulus before and after adaptation. Adaptation can occur at any stage of a perceptual system and, depending on its place in the hierarchy of the system, adaptation behaves in a different manner (Kohn, 2007).

An example can be found in the visual system. Adaptation can occur in a lower-order cortex such as the primary visual cortex (V1), where adaptation is strongest for stimuli similar to the adapter. However, adaptation in a higher-order cortex such as the middle temporal area (MT) adaptation for stimuli similar to the adapter is weakest rather than strongest (Kohn & Movshon, 2004; Krekelberg et al., 2006).

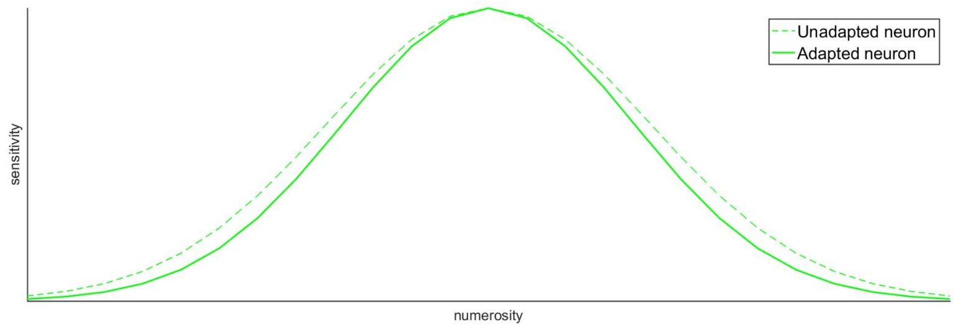
There are three tuning curve properties which adaptation can influence. The first property is the amplitude of the individual tuning curves, specifically the decrease of the amplitude as a result of adaptation as shown in figure 1.2(a). A decrease in amplitude results in a lesser excitation response of the neuron when presented a stimulus similar to the adapter after adaptation (Jin et al., 2005; Kohn, 2007). This results in a shift in perception of the stimulus after adaptation as seen in figure 1.3(a).

The second property is the width of the individual tuning curves, which may narrow as result of adaptation and shown in figure 1.2(b). Narrowing of an individual tuning curve results in the neuron gaining specificity around the adapter. The excitation response of the neuron should not be reduced much for a stimulus close to the numerosity preference, but it should differ greatly if it is further from the numerosity preference (Grill-Spector et al., 2006; Kohn, 2007). The reduced individual response cause a shift of the population excitation curve as seen in figure 1.3(b) which ensures a shift in perception of the stimulus after adaptation.

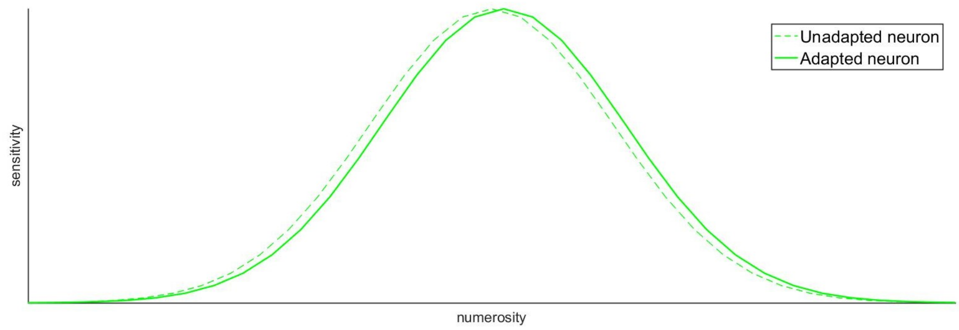
The final property is the shifting of the individual tuning curves either towards or away from the adapter, respectively an attractive or repulsive shift. Figure 1.2(c) shows a repulsive shift. The shifting of an individual tuning curve results in a change of numerosity preference for that particular neuron (Kohn & Movshon, 2004; Jin et al., 2005; Clifford et al., 2007; Quiroga et al., 2016). A repulsive shift results in a greater excitation response of neurons further away of the adapter and a reduced excitation response of neurons closer to the adapter. An attractive shift results in an opposite response, a greater response for stimuli similar to the adapter and a reduced response for stimuli different from the adapter. Shifting of the individual curves causes an overall shift of the population excitation curve as seen in figure 1.3(c) and thus a shift in perception of the stimulus after adaptation.



(a) **Amplitude reduction for individual tuning curves**

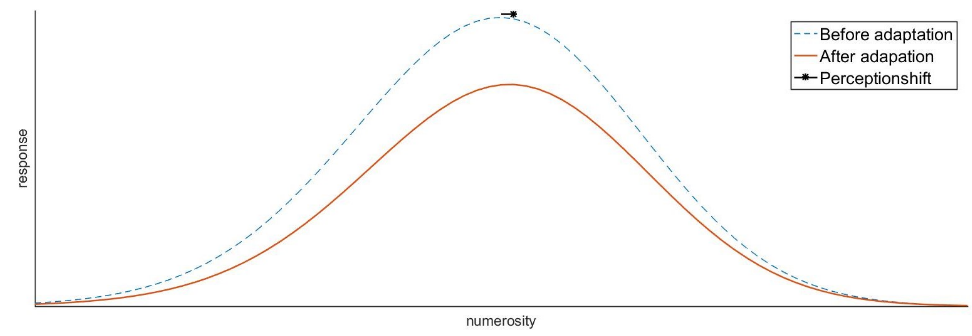


(b) **Width narrowing for individual tuning curves**

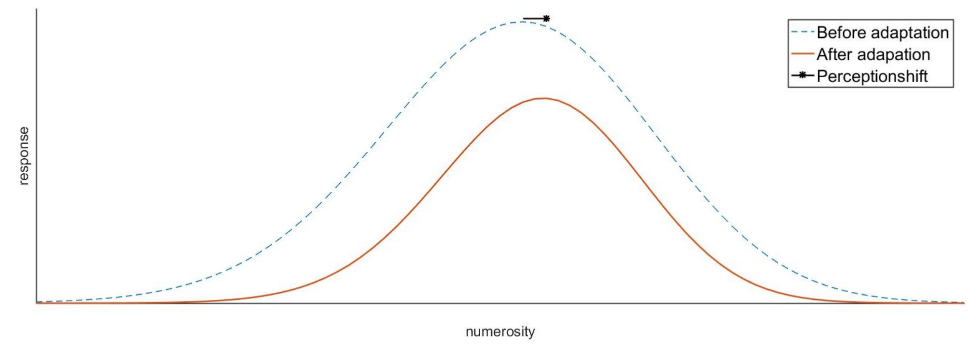


(c) **Shifting for individual tuning curves**

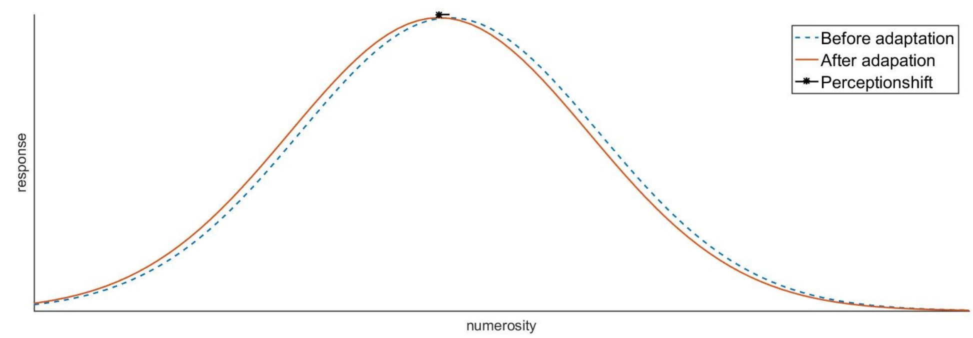
FIGURE 1.2: EXAMPLES OF ADAPTATION PROPERTIES FOR ONE INDIVIDUAL CURVE
Figures (a), (b) and (c) show the effect of a decrease in amplitude, narrowing of the width and shifting away from the adapter for an individual tuning curve with an adaptation strength of 10%, respectively. Consider an arbitrary adapter α and an arbitrary stimulus β. For figures (a) and (b) the shift shown resulted when $\alpha < \beta$. For figure (c) the shift shown is the result when $\alpha > \beta$. The dotted lines represent the unadapted curves and the solid lines represent the 10% adapted curves.



(a) **The effect of amplitude reduction on the population excitation response**



(b) **The effect of width narrowing on the population excitation response**



(c) **The effect of shifting on the population excitation response**

FIGURE 1.3: EXAMPLES OF PERCEPTIONS SHIFTS FOR ADAPTATION PROPERTIES
Figures (a), (b) and (c) show the effect of the adapted individual responses on the perception through the shift in perception for reduction of the amplitude, narrowing of the tuning width and shifting of the curve, respectively. Consider an arbitrary adapter α and an arbitrary stimulus β. For figures (a) and (b) the shift shown resulted for the condition $\alpha < \beta$. For figure (c) the shift shown is a result of the condition $\alpha > \beta$. For all figures a population of 100 neurons is used. The blue dotted lines represent the excitation curves for the unadapted population. The red lines represent adapted excitation curves. The shift in perception is represent by the solid black line. If the shift in perception line leans to the right the shift is attractive indicating overestimation. If the shift in perception line leans left the shift is repulsive and for perception indicating underestimation of the second stimulus.

1.3 Numerosity Adaptation

Considering there is no apparent model for numerosity adaptation yet, earlier found models for neural adaptation of other primary senses and perceptual attributes should be consulted as to how numerosity adaptation may work. The simplest known model which can then be considered for numerosity adaptation is a model in which only the first property, a decrease in amplitude of the tuning width, is influenced by adaptation. Such models are often found for the adaptation of visual attributes in V1 such as orientation and spatial frequency (Wilson & Humanski, 1993; Jin et al., 2005; Kohn, 2007; Thompson & Burr, 2009; King & Crowder, 2018). The individual tuning curves for these attributes follow a Gaussian fit rather than the log Gaussian fit that was found for numerosity neurons by Harvey et al. (2013). However, the principle of adaptation remains the same as Heron et al. (2011) have shown that log Gaussian fit neurons can also be reduced in amplitude while retaining their shape.

The adapted population excitation curve that follows from the individually adapted tuning curves would shift compared to the unadapted population excitation curve. The difference between the peak of the unadapted and adapted population excitation curve represent the underestimation or overestimation of the perception of the stimulus after adaptation.

Earlier research by Tsouli et al. (2018) has found on average a symmetrical shift in perception of about 5 dots of the stimulus numerosity 40 shown after adaptation to numerosities 20 and 80. Research done by Aagten-Murphy and Burr (2016) found a symmetrical shift in perception of about 25% to 30% underestimation and overestimation of the stimulus numerosity 40 to adapter numerosities 20 and 80. In comparison to Tsouli et al. (2018) the absolute number of dots underestimated or overestimated for stimulus numerosity 40 to adapter numerosities 20 and 80 found by Aagten-Murphy and Burr (2016) is asymmetrical. Research done by Castaldi et al. (2016) does not provide a shift in perceived number of dots after adaptation, but they found that when adapting to a higher numerosity of 80 the stimulus numerosity 40 is underestimated when it is perceived after adaptation. Such an overall underestimation of stimulus numerosity 40 is also seen in the research of Tsouli et al. (2018) and of Aagten-Murphy and Burr (2016). According to both psychophysical and fMRI research, the model should return an overall underestimation when adapting to numerosity 80 with a stimulus numerosity 40.

In order to discover the relevance of the model as an explanation for the behavior of numerosity adaptation, the question which needs to be answered is how well this model can explain earlier found results. Considering the model has to account for both a symmetrical and asymmetrical shift in perception found for stimulus numerosity 40 and adapter numerosities 20 and 80, the question arises which parameters may cause the different shifts in perception found. Aagten-Murphy and Burr (2016) used two different adaptation strengths in their research (1s and 5s adaptation), which indicates that adaptation strength influences the shift in perception of the stimulus.

Another parameter which may influence the shift in perception of the stimulus is the tuning width. The width of the tuning curves influences the overlap between the tuning curves; the wider the tuning curves the more overlap between the tuning curves and the more neurons contribute to the overall perception of the stimulus. The more neurons contribute to the overall perception, the stronger the adaptation effect will be. The exception to this is when the tuning curves are so wide that there is too much overlap. If the tuning curves overlap too much most of the population will be adapted rather than the range surrounding the adapter.

For example when adapting to numerosity 20 to test the perception of a stimulus numerosity 17. Since the overall perception of a stimulus consists of all neurons which fire when presented the stimulus, the perception of numerosity 17 is based upon the excitations of the neurons for numerosities 16, 17 and 18. The surrounding area of about numerosities 15 to 25 should adapt. If the tuning width is too narrow only the range of about 19 to 21 may adapt, which would not effect the perception of the stimulus numerosity 17. However, if the tuning curves are a little wider the range of about 15 to 25 would adapt, which would indeed effect the perception of numerosity 17. Even so, if the tuning curves are too wide the range of about 14 to 26 may adapt, causing most of the tuning curves to adapt. Because most of the tuning curves have been adapted the overall perception for numerosity 17 will not differ before and after adaptation.

Yet it is difficult to work with tuning width since the tuning width of human tuning curves is not known. Because of this lack of knowledge multiple combinations of tuning width and adaptation strength might produce the same shift in perception.

So the research question to be examined is which combinations of tuning width and adaptation strength parameters produce the earlier symmetrical shift in perception found by Tsouli et al. (2018), the asymmetrical shift in perception found by Aagten-Murphy and Burr (2016), and the overall underestimation of stimulus numerosity 40 after adaptation to numerosity 80 found by Castaldi et al. (2016).

1.4 A.I. Context

Research into numerosity adaptation is a relatively new subject. Though numerosity perception has been researched before, the specific mechanisms of numerosity adaptation remain unclear. Research on numerosity adaptation mostly employ psychophysical methods and there is currently only one study examining numerosity adaptation using fMRI (Castaldi et al., 2016).

There have been very few previous attempts to formalize numerosity adaptation, which makes the research into numerosity adaptation especially relevant for the field of artificial intelligence (A.I.). As mentioned in the previous section, an apparent model for numerosity adaptation does not yet exist. The relevance of modeling adaptive behavior is that with a model predictions can be made about numerosity adaptation. These predictions can then be tested to reveal more about the working of our numerosity perception system. Moreover, implementing a computational model for numerosity adaptation provides a way to rapidly make predictions based upon different parameters. Finally, a computational model of numerosity perception and adaptation might pave the way for a A.I. system in which numerical estimation is optimized. Such as a system can be used in real life applications such as maintaining crowd control limits.

2 Method & Materials

In order to find the optimum combinations of tuning width and adaptation strength that produce the earlier found result, the method of the model needs to be specified and simulations with the model must be done. As discussed earlier, the model used here is a simple adaptation model in which only the amplitude of the individual tuning curves is reduced. A mathematical specification of the model provides a theoretical description of the model. However, to run simulations and gain results with the model an implemented model is necessary.

The following sections of this chapter describe the underlying mathematical model of V1 adaptation. The last section describes the method used to simulate the research question with the model, for which the implementation was done with MATLAB version R2016b[1].

An adaptation model with a Gaussian fit for the tuning curves was implemented as well as a model with a log Gaussian fit using MATLAB (for the implementation see appendices A.1 and A.2). The Gaussian fit tuning curves were used as a baseline since earlier research shows that the tuning curves in the V1 mostly follow Gaussian tuning (Wilson & Humanski, 1993; Westrick et al., 2016). The Gaussian fit model can therefore be used as a validity check for the implementation of the adaptation model. The log Gaussian fit curves are used for the numerosity tuning curves. The log Gaussian fit model can then be used to find the optimum sets of parameters.

The implementation of the model is for one specific set of parameters. For an implementation of the model with multiple sets of parameters see appendix A.3. If multiple sets of parameters are used the model keeps track of which set of parameters produces which output. This is useful for comparing different comparison between different sets of parameters and their outcomes.

[1]In the implementation a function *COG* is used to calculate the center of gravity, a measurement which relates to the middle point of the area under a curve. This function can be found via https://nl.mathworks.com/matlabcentral/fileexchange/48451-find-the-center-of-gravity-of-an-array?s_tid=prof_contriblnk.

2.1 Mathematical Model Overview

The model can be divided into a three-step process:

1. The model generates a set number of unadapted individual tuning curves and an unadapted population tuning curve based upon either a Gaussian or log Gaussian fit.
2. After creating the tuning curves the model takes these same tuning curves and a specified adapter to which it adapts the tuning curves.
3. The unadapted and adapted population excitation curves are calculated for a specified stimulus. The curves are then compared in order to calculate the peak shift in perception of the stimulus.

The outcome of a simulation of the model is a shift in perceived number of dots (i.e. the peak of the excitation curve) for the specified model parameters. If the shift is positive, the stimulus should be overestimated after adaptation. For example, a stimulus of 20 dots may be perceived as 20 dots before adaptation but should be perceived as more than 20 dots after adaptation. If the shift is negative, the stimulus should be underestimated after adaptation.

2.2 Model Parameters

The parameters shown in table 2.1 are considered for the model.[2] With the exception of the parameters *numerosity range* and *adaptation strength* the parameters should be given as natural numbers. As such, the parameters can be set to a value $n \in \mathbb{N}$. However, considering the literature, the limit of the human system for numerosity estimation is 100. For higher numerosities estimation probably depends on a texture density system (Anobile et al., 2016). This should be taken into consideration when selecting parameters.

TABLE 2.1: THE PARAMETERS OF THE MODEL
In the left column the parameter names are given. The middle column shows explanations of the parameters alongside an example value in the right column.

Parameter	Info	Example value
numerosity range	The range of numerosities for which the individual numerosity tuning curves are shown	$0 - 200$
numerosity range step size	The step size of the numerosity range. Increasing the step size changes the resolution of the generated curves	0.1
maximum peak numerosity	The maximum numerosity of the peak for which the individual tuning curves are generated	100
σ	The standard deviation with which the width of the individual tuning curves is determined	0.5
number of neurons	The number of neurons the model should generate individual tuning curves for	20
adaptation strength	The adaptation strength strength for the individual tuning curves tuned to the adapter numerosity	25%
stimulus	The numerosity for which the excitation curves should be calculated	40
adapter	The numerosity to which the individual tuning curves should be adapted	20

[2]In the MATLAB implementation one parameter is used for the *numerosity range* and the *numerosity range step size*. When specifying the parameters for a simulation of the model, the documentation of the model should be consulted on the format of the combined parameter.

2.3 Generating the Tuning Curves

Considering the parameters given as example values in table 2.1, the resulting tuning curves for a Gaussian and a log Gaussian fit are shown in respectively figures 2.1 and 2.2. In the figures two types of spaces are considered, the linear and logarithmic space. The difference can be seen along the x-axis, which increments either linearly or logarithmic. Because for the generation of the Gaussian fit tuning curves no logarithmic equations are used, the Gaussian fit tuning curves are only shown in linear space. However, for the generation of the log Gaussian fit tuning curves logarithmic equations are used (see equation 2.4) which is why the logarithmic space should be considered. The linear space can also be considered for the log Gaussian fit tuning curves in order to compare the Gaussian and log Gaussian fit tuning curves. Figures 2.1(a) and 2.2(a), and figures 2.1(b) and 2.2(c) show that the Gaussian and log Gaussian tuning curves are generated using the same method. However, figure 2.2(b) shows that even though the method is the same, the logarithmic conversion for the log Gaussian fit changes the individual tuning curves in linear space.[3]

The individual tuning curves are equally distributed across the numerosity range for the Gaussian fit model, whereas for the log Gaussian fit model, the tuning curves are distributed unequally to reflect the decrease in dedicated cortical space the higher the numerosity found by Harvey et al. (2013). The standard deviation σ has a different effect on the tuning widths for the Gaussian and log Gaussian fit neurons. For the Gaussian fit neurons the tuning curves all have an equal width as is often found in area V1 (Wilson & Humanski, 1993; Westrick et al., 2016). For the log Gaussian tuning curves the width of the tuning curves is equal in logarithmic space. When considering the curves in linear space the tuning width grows for each subsequent neuron. As a result, the log Gaussian tuning curves follow the pattern found in earlier research (Harvey et al., 2013). The subsequent growth of the tuning width allows more narrow tuning widths for lower numerosities and broader tuning widths for higher numerosities. In other words, the higher the numerosity the less specific the tuning curves.

The mean changes for every subsequent neuron. The numerosity preference of each neuron shifts when the mean increases. Because the mean increases per neuron, the preference of the neuron increases which creates a distribution of the tuning curves along the *numerosity range*. This change is seen in both the Gaussian and the log Gaussian tuning curves. The mean is calculated per Gaussian fit neuron:

$$\mu_i = \frac{maximum\ peak\ numerosity}{number\ of\ neurons + 1} \times i_{neuron} \tag{2.1}$$

For the log Gaussian fit the mean is calculated using the natural logarithm:

$$\mu_i = \frac{\ln(maximum\ peak\ numerosity)}{maximum\ neuron \in number\ of\ neurons} \times i_{neuron} \tag{2.2}$$

Where μ_i is the mean of the i^{th} neuron in *number of neurons*. i_{neuron} is the i^{th} neuron in *number of neurons*. $maximum\ neuron \in neurons$ is the highest value in *number of neurons*.

[3] Note that for a comparison between the two fits the σ used for the Gaussian fit should be the exponent of the σ used for the log Gaussian fit to account for the logarithmic conversion of the σ.

Using the *numerosity range*, σ and mean, the tuning curve for a particular neuron is generated by calculating the sensitivity values for every x-value in the *numerosity range* for that neuron. This is done for every neuron in *number of neurons* so that for each neuron a tuning curve is generated. The calculation is done through either the normal (shown in equation 2.3) or log normal (shown in equation 2.4) variant of the probability density function (PDF):

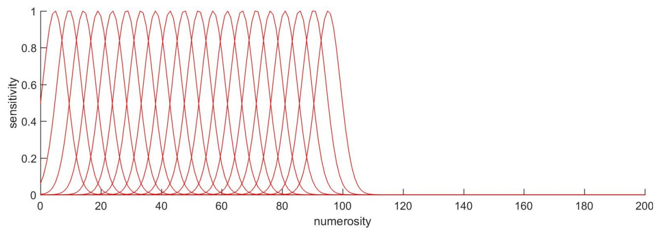
Gaussian fit

$$\begin{gathered} unadapted\ sensitivity_i = PDF(numerosity\_range, \mu_i, \sigma) \\ \text{In other words, for every } x \text{ in the } numerosity\ range \\ unadapted\ sensitivity_i^x = \frac{1}{\sigma \times \sqrt{2\pi}} \times \mathrm{e}^{\dfrac{-(x-\mu_i)^2}{2 \times \sigma^2}} \end{gathered} \tag{2.3}$$

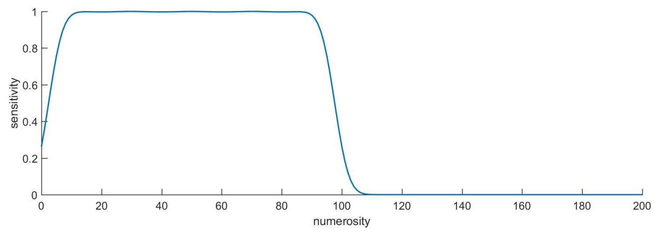
Log Gaussian fit

$$\begin{gathered} unadapted\ sensitivity_i = PDF(numerosity\_range, \mu_i, \sigma) \\ \text{In other words, for every } x \text{ in the } numerosity\ range \\ unadapted\ sensitivity_i^x = \frac{1}{x \times \sigma \times \sqrt{2\pi}} \times \mathrm{e}^{\dfrac{-(\ln x-\mu_i)^2}{2 \times (\sigma)^2}} \end{gathered} \tag{2.4}$$

Where $unadapted\ sensitivity_i$ is the generated tuning curve for the i^{th} neuron in *number of neurons*. $unadapted\ sensitivity_i^x$ is the unadapted sensitivity for numerosity x in the tuning curve of the i^{th} neuron in *number of neurons*.

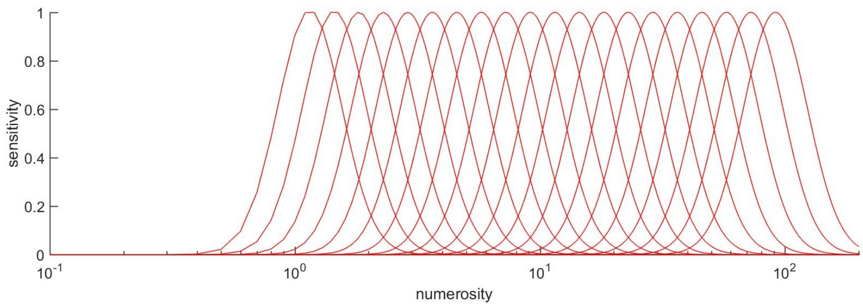


(a) **Gaussian individual tuning curves in linear space**

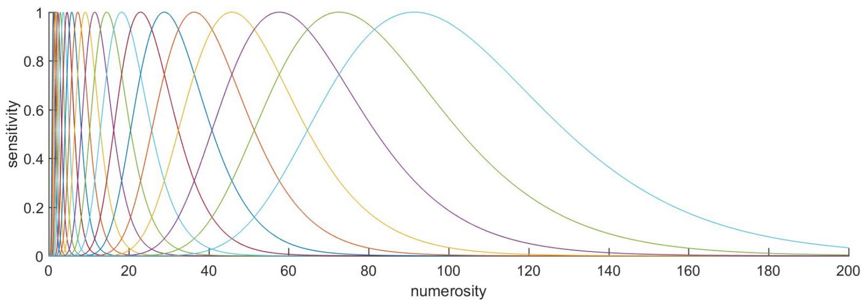


(b) **Gaussian population tuning curve**

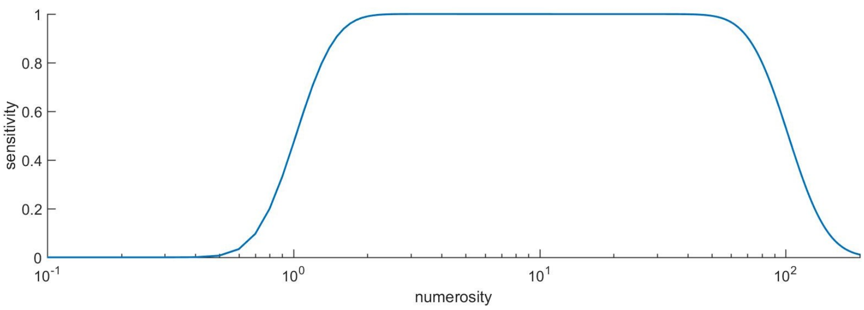
FIGURE 2.1: EXAMPLES OF GENERATED GAUSSIAN TUNING CURVES
Figure (a) shows the individual tuning curves and figure (b) shows the population tuning curve for the Gaussian fit model. Both figures are shown in linear space.



(a) **Log Gaussian individual tuning curves in logarithmic space**



(b) **Log Gaussian individual tuning curves in linear space**



(c) **Log Gaussian population tuning curve**

FIGURE 2.2: EXAMPLES OF GENERATED LOG GAUSSIAN TUNING CURVES
Figures (a) and (b) shown the individual tuning curves for the log Gaussian fit model in respectively logarithmic and linear space. Figure (c) shows the population tuning curve in logarithmic space. Note that in linear space the tuning width grows per subsequent curve for the log Gaussian fit, because value x is used (as opposed to equation 2.3) in the denominator of equation 2.4.

2.4 Adapting the Tuning Curves

The adaptive behavior modeled here follows the principle of the adaptation process found in area V1 and described in the previous section. The model only affects the amplitude property of the individual tuning curves by decreasing the amplitude based on the strength of the adaptation. The other two properties, the width of the tuning curve and the shifting of the tuning curve are not affected. The individual tuning curves retain their width and do not shift from place as shown in figure 2.3 below and as proposed by Jin et al. (2005).

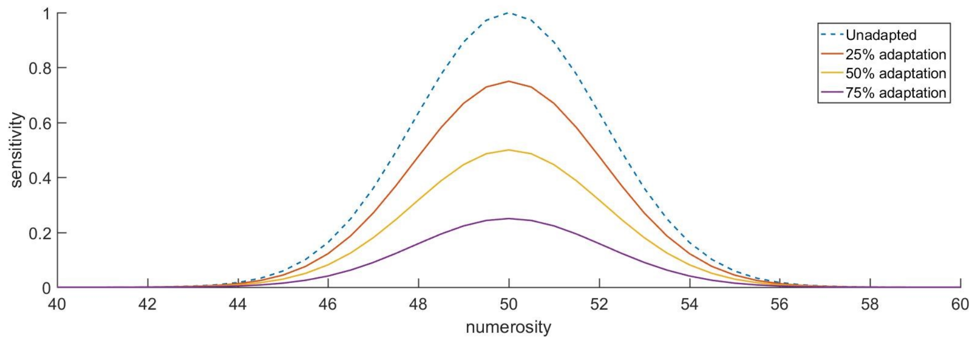


FIGURE 2.3: SIMPLE V1 ADAPTATION FOR AN INDIVIDUAL NEURON
Only the amplitude of the tuning curve decreases while the width remains unchanged and the curve does not shift. For adapter numerosity 50 the tuning curve is adapted at 25%, 50% and 75%. The adapted curves are shown by the red, yellow and purple lines respectively. The blue line shows the unadapted tuning curve.

Since in V1 most neuronal tuning curves follow a Gaussian fit figure 2.3 shows an individual neuronal tuning curve with a Gaussian fit, rather than the log Gaussian fit found for numerosity neurons. The method however, can also be applied to the log Gaussian fitted tuning curves. In logarithmic space the adaptation of an individual tuning curve will be exactly the same as for a Gaussian fitted tuning curve in the linear space show in figure 2.3. Figure 2.4 shows the adaptation for a log Gaussian fitted tuning curve in linear space.

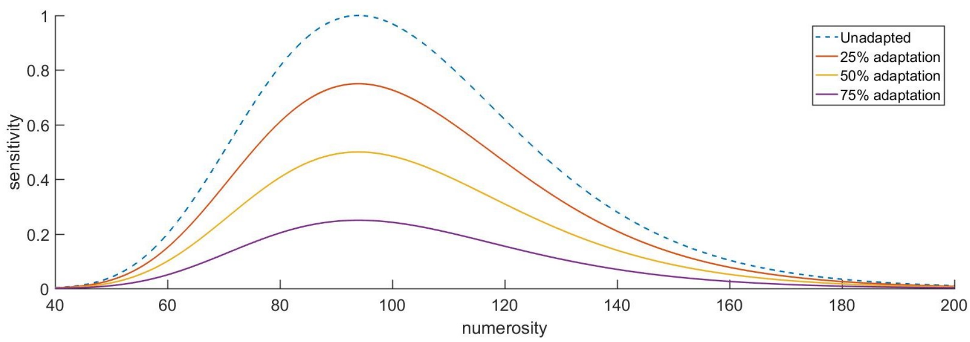


FIGURE 2.4: SIMPLE V1 ADAPTATION FOR A LOG GAUSSIAN NEURON
For adapter numerosity 90 the tuning curve is adapted at 25%, 50% and 75%. The adapted curves are shown by the red, yellow and purple lines respectively. The blue line shows the unadapted tuning curve.

As for our model, it adapts the neurons by changing the previously generated tuning curves of these neurons. Per neuron an adaptation weight is determined. This adaptation weight ensures that neurons tuned to the adapter numerosity are reduced most in amplitude. Moreover, this ensures that the further away neuron is tuned from the adapter numerosity, the less the corresponding curve is reduced in amplitude. The adaptation weight is based on the activation of the adapter numerosity for that specific neuron:

$$adaptation\ weight_i = adaptation\ strength \times activation\ for\ the\ adapter_i \tag{2.5}$$

Where $adaptation\ weight_i$ is the adaptation weight for the tuning curve of the i^{th} neuron in the *number of neurons* and $activation\ for\ the\ adapter_i$ is the activation for the adapter numerosity for the tuning curve of the i^{th} neuron in the *number of neurons*.

The parameter *adaptation strength* is set as a ratio and indicates the adaptation strength for the individual neurons who's numerosity preference is equal to the adapter numerosity. That is to say, the adaptation strength for the individual tuning curve which is tuned to the adapter numerosity. For example, if *adaptation strength* is set to a decrease of 50% the ratio for the adaptation strength would be 0.50.

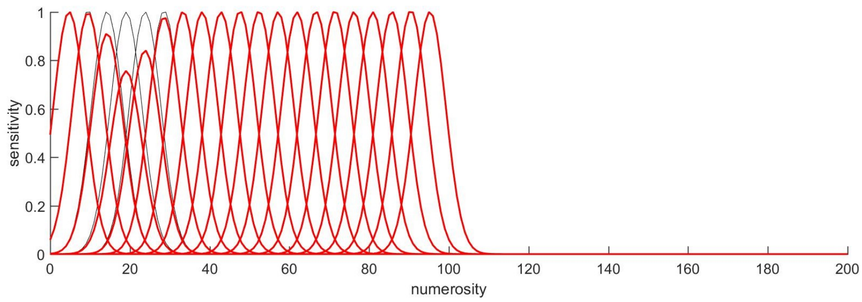
The activation for the adapter numerosity is calculated per neuron as adaptation to any point in the population will reduce the sensitivities of all neurons in the population (Mollon, 1974). The higher the activation, the closer the neuron is tuned to the adapter numerosity and the greater the adaptation should be as it has a larger influence on the overall adapted perception. Similarly, the lower the activation, the further away the neurons is tuned and the smaller its influence on the overall adapted perception. For example, when the *adaptation strength* is set to 50% the neuron with its preference equal to the adapter numerosity will be adapted by 50%. The neurons with preferences surrounding the adapter numerosity should adapt less than 50%.

The adaptation weight is then used to calculate the adapted sensitivity of the previously generated tuning curves:

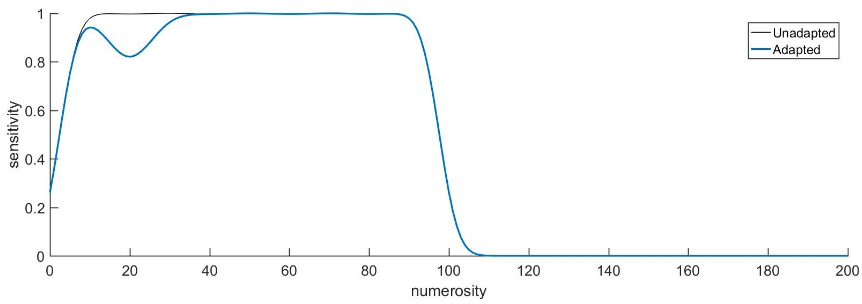
$$adapted\ sensitivity_i^x = unadapted\ sensitivity_i^x \times (1 - adaptation\ weight_i) \tag{2.6}$$

Where $adapted\ sensitivity_i^x$ is the adapted sensitivity for numerosity x in the tuning curve of the i^{th} neuron in *number of neurons*. When the tuning curves do not adapt, the adapted sensitivity will be equal to the unadapted sensitivity. On the contrary, when the *adaptation strength* is set to 100% the adapted sensitivity will be equal to zero.

Continuing with our example parameters, an adaptation strength ratio of 0.25 and an adapter numerosity of 20, figures 2.5 and 2.6 respectively show the adapted tuning curves for the Gaussian and log Gaussian fit. Comparing figures 2.5(a) and 2.6(a) shows that the adaptation works the same way for the individual tuning curves in their respective linear and logarithmic spaces. However, comparing figures 2.5(a) and 2.6(b) shows the different outcomes of the adaptation in linear space using the same method.



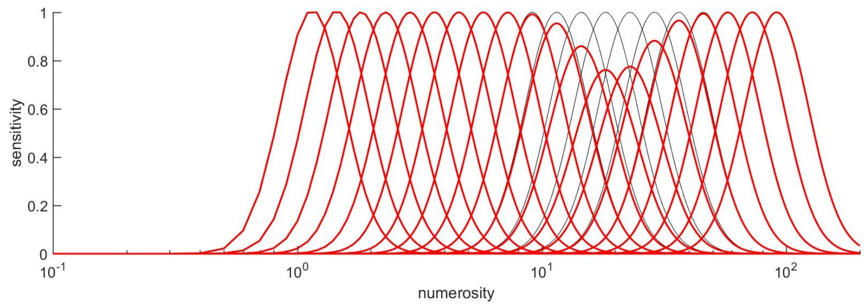
(a) **Adapted Gaussian individual tuning curves in linear space**



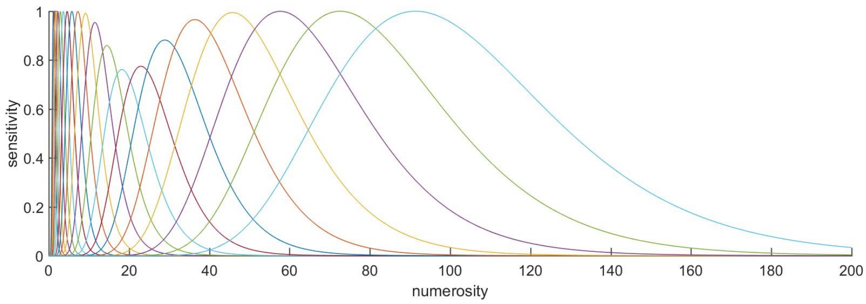
(b) **Adapted Gaussianpopulation tuning curve**

FIGURE 2.5: EXAMPLES OF GENERATED GAUSSIAN TUNING CURVES

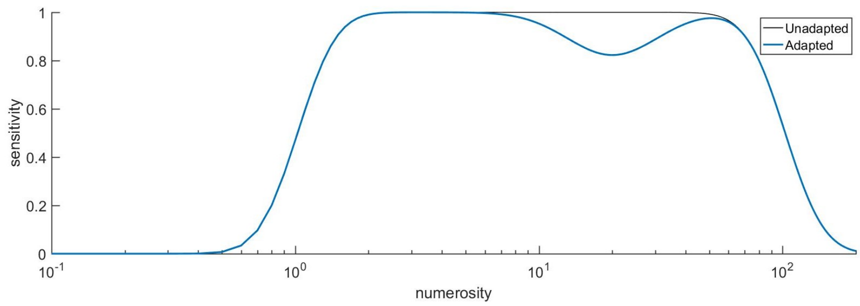
Figure (a) shows the individual tuning curves and figure (b) shows the population tuning curve for the Gaussian fit model. Both figures are shown in linear space. The black lines represent the unadapted curves. In figure (a) the red lines and in figure (b) the blue line represents the adapted curves. The effect of adaptation is shown through the resulting of the tuning curves around the adapter numerosity 20.



(a) **Adapted Log Gaussian individual tuning curves in logarithmic space**



(b) **Adapted Log Gaussian individual tuning curves in linear space**



(c) **Adapted Log Gaussianpopulation tuning curve**

FIGURE 2.6: EXAMPLES OF ADAPTED LOG GAUSSIAN TUNING CURVES
Figures (a) and (b) shown the individual tuning curves for the log Gaussian fit model in respectively logarithmic and linear space. Figure (c) shows the population tuning curve in logarithmic space. In figures (a) and (c) the black lines represent the unadapted curves. In figure (a) the red lines and in figure (c) the blue line represent the adapted curves. The effect of adaptation is shown through the resulting of the tuning curves around the adapter numerosity 20.

2.5 Calculating Excitations and Shifts in Perception

When both the unadapted and adapted tuning curves have been created, the model calculates the population excitation curves for a specified stimulus. The method for calculating the excitation is much the same as calculating adaptation as an excitation weight is used. This excitation weight is based upon the activation of the stimulus for each individual tuning curve:

$$excitation\ weight_i = activation\ for\ the\ stimulus_i \tag{2.7}$$

Where $excitation\ weight_i$ is the excitation weight for the ith neuron and $activation\ for\ the\ stimulus_i$ is the activation of the chosen stimulus numerosity for the i^{th} neuron.

With the excitation weight the population excitation curve is calculated by the following formulas:

$$excitation_i^x = sensitivity_i^x \times excitation\ weight_i \times 10 \tag{2.8}$$

$$population\ excitation^x = \sum_{i=1}^{number\ of\ neurons} (excitation_i^x) \tag{2.9}$$

Where $excitation_i^x$ is the excitation of a stimulus numerosity x for the i^{th} neuron in $number\ of\ neurons$. $sensitivity_i^x$ can refer to $unadapted\ sensitivity_i^x$ or $adapted\ sensitivity_i^x$ depending on which population excitation curve is calculated. $population\ excitation^x$ is the population excitation for the numerosity stimulus x.

From the population excitation curves the peaks of these curves are calculated. The difference between the unadapted and adapted peaks provides an estimation for a shift in perception. The shift in perception represents an underestimation or overestimation of the stimulus after adaptation compared to before adaptation. A positive peak shift corresponds to overestimation of a stimulus after adaptation, whereas a negative peak shift corresponds to underestimation.

2.6 Simulating the Research Question

Having specified the model and an implementation of it in MATLAB we first need to know whether the implementation of our model works; a validity check must be done for the Gaussian fit model. If the model shows adaptive behavior which corresponds with earlier research the model is valid.

After validating the model, the optimum combinations of tuning width and adaptation strength must be determined to answer the research question. In order to find the optimum combinations the parameters σ for tuning width and *adaptation strength* for adaptation strength must be varied per simulation. The other parameters of the model must remain the same, else the combinations cannot be compared against each other. A series of simulations will provide shifts in perception which can be compared to earlier research to determine the optimum combinations.

Since it is impossible to count the exact number of neurons in the brain, the *number of neurons* used in the simulation should not highly influence the outcome. To check this, two values for *number of neurons* will be used in the simulations to determine if this parameter might also influence the shift in perception.

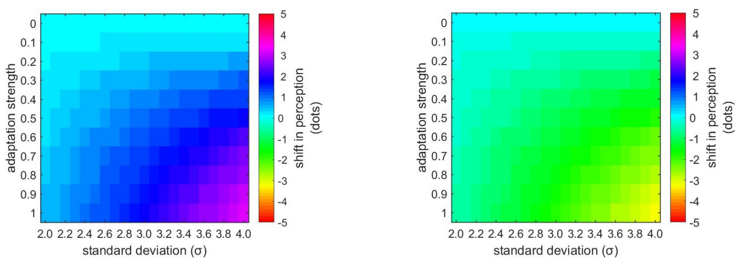
3 Results

3.1 Baseline Gaussian Curves

TABLE 3.1: PARAMETERS USED IN THE GAUSSIAN FIT SIMULATION.
In the right column the parameter name is shown while in the left column the parameter value used is shown.

Parameter	Values used
numerosity range	0-160
numerosity range step size	0.1
maximum peak numerosity	120
σ	2.0 - 4.0 incremented by 0.1
number of neurons	50 and 100
adaptation strength	0% - 100% incremented by 10%
stimulus	40
adapter	35 and 45

Four different conditions were used to test the validity of the model and the simulations were run with the parameters specified in table 3.1. The adapter numerosities 35 and 45 are chosen because they are equally apart from the chosen stimulus numerosity of 40. Two different values for *number of neurons* are chosen to discern if this parameter influences the shift in perception. The simulations were run for only the Gaussian fit model. 21 different values for σ and 11 different values for *adaptation strength* resulted in 231 combinations per condition and 924 simulations of the Gaussian fit model in total. The resulting shifts in perception found for the different combinations of σ and *adaptation strength* are shown in figure 3.1.



(a) Adapter 35 and 50 neurons (b) Adapter 45 and 50 neurons

FIGURE 3.1: PEAK SHIFTS IN PERCEPTION FOR STIMULUS 40
Figure (a) shows the shifts for adapter numerosity 35 and figure (b) shows the shifts for adapter numerosity 45. The shifts are shown per combination of adaptation strength and σ value. The underlying values of the color maps can be found in appendix B.1.

The *number of neurons* seems to have almost no effect on the shift in perception (see appendix B.1). The larger the σ, the larger the standard deviation and the width of the tuning curves and figure 3.1 shows that the larger the σ, the larger the shift in perception is. As hypothesized the tuning width of the course seems to influence the shift in perception. Moreover, the stronger the *adaptation strength*, the larger the shift in perception is as well. The largest shift in perception is found for the combination of the largest σ and the strongest *adaptation strength*. Furthermore, figure 3.1 shows that, as expected, the shifts found for adapter numerosities 35 and 45 are identical, safe that for adapter 35 the shift is positive whereas for adapter 45 the shift is negative. Overall the simulations show that the adaptation behavior for the model behaves as found in earlier research for neurons in the V1 area (Mollon, 1974; Thompson & Burr, 2009; Jin et al., 2005).

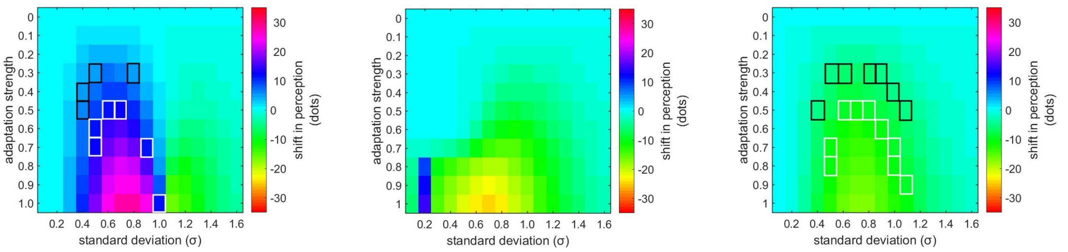
3.2 Numerosity Log Gaussian Curves

To simulate the research question six different conditions were chosen to match the conditions of earlier research (Aagten-Murphy & Burr, 2016; Tsouli et al., 2018). The simulations were run with the parameters shown in table 3.2. The two different values 50 and 100 were chosen for *number of neurons* to discern if the parameter influences the shift in perception. The simulations were run only for the log Gaussian fit model. For the parameter σ 16 different values were used and for the parameter *adaptation strength* 11 different values. This resulted in 176 combinations per condition, totaling in 1056 simulations of the model. The shifts in perception found per combination of tuning width and adaptation strength are shown in figure 3.2.

TABLE 3.2: PARAMETERS USED IN THE LOG GAUSSIAN SIMULATIONS
In the right column the parameter name is shown while in the left column the parameter value used is shown.

Parameter	Values used
numerosity range	0-250
numerosity range step size	0.1
maximum peak numerosity	160
σ	0.1 - 1.6 incremented by 0.1
number of neurons	50 and 100
adaptation strength	0% - 100% incremented by 10%
stimulus	40
adapter	20, 40 and 80

Congruent with the results found for the Gaussian fit model, there is almost no difference in the shifts in perception found for conditions with the parameter *number of neurons* set to 50 compared to the conditions with *number of neurons* set to 100 (see appendix B.2). Also, as expected, figure 3.2 shows that the shifts in perception are positive for adapter numerosity 20 and negative for adapter numerosity 80 representing overestimation and underestimation after adaptation, respectively.



(a) Adapter 20 and 50 neurons (b) Adapter 40 and 50 neurons (c) Adapter 80 and 50 neurons

FIGURE 3.2: PEAK SHIFTS IN PERCEPTION FOR STIMULUS 40
The figures show the shifts in perception found for the three conditions with 50 neurons. The shifts are shown per combination of adaptation strength and σ value. In figures (a) and (c) the shifts in perception matching the results found by Tsouli et al. (2018) and Aagten-Murphy and Burr (2016) are shown by the black and white squares, respectively. The underlying values of the color maps can be found in appendix B.2.

3.2.1 Adapter Numerosity 20

Figure 3.2(a) shows that there are quite a few combinations of σ and *adaptation strength* for which a shift of around 5 dots occurs as indicated by the black squared markings. Furthermore, the asymmetrical shift found by Aagten-Murphy and Burr (2016) is also found for certain combinations as indicated by the white squared markings. A trend that is clearly visible when looking at the figures is that for all simulations with a *adaptation strength* of 0 or with a σ of 0.1 the shift in perception is so small that it is not perceived. Another trend which holds until the σ value reaches 0.7 is that the larger the σ or stronger the *adaptation strength*, the larger the shift in perception is as well. Between the σ values of 0.8 and 1.0 the shift in perception diminishes again.

A curiosity is that if the sigma is greater than 1.0 the shift in perception shifts to a negative shift rather than the earlier found positive shift, which might be attributed to the change in the form of the tuning curves with a σ value of 1.0 or higher. Figure 3.3 shows that the change in the shape causes the top of the tuning curve to shift, which in the model translates to a shift in the preference of neurons. The shift in preference might cause the adaptation to differ greatly, resulting in the negative shifts in perception rather than the expected positive shifts in perception.

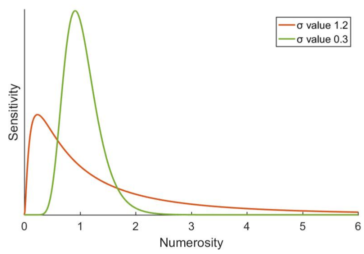


FIGURE 3.3: EXAMPLES OF σ VALUES IN LOG GAUSSIAN FIT CURVES
Two log Gaussian fit curves with a small and large σ value are shown by the green and red curve, respectively. The μ value used for both curves is 0.

3.2.2 Adapter numerosity 40

Tsouli et al. (2018) found an effect on when adapting to numerosity 40. However, the effect was attributed to duration adaptation. Aagten-Murphy and Burr (2016) also adapted to numerosity 40 in their experiment, but the adaptation showed no significant change in the perceived number of dots. Even so, figure 3.2(b) shows an interesting series of shifts in perception for adapter numerosity 40 which seem positive for a σ value of 0.2 and negative for values of σ larger than 0.2. Following this, the reduction of the shift in perception rather than the increase for σ values of 1.0 and larger for adapter numerosities 20 and 80 is also found for adapter numerosity 40. Additionally, for adapter numerosity 40 more combinations of σ and *adaptation strength* values yield a shift in perception of 0.

3.2.3 Adapter Numerosity 80

Figure 3.2(c) shows that for adapter numerosity 80 all shifts in perception found, not including 0, are negative. This is expected because considering an arbitrary adapter α and stimulus β, there should be an underestimation when $\alpha < \beta$ (see section 1.2 and figures 1.2 and 1.3). The overall underestimation of stimulus numerosity 40 after adapting to numerosity 80 is a trend found in earlier research (Aagten-Murphy & Burr, 2016; Castaldi et al., 2016; Tsouli et al., 2018). It is important to note that the switch from positive to negative in the shift in perception that happened for adapter numerosity 20 and a σ value larger than 1.0 does not happen vice versa for adapter numerosity 80. However, just as found for adapter numerosity 20, for adapter 80 the shifts in perception get smaller rather than larger for σ values of 0.7 and larger. Furthermore, the 25% to 30% shift found by Aagten-Murphy and Burr (2016) is also found for certain combinations as marked by the white squares in figure 3.2(c).

3.2.4 Optimum Combination of Tuning Width and Adaptation Strength

Considering the different combinations found for adapter numerosities 20 and 80 that produce shifts in perception matching the earlier found results, a comparison of the combinations yield the following combinations for which the symmetrical 5 dots shift in perception reported by Tsouli et al. (2018) is found:

- A *adaptation strength* ratio of 0.3 and a σ of 0.5
- A *adaptation strength* ratio of 0.3 and a σ of 0.8
- A *adaptation strength* ratio of 0.5 and a σ of 0.4

Moreover, the following combinations found the asymmetrical shift reported by Aagten-Murphy and Burr (2016):

- A *adaptation strength* ratio of 0.5 and a σ of 0.6
- A *adaptation strength* ratio of 0.5 and a σ of 0.7
- A *adaptation strength* ratio of 0.7 and a σ of 0.5

Looking at all the combinations of *adaptation strength* ratio and σ found, the values for σ which seem to overlap are between 0.4 and 0.8 while the *adaptation strength* ratios seem to be at the lower end for the shift in perception found by Tsouli et al. (2018) and at the higher end for the shift in perception found by Aagten-Murphy and Burr (2016). In particular, the σ value of 0.5 allows for the reproduction of the results found by both studies.

4 Discussion

The goal was to determine if the model specified in section 2 could explain human numerosity adaptation behavior. Specifically, the purpose was to determine which combinations of tuning curve width and adaptation strength produce the results found in earlier research. That is to say which combinations produce the symmetrical shift in perception found by Tsouli et al. (2018), the asymmetrical shift in perception found by Aagten-Murphy and Burr (2016), and the overall underestimation of a stimulus after adaptation when the stimulus is smaller than the adapter found by Castaldi et al. (2016)?

To answer this question the shift in perception produced by specific combinations of tuning width and adaptation strength were measured. At first, for the shift in perception two different points of measurement were used. Firstly, the shift in perception according to the peak (see section 1.2) as was used by (Jin et al., 2005). Secondly, the shift in perception according to the center of gravity (COG) which relates to the middle point of the area under the population excitation curve. However, the COG shifts in perception found by the model were of a magnitude that was not feasible (see appendix C). Because of this the COG measurement is not used further in this paper.

However, before any shifts in perception could be calculated the model had to be validated. To validate the model, the Gaussian fit variant of the model was tested with four different conditions. The results found for these conditions correspond with typical adaptation behavior found in the V1; a stronger adaptation strength increased the resulting shift in perceived number of dots (Mollon, 1974; Thompson & Burr, 2009; Jin et al., 2005). Because of this the implementation of the model in MATLAB can be considered a valid representation of the mathematical model specified earlier (see section 2).

After validating the model, the log Gaussian fit variation of the model was run for six conditions in order to determine the optimum combinations of tuning width and adaptation strength. Here three adapter numerosities were used (20, 40 and 80) alongside a single stimulus numerosity 40. Congruent with earlier findings (Aagten-Murphy & Burr, 2016; Castaldi et al., 2016; Tsouli et al., 2018), when adapting to numerosity 80 an overall underestimation of the stimulus numerosity 40 was found for all combinations. Furthermore, congruent with earlier psychophysical results an overestimation of the stimulus numerosity 40 was found when adapting to numerosity 20 (Aagten-Murphy & Burr, 2016; Tsouli et al., 2018). These findings do not give an optimum combination of adaptation strength and tuning width, but do show that at least a part of the human adaptive behavior of numerosity can be mimicked through the model.

In the six log Gaussian conditions σ values ranging from 0.1 to 1.6 were used. However, when log Gaussian fit curves are created using a σ value higher than 1.0 the shape of the curve changes (see figure 3.3). This change can also be seen in the color maps of figure 3.2. In the conditions with adapter numerosity 20 the resulting shifts in perception turn negative instead of positive and in the conditions with adapter numerosities 40 and 80 the shifts in perception decrease much faster than with a σ value between 0.7 and 1.0. This shows that using log Gaussian fit tuning curves has a limit. However, since the tuning width of human tuning curves is not known this limit poses a constraint for modeling numerosity adaptation. Furthermore, in the current model the natural logarithm was used to generate the log Gaussian fit tuning curves. However the curves can also be generated with a common base 10 logarithm which might lessen the constraint. However, because another logarithm is used the resulting shift in perception could also be influenced greatly. To determine the influence of the logarithm a model can be implemented which substitutes the natural logarithms used in the current model and then it can be researched if the logarithm greatly influences the shift in perception.

Another curious finding for the conditions with adapter numerosity 40 is that for the higher *adaptation strength* values and a σ value of 0.2 the shift in perception is positive rather than negative. For these parameter values the overlap between the individual tuning curves is just right that after adaptation less tuning curves overlap than for the other combinations. Because of this, less neurons contribute to the population's response causing the shift in perception to become positive. Even so, a definite trend can be seen when comparing an increase in tuning width or adaptation strength with the resulting shift in perception. For all conditions, except for the aforementioned curiosity, the shift in perception increases when the adaptation strength or tuning width increases. However, after a σ value of about 0.7 the shifts in perception decrease, following the theory discussed in section 1.3 that very wide tuning curves level the shift in perception.

To answer the research question, one optimum combination was found that produces the symmetrical shift in perception of about 5 dots in agreement with Tsouli et al. (2018) and produces the asymmetrical shift in perception found by Aagten-Murphy and Burr (2016) when adapting to numerosities 20 and 80. This further shows that the model can mimic human behavior for at least the stimulus numerosity 40. However, in order to fully explain numerosity adaptation through the model presented here other stimulus numerosities should be considered. A possible way to check the model's relevance is to increase the data to test the model by using stimulus numerosities which are either much lower (such as 4 or 5) or much higher (such as 80) than the numerosity 40 used here.

It is worth mentioning that the differing shifts in perception found by Tsouli et al. (2018) and Aagten-Murphy and Burr (2016) were produced through the same exposure duration of adaptation (333 msec). This supports the idea that numerosity adaptation does not only require exposure duration and that adaptation strength does not equal exposure duration (Aagten-Murphy & Burr, 2016). This should be taken into account when devising new experiments for numerosity adaptation based upon the model.

Finally, the results show that modeling is a great tool to use in order to bridge the gap between the earlier found data. It provides a way to easily and quickly compare different datasets. Furthermore, the results can be used for predictions of numerosity adaptation behavior which can help the understanding of the numerosity system.

References

Aagten-Murphy, D., & Burr, D. (2016). Adaptation to numerosity requires only brief exposures, and is determined by number of events, nor exposure duration. *Journal of Vision*, *16*(10), 1–14. doi: 10.1167/16.10.22

Agrillo, C., Piffer, L., Bisazza, A., & Butterworth, B. (2012). Evidence for two numerical systems that are similar in humans and guppies. *PLoS ONE*, *7*(2), e31923. doi: 10.1371/journal.pone.0031923

Anobile, G., Cicchini, G., & Burr, D. (2016). Number as a primary perceptual attribute: A review. *Perception*, *45*(1-2), 5–31. doi: 10.1177/0301006615602599

Arrighi, R., Togoli, I., & Burr, D. (2014). A generalized sense of number. *Proceedings of the Royal Society B*, *281*, 20141791. doi: 10.1098/rspb.2014.1791

Barth, H., La Mont, K., Lipton, J., & Spelke, E. (2005). Abstract number and arithmetic in preschool children. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(39), 14116–14121. doi: 10.1073/pnas.0505512102

Burr, D., & Ross, J. (2008). Report a visual sense of number. *Current Biology*, *18*(6), 425-–428. doi: 10.1016/j.cub.2008.02.052

Cantlon, J., & Brannon, E. (2007). Basic math in monkeys and college students. *PLOS Biology*, *5*(12), e328. doi: 10.1371/journal.pbio.0050328

Castaldi, E., Aagten-Murphy, D., Tosetti, M., Burr, D., & Morrone, M. (2016). Effects of adaptation on numerosity decoding in the human brain. *NeuroImage*, *143*, 364–377. doi: 10.1016/j.neuroimage.2016.09.020

Clifford, C., Webster, M., Stanley, G., Stocker, A., Kohn, A., Sharpee, T., & Schwartz, O. (2007). Visual adaptation: Neural, psychological and computational aspects. *Vision Research*, *47*, 3125–3131. doi: 10.1016/j.visres.2007.08.023

Grill-Spector, K., Henson, R., & Martin, A. (2006). Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Sciences*, *10*(1), 14–23. doi: 10.1016/j.tics.2005.11.006

Harvey, B., & Dumoulin, S. (2017). A network of topographic numerosity maps in human association cortex. *Nature Human Behaviour*, *1*, 0036. doi: 10.1038/s41562-016-0036

Harvey, B., Fracasso, A., Petridou, N., & Dumoulin, S. (2015). Topographic representations of object size and relationships with numerosity reveal generalized quantity processing in human parietal cortex. *Proceedings of the National Academy of Sciences*, *112*(44), 13525–13530. doi: 10.1073/pnas.1515414112

Harvey, B., Petridou, N., & Dumoulin, S. (2013). Topographic representation of numerosity in the human parietal cortex. *Science*, *341*(6150), 1123–1126. doi: 10.1126/science.1239052

Heron, J., Aaen-Stockdale, C., Hotchkiss, J., Roach, N., McGraw, P., & Whitaker, D. (2011). Duration channels mediate human time perception. *Proceedings of the Royal Society B*, *279*(1729), 690–698. doi: 10.1098/rspb.2011.1131

Jin, D., Dragoi, V., Sur, M., & Seung, H. (2005). Tilt aftereffect and adaptation-induced changes in orientation tuning in visual cortex. *Journal of Neurophysiology*, *94*, 4038-–4050. doi: 10.1152/jn.00571.2004

Jones, S., Pearson, J., Dewind, N., Paulsen, D., Tenekedjieva, A., & Brannon, E. (2013). Lemurs and macaques show similar numerical sensitivity. *Animal Cognition*, *17*(3), 503–515. doi: 10.1007/s10071-013-0682-3

King, J., & Crowder, N. (2018). Adaptation to stimulus orientation in mouse primary visual cortex. *European Journal of Neuroscience*, *47*, 346–357. doi: 10.1111/ejn.13830

Kohn, A. (2007). Visual adaptation: Physiology, mechanisms, and functional benefits. *Journal of Neurophysiology*, *97*(5), 3155–3164. doi: 10.1152/jn.00086.2007

Kohn, A., & Movshon, J. (2004). Adaptation changes the direction tuning of macaque mt neurons. *Nature Neuroscience*, *7*(7), 764–772. doi: 10.1038/nn1267

Krekelberg, B., van Wezel, R., & Albright, T. (2006). Adaptation in macaque mt reduces perceived speed and improves speed discriminations. *Journal of Neurophysiology*, *95*(1), 255–270. doi: 10.1152/jn.00750.2005

Mease, R., Famulare, M., Gjorgjieva, J., Moody, W., & Fairhall, A. (2013). Emergence of adaptive computation by single neurons in the developing cortex. *The Journal of Neuroscience*, *33*(30), 12154–12170. doi: 10.1523/JNEUROSCI.3263-12.2013

Mollon, J. (1974). After-effects and the brain. *New Scientist*, *61*(886), 479—482. Retrieved from `http://vision.psychol.cam.ac.uk/jdmollon/papers/Mollon1974.pdf`

Parra, L., & Pearlmutter, B. (2007). Illusory percepts from auditory adaptation. *The Journal of the Acoustical Society of America*, *121*(3), 1632–1641. doi: 10.1121/1.2431346

Quiroga, M. d. M., Morris, A., & Krekelberg, B. (2016). Adaptation without plasticity. *Cell Reports*, *17*(1), 58–68. doi: 10.1016/j.celrep.2016.08.089

Schweinberger, S., Casper, C., Hauthal, N., Kaufmann, J., Kawahara, H., Kloth, N., ... Zäske, R. (2008). Auditory adaptation in voice perception. *Current Biology*, *18*(9), 684–688. doi: 10.1016/j.cub.2008.04.015

Thompson, P., & Burr, D. (2009). Visual aftereffects. *Current Biology*, *19*(1), R11–R14. doi: 10.1016/j.cub.2008.10.014

Tsouli, A., Dumoulin, S., te Pas, S., & van der Smagt, M. (2018). Adaptation reveals unbalanced interaction between numerosity and time. *Cortex*. doi: 10.1016/j.cortex.2018.02.013

Westrick, Z., Heeger, D., & Landy, M. (2016). Pattern adaptation and normalization reweighting. *The Journal of Neuroscience*, *36*(38), 9805—9816. doi: 10.1523/JNEUROSCI.1067-16.2016

Wilson, H., & Humanski, R. (1993). Spatial frequency adaptation and gain control. *Vision Res*, *33*(8), 1133–1149. doi: 10.1016/0042-6989(93)90248-U

Xu, F., & Spelke, E. (2000). Large number discrimination in 6-month-old infants. *Cognition*, *74*(1), B1–B11. doi: 10.1016/s0010-0277(99)00066-9

Appendices

A A MATLAB interpretation

A.1 Implementation of a Log Gaussian Fit Model

```
function output = run_lognormal(adapter,stimulus,adaptation_strength,neurons,maxnumerosity,sigma,x)
% RUN_LOGNORMAL Run a simulation of the model which creates and adapts
% log Gaussian numerosity neural tuning curves based on the parameters:
%
%   ADAPTER = the numerosity value to which the numerosity tuning curves should adapt
%   ADAPTER always needs to be specified
%
%   STIMULUS = the numerosity value for which the exitation is calculated
%   STIMULUS always needs to be specified
%
%   ADAPTATION_STRENGTH = the maximum amount of adaptation that the neuron with
%   the adapter as preference adapts. Must be given as a ratio, e.g. 0.5
%   for 50% adaptation
%   If not specified, default ADAPTATION_STRENGTH is set to 0.5
%
%   NEURONS = the amount of neuronal tuning curves generated by the model
%   If not specified, default NEURONS is set to 100
%
%   MAXNUMEROSITY = the maximum numerosity preference that a neuron can reach
%   If not specified, default MAXNUMEROSITY is set to 100
%
%   SIGMA = the standard deviation used for calculation of the width of the
%   individual lognormal tuning curves
%   If not specified, default SIGMA is set to 0.5;
%
%   X = the rangeview of numerosities over which tuning curves can spread.
%   Should be given in the form of an array of integers such as:
%               [beginnumerosity:stepsize:endnumerosity]
%   If not specified, default X is set to [0:0.1:160]
%
% Include paramters as the following sequence:
%   RUN_LOGNORMAL(adapter,stimulus,adaptation_strength,neurons,maxnumerosity,sigma,x)
%
% See also RUN_NORMAL and RUNME.

%% Default values if not detailed in function:
%If only adapter and stimuslus are specified
if nargin == 2
    adaptation_strength = 0.5;
    neurons = 100;
    maxnumerosity = 100;
    sigma = 0.5;
    x = [0:0.1:160];
%If only adapter, stimulus and adaptation_strength are specified
elseif nargin == 3
    neurons = 100;
    maxnumerosity = 100;
    sigma = 0.5;
    x = [0:0.1:160];
%If only adapter, stimulus, adaptation_strength and neurons are specified
elseif nargin == 4
    maxnumerosity = 100;
    sigma = 0.5;
    x = [0:0.1:160];
%If only adapter, stimulus, adaptation_strength, neurons and maxnumerosity are specified
elseif nargin == 5
    sigma = 0.5;
    x = [0:0.1:160];
```

```
%If only adapter, stimulus, adaptation_strength, neurons, maxnumerosity and sigma are specified
elseif nargin == 6
    x = [0:0.1:160];
end

%% Creating the unadapted tuning curves (individual, population and normalised population curves)
% Calculating sensitivity for the individual tuning curves per neuron:
for neuron = 1:neurons
    %Calculate the mean of a neuron
    mean(neuron) = (log(maxnumerosity)/max(neurons)) * neuron;
    %Setting the standard deviation of a neuron
    std = sigma;
    %Calculate the sensitivity of every x-value per neuron
    sensitivity(neuron,:) = lognpdf(x,mean(neuron),std);
    %Determine the maximum value of the neuron
    topval(neuron) = max(sensitivity(neuron,:));
    %Scale the tuning function so that the maximum sensitivity is 1
    yval1(neuron,:) = sensitivity(neuron,:)/topval(neuron);
end

% Calculate the sensitivity for the normalised population tuning curve
totaltune = sum(yval1);
noramlised_totaltune = totaltune/max(totaltune);

%% Adapting the generated tuning curves according to V1 adaptation
% Find the index number for the stimulus specified in the function in the
% rangeview of numerosities in order to determine adaptation weights
adapter_in_array = find(x==adapter);

% Adapt the amplitude of the individual tuning curves
for adapted_neuron = 1:neurons
    %Determine the adaptation_weight based on the activation of the neuron
    %for the x-value of the adapter numerosity
    adaptation_weight = adaptation_strength * yval1(adapted_neuron,adapter_in_array);
    %Calculate the adapted sensitivity with the adaptation weight
    yval2(adapted_neuron,:) = yval1(adapted_neuron,:) * (1 - adaptation_weight);
end

% Calculate the sensitivity for the adapted normalised population tuning curve
adapted_totaltune = sum(yval2);
adapted_noramlised_totaltune = adapted_totaltune/max(adapted_totaltune);

%% Create the population exitation curves for the unadapted and adapted tuning curves
% Find the index number for the stimulus specified in the function in the
% rangeview of numerosities in order to determine exitation weights
stimulus_in_array = find(x==stimulus);

% Calculate the unadapted individual exitation curves
for neuron = 1:neurons
    %Determine the excitation_weight based on the activation of the neuron
    %for the x-value of the adapter numerosity
    exitation_weight = yval1(neuron,stimulus_in_array);
    %Calculate the exitation curve based on the exitation weight
    exitation(neuron,:) = yval1(neuron,:) * (exitation_weight) *10;
end

% Calculate the adapted individual exitation curves
for neuron = 1:neurons
    adapted_exitation_weight = yval2(neuron,stimulus_in_array);
    adapted_exitation(neuron,:) = yval2(neuron,:) * (adapted_exitation_weight) *10;
end

% Calculate the unadapted and adapted population exitation curve
totalexitation = sum(exitation);
totaladaptedexitation = sum(adapted_exitation);

%% Calculate the perception shifts (peak and cog):
%Calculate the perceptionshift according to the peak of the exitation curve
findpeakbefore = find(totalexitation==max(totalexitation));
findpeakafter = find(totaladaptedexitation==max(totaladaptedexitation));
peakbefore = x(findpeakbefore);
peakafter = x(findpeakafter);
```

```
perceptionshiftpeak = peakafter - peakbefore;

%Calculate the perceptionshift according to the center of gravity:
cogbefore = COG(totalexitation);
cogafter = COG(totaladaptedexitation);
perceptionshiftcog = cogafter - cogbefore;

%% Create and fill the output structure with relevant info:
%Create the output structure:
output = struct('input',{},'perceptionshiftpeak',{},'perceptionshiftcog',{},'individualsensitivitybefore',{
%Create an input structure as part of the output structure:
output(1).input = struct('adapter',{},'stimulus',{},'adaptation_strength',{},'neurons',{},'maxnumerosity',{

%Fill the output structure:
output(1).input(1).adapter = adapter;
output(1).input(1).stimulus = stimulus;
output(1).input(1).adaptation_strength = adaptation_strength;
output(1).input(1).neurons = neurons;
output(1).input(1).maxnumerosity = maxnumerosity;
output(1).input(1).standarddeviation = sigma;
output(1).input(1).numerosityrange = x;
output(1).perceptionshiftpeak = perceptionshiftpeak;
output(1).perceptionshiftcog = perceptionshiftcog;
output(1).individualsensitivitybefore = yval1;
output(1).individualsensitivityafter = yval2;
output(1).populationsensitivitybefore = noramlised_totaltune;
output(1).populationsensitivityafter = adapted_noramlised_totaltune;
output(1).populationexitationcurvebefore = totalexitation;
output(1).populationexitationcurveafter = totaladaptedexitation;
```

A.2 Implementation of a Gaussian Fit Model

```
function output = run_normal(adapter,stimulus,adaptation_strength,neurons,maxnumerosity,sigma,x)
% RUN_NORMAL Run a simulation of the model which creates and adapts
% Gaussian numerosity neural tuning curves based on the parameters:
%
%   ADAPTER = the numerosity value to which the numerosity tuning curves should adapt
%   ADAPTER always needs to be specified
%
%   STIMULUS = the numerosity value for which the exitation is calculated
%   STIMULUS always needs to be specified
%
%   ADAPTATION_STRENGTH = the maximum amount of adaptation that the neuron with
%   the adapter as preference adapts. Must be given as a ratio, e.g. 0.5
%   for 50% adaptation
%   If not specified, default ADAPTATION_STRENGTH is set to 0.5
%
%   NEURONS = the amount of neuronal tuning curves generated by the model
%   If not specified, default NEURONS is set to 100
%
%   MAXNUMEROSITY = the maximum numerosity preference that a neuron can reach
%   If not specified, default MAXNUMEROSITY is set to 100
%
%   SIGMA = the standard deviation used for calculation of the width of the
%   individual lognormal tuning curves
%   If not specified, default SIGMA is set to 2.1;
%
%   X = the rangeview of numerosities over which tuning curves can spread.
%   Should be given in the form of an array of integers such as:
%               [beginnumerosity:stepsize:endnumerosity]
%   If not specified, default X is set to [0:0.1:160]
%
% Include paramters as the following sequence:
%   RUN_NORMAL(adapter,stimulus,adaptation_strength,neurons,maxnumerosity,sigma,x)
%
% See also RUN_LOGNORMAL and RUNME.
```

```
%% Default values if not detailed in function:
%If only adapter and stimuslus are specified
if nargin == 2
    adaptation_strength = 0.5;
    neurons = 100;
    maxnumerosity = 100;
    sigma = 2.1;
    x = [0:0.1:160];
%If only adapter, stimulus and adaptation_strength are specified
elseif nargin == 3
    neurons = 100;
    maxnumerosity = 100;
    sigma = 2.1;
    x = [0:0.1:160];
%If only adapter, stimulus, adaptation_strength and neurons are specified
elseif nargin == 4
    maxnumerosity = 100;
    sigma = 2.1;
    x = [0:0.1:160];
%If only adapter, stimulus, adaptation_strength, neurons and maxnumerosity are specified
elseif nargin == 5
    sigma = 2.1;
    x = [0:0.1:160];
%If only adapter, stimulus, adaptation_strength, neurons, maxnumerosity and sigma are specified
elseif nargin == 6
    x = [0:0.1:160];
end

%% Creating the unadapted tuning curves (individual, population and normalised population curves)
% Calculating sensitivity for the individual tuning curves per neuron:
for neuron = 1:neurons
    %Calculate the mean of a neuron
    mean(neuron) = (neuron*(maxnumerosity/(neurons+1)));
    %Setting the standard deviation of a neuron
    std = sigma;
    %Calculate the sensitivity of every x-value per neuron
    sensitivity(neuron,:) = normpdf(x,mean(neuron),std);
    %Determine the maximum value of the neuron
    topval(neuron) = max(sensitivity(neuron,:));
    %Scale the tuning function so that the maximum sensitivity is 1
    yval1(neuron,:) = sensitivity(neuron,:)/topval(neuron);
end

% Calculate the sensitivity for the normalised population tuning curve
totaltune = sum(yval1);
noramlised_totaltune = totaltune/max(totaltune);

%% Adapting the generated tuning curves according to V1 adaptation
% Find the index number for the stimulus specified in the function in the
% rangeview of numerosities in order to determine adaptation weights
adapter_in_array = find(x==adapter);

% Adapt the amplitude of the individual tuning curves
for adapted_neuron = 1:neurons
    %Determine the adaptation_weight based on the activation of the neuron
    %for the x-value of the adapter numerosity
    adaptation_weight = adaptation_strength * yval1(adapted_neuron,adapter_in_array);
    %Calculate the adapted sensitivity with the adaptation weight
    yval2(adapted_neuron,:) = yval1(adapted_neuron,:) * (1 - adaptation_weight);
end

% Calculate the sensitivity for the adapted normalised population tuning curve
adapted_totaltune = sum(yval2);
adapted_noramlised_totaltune = adapted_totaltune/max(adapted_totaltune);

%% Create the population exitation curves for the unadapted and adapted tuning curves
% Find the index number for the stimulus specified in the function in the
% rangeview of numerosities in order to determine exitation weights
stimulus_in_array = find(x==stimulus);

% Calculate the unadapted individual exitation curves
for neuron = 1:neurons
```

```
    %Determine the excitation_weight based on the activation of the neuron
    %for the x-value of the adapter numerosity
    exitation_weight = yval1(neuron,stimulus_in_array);
    %Calculate the exitation curve based on the exitation weight
    exitation(neuron,:) = yval1(neuron,:) * (exitation_weight) *10;
end

% Calculate the adapted individual exitation curves
for neuron = 1:neurons
    adapted_exitation_weight = yval2(neuron,stimulus_in_array);
    adapted_exitation(neuron,:) = yval2(neuron,:) * (adapted_exitation_weight) *10;
end

% Calculate the unadapted and adapted population exitation curve
totalexitation = sum(exitation);
totaladaptedexitation = sum(adapted_exitation);

%% Calculate the perception shifts (peak and cog):
%Calculate the perceptionshift according to the peak of the exitation curve
findpeakbefore = find(totalexitation==max(totalexitation));
findpeakafter = find(totaladaptedexitation==max(totaladaptedexitation));
peakbefore = x(findpeakbefore);
peakafter = x(findpeakafter);
perceptionshiftpeak = peakafter - peakbefore;

%Calculate the perceptionshift according to the center of gravity:
cogbefore = COG(totalexitation);
cogafter = COG(totaladaptedexitation);
perceptionshiftcog = cogafter - cogbefore;

%% Create and fill the output structure with relevant info:
%Create the output structure:
output = struct('input',{},'perceptionshiftpeak',{},'perceptionshiftcog',{},'individualsensitivitybefore',{
%Create an input structure as part of the output structure:
output(1).input = struct('adapter',{},'stimulus',{},'adaptation_strength',{},'neurons',{},'maxnumerosity',{

%Fill the output structure:
output(1).input(1).adapter = adapter;
output(1).input(1).stimulus = stimulus;
output(1).input(1).adaptation_strength = adaptation_strength;
output(1).input(1).neurons = neurons;
output(1).input(1).maxnumerosity = maxnumerosity;
output(1).input(1).standarddeviation = sigma;
output(1).input(1).numerosityrange = x;
output(1).perceptionshiftpeak = perceptionshiftpeak;
output(1).perceptionshiftcog = perceptionshiftcog;
output(1).individualsensitivitybefore = yval1;
output(1).individualsensitivityafter = yval2;
output(1).populationsensitivitybefore = noramlised_totaltune;
output(1).populationsensitivityafter = adapted_noramlised_totaltune;
output(1).populationexitationcurvebefore = totalexitation;
output(1).populationexitationcurveafter = totaladaptedexitation;
```

A.3 Implementation of a Multiple Parameters Model

```
function output = runme(distribution,adapters,stimuli,max_adaptations,neuronranges,maxnumerosities,sigmas,x
% RUNME Run a calculated series of simulations based on the different values
% given for the parameters specified:
%
%   DISTRIBUTION = the distribution used to generate the tuning curves
%   DISTRIBUTION can be one of two values: 'normal' or 'lognormal'
%   DISTRIBUTION should always be specified
%
%   ADAPTERS = an array containing the different numerosity values to which
%   the simulated tuning curves should adapt
%   ADAPTERS always needs to be specified
%
%   STIMULI = an array containing the different numerosity values for which
%   the unadapted and adapted exitation is calculated
%   STIMULI always needs to be specified
%
%   MAX_ADAPTATIONS = an array containing the different maximum amounts of
%   adaptation that the neuron with the adapter as preference adapts to.
%   Must be given as an array of ratios, e.g. [0.5] for 50% adaptation
%   If not specified, default MAX_ADAPTATION is set to [0.5]
%
%   NEURONRANGES = an array containing the different amounts of neuronal tuning
%   curves generated by the model.
%   If not specified, default NEURONRANGES is set to [100]
%
%   MAXNUMEROSITIES = an array containing the different maximum numerosity
%   preferences that the tuning curves may reach.
%   If not specified, default MAXNUMEROSITY is set to [100]
%
%   SIGMAS = an array containing the different standard deviation values
%   used for the calculation of the widths of the neuronal tuning curves
%   If not specified, default SIGMA is set to [0.5]
%
%   XS = an array containing the rangeview numersoty arrays over which the
%   the tuning curves can spread. Each rangeview should be given in the
%   from of an integer array: [beginnumerosity:stepsize:endnumerosity]
%   If not specified, default X is set to [[0:0.1:160]]
%
% Include the parameters as the following sequence:
%   RUNME(distribution,adapters,stimuli,max_adaptations,neuronranges,maxnumerosities,sigmas,xs)
%
% See also RUN_NORMAL and RUN_LOGNORMAL.

%% Default values if not detailed in function:
%If only distribution, adapters and stimuli are specified
if nargin == 3
    max_adaptations = [0.5];
    neuronranges = [100];
    maxnumerosities = [100];
    sigmas = [0.5];
    xs = [[0:0.1:160]];
%If only distribution, adapters, stimuli and max_adaptations are specified
elseif nargin == 4
    neuronranges = [100];
    maxnumerosities = [100];
    sigmas = [0.5];
    xs = [[0:0.1:160]];
%If only distribution, adapters, stimuli, max_adaptations and neuronranges are specified
elseif nargin == 5
    maxnumerosities = [100];
    sigmas = [0.5];
    xs = [[0:0.1:160]];
%If only distribution, adapters, stimuli, max_adaptations, neuronranges and maxnumerosities are specified
elseif nargin == 6
    sigmas = [0.5];
    xs = [[0:0.1:160]];
%If only distribution, adapters, stimuli, max_adaptations, neuronranges, maxnumerosities and sigmas are sp
elseif nargin == 7
    xs = [[0:0.1:160]];
end
```

```
%The two distributions to check for
distributionchecklognormal = 'lognormal';
distributionchecknormal = 'normal';


%% Transfer the input data into an array of structures
simulations = length(adapters)*length(stimuli)*length(max_adaptations)*length(neuronranges)*length(maxnum
input = struct('adapter',{},'stimulus',{},'max_adaptation',{},'neurons',{},'maxnumerosity',{},'sigma',{},'x
n=1;

for adapter = adapters
    for stimulus = stimuli
        for max_adaptation = max_adaptations
            for neurons = neuronranges
                for maxnumerosity = maxnumerosities
                    for inputsigma = sigmas
                        for x = 1:size(xs,1)
                            input(n).adapter = adapter;
                            input(n).stimulus = stimulus;
                            input(n).max_adaptation = max_adaptation;
                            input(n).neurons = neurons;
                            input(n).maxnumerosity = maxnumerosity;
                            input(n).sigma = inputsigma;
                            input(n).x = xs;
                            n = n+1;
                        end
                    end
                end
            end
        end
    end
end

%% Run the simulations
% Save the output of every run in a structure and save this structure in an
% array of structures. A simulation of 10 runs should have 10 output
% structures in the final outputstructure array.

%Run the simulations with the log Gaussian distribution
if strcmp(distribution,'lognormal') == 1
    for run = 1:simulations
      output(run) = run_lognormal(input(run).adapter,input(run).stimulus,input(run).max_adaptation,input(ru
    end
%Run the simulations with the Gaussian distribution
elseif strcmp(distribution,'normal') == 1
    for run = 1:simulations
      output(run) = run_normal(input(run).adapter,input(run).stimulus,input(run).max_adaptation,input(run).
    end
end

% Save the simulated data to a file which is saved in the same folder as
% the MATLAB files
save('simulationoutput','output');
```

B Shifts in Perception Tables

B.1 Baseline Gaussian Curves

Conditions with *number of neurons* set to 50

TABLE B.1: THE RESULTING PEAK SHIFTS FOR 35/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of Gaussian fit model simulations in which the parameter adapter is set to numerosity 35, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

adaptation strength \ standard deviation σ	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.2	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.7	0.7
0.3	0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.5	0.6	0.6	0.7	0.7	0.8	0.8	0.9	0.9	0.9	1.0	1.0	1.1
0.4	0.2	0.3	0.4	0.4	0.5	0.5	0.6	0.7	0.7	0.8	0.8	0.9	1.0	1.0	1.1	1.2	1.2	1.3	1.3	1.4	1.4
0.5	0.3	0.4	0.4	0.5	0.6	0.6	0.7	0.8	0.9	1.0	1.0	1.1	1.2	1.3	1.3	1.4	1.5	1.6	1.7	1.7	1.8
0.6	0.3	0.4	0.5	0.6	0.7	0.8	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2
0.7	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.3	2.4	2.5
0.8	0.4	0.5	0.6	0.7	0.8	0.9	1.1	1.2	1.3	1.4	1.5	1.6	1.8	1.9	2.0	2.1	2.3	2.4	2.5	2.6	2.8
0.9	0.5	0.6	0.7	0.8	0.9	1.0	1.2	1.3	1.4	1.5	1.7	1.8	1.9	2.1	2.2	2.3	2.5	2.6	2.7	2.9	3.0
1	0.5	0.6	0.7	0.9	1.0	1.1	1.2	1.4	1.5	1.6	1.8	1.9	2.1	2.2	2.4	2.5	2.7	2.8	2.9	3.1	3.2

TABLE B.2: THE RESULTING PEAK SHIFTS FOR 45/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of Gaussian fit model simulations in which the parameter adapter is set to numerosity 45, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

adaptation strength \ standard deviation σ	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3
0.2	-0.1	-0.2	-0.2	-0.2	-0.2	-0.3	-0.3	-0.3	-0.4	-0.4	-0.4	-0.5	-0.5	-0.5	-0.5	-0.6	-0.6	-0.6	-0.6	-0.7	-0.7
0.3	-0.2	-0.2	-0.3	-0.3	-0.4	-0.4	-0.5	-0.5	-0.5	-0.6	-0.6	-0.7	-0.7	-0.8	-0.8	-0.9	-0.9	-0.9	-1.0	-1.0	-1.1
0.4	-0.2	-0.3	-0.4	-0.4	-0.5	-0.5	-0.6	-0.7	-0.7	-0.8	-0.8	-0.9	-1.0	-1.0	-1.1	-1.2	-1.2	-1.3	-1.3	-1.4	-1.4
0.5	-0.3	-0.4	-0.4	-0.5	-0.6	-0.6	-0.7	-0.8	-0.9	-1.0	-1.0	-1.1	-1.2	-1.3	-1.3	-1.4	-1.5	-1.6	-1.7	-1.7	-1.8
0.6	-0.3	-0.4	-0.5	-0.6	-0.7	-0.8	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.2
0.7	-0.4	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.3	-2.4	-2.5
0.8	-0.4	-0.5	-0.6	-0.7	-0.8	-0.9	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.8	-1.9	-2.0	-2.1	-2.3	-2.4	-2.5	-2.6	-2.8
0.9	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.2	-1.3	-1.4	-1.5	-1.7	-1.8	-1.9	-2.1	-2.2	-2.3	-2.5	-2.6	-2.7	-2.9	-3.0
1	-0.5	-0.6	-0.7	-0.9	-1.0	-1.1	-1.2	-1.4	-1.5	-1.6	-1.8	-1.9	-2.1	-2.2	-2.4	-2.5	-2.7	-2.8	-2.9	-3.1	-3.2

Conditions with *number of neurons* set to 100

TABLE B.3: THE RESULTING PEAK SHIFTS FOR 35/40/100

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of Gaussian fit model simulations in which the parameter adapter is set to numerosity 35, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
0.2	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.4	0.4	0.5	0.5	0.5	0.5	0.6	0.6	0.6	0.6	0.7	0.7
0.3	0.2	0.2	0.3	0.3	0.4	0.4	0.5	0.5	0.5	0.6	0.6	0.7	0.7	0.8	0.8	0.9	0.9	0.9	1.0	1.0	1.1
0.4	0.2	0.3	0.4	0.4	0.5	0.5	0.6	0.7	0.7	0.8	0.8	0.9	1.0	1.0	1.1	1.2	1.2	1.3	1.3	1.4	1.4
0.5	0.3	0.4	0.4	0.5	0.6	0.6	0.7	0.8	0.9	1.0	1.0	1.1	1.2	1.3	1.3	1.4	1.5	1.6	1.7	1.7	1.8
0.6	0.4	0.4	0.5	0.6	0.7	0.8	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2
0.7	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.3	2.4	2.5
0.8	0.4	0.5	0.6	0.7	0.8	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.8	1.9	2.0	2.1	2.3	2.4	2.5	2.6	2.8
0.9	0.5	0.6	0.7	0.8	0.9	1.0	1.2	1.3	1.4	1.5	1.7	1.8	1.9	2.1	2.2	2.3	2.5	2.6	2.7	2.9	3.0
1	0.5	0.6	0.7	0.9	1.0	1.1	1.2	1.4	1.5	1.7	1.8	1.9	2.1	2.2	2.4	2.5	2.7	2.8	2.9	3.1	3.2

TABLE B.4: THE RESULTING PEAK SHIFTS FOR 45/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of Gaussian fit model simulations in which the parameter adapter is set to numerosity 45, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	2.0	2.1	2.2	2.3	2.4	2.5	2.6	2.7	2.8	2.9	3.0	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	4.0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.2	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3
0.2	-0.1	-0.2	-0.2	-0.2	-0.2	-0.3	-0.3	-0.3	-0.4	-0.4	-0.4	-0.5	-0.5	-0.5	-0.5	-0.6	-0.6	-0.6	-0.6	-0.7	-0.7
0.3	-0.2	-0.2	-0.3	-0.3	-0.4	-0.4	-0.5	-0.5	-0.5	-0.6	-0.6	-0.7	-0.7	-0.8	-0.8	-0.9	-0.9	-0.9	-1.0	-1.0	-1.1
0.4	-0.2	-0.3	-0.4	-0.4	-0.5	-0.5	-0.6	-0.7	-0.7	-0.8	-0.8	-0.9	-1.0	-1.0	-1.1	-1.2	-1.2	-1.3	-1.3	-1.4	-1.4
0.5	-0.3	-0.4	-0.4	-0.5	-0.6	-0.6	-0.7	-0.8	-0.9	-1.0	-1.0	-1.1	-1.2	-1.3	-1.3	-1.4	-1.5	-1.6	-1.7	-1.7	-1.8
0.6	-0.4	-0.4	-0.5	-0.6	-0.7	-0.8	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.2
0.7	-0.4	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.7	-1.8	-1.9	-2.0	-2.1	-2.3	-2.4	-2.5
0.8	-0.4	-0.5	-0.6	-0.7	-0.8	-1.0	-1.1	-1.2	-1.3	-1.4	-1.5	-1.6	-1.8	-1.9	-2.0	-2.1	-2.3	-2.4	-2.5	-2.6	-2.8
0.9	-0.5	-0.6	-0.7	-0.8	-0.9	-1.0	-1.2	-1.3	-1.4	-1.5	-1.7	-1.8	-1.9	-2.1	-2.2	-2.3	-2.5	-2.6	-2.7	-2.9	-3.0
1	-0.5	-0.6	-0.7	-0.9	-1.0	-1.1	-1.2	-1.4	-1.5	-1.7	-1.8	-1.9	-2.1	-2.2	-2.4	-2.5	-2.7	-2.8	-2.9	-3.1	-3.2

B.2 Numerosity Log Gaussian Curves

Conditions with *number of neurons* set to 50

TABLE B.5: THE RESULTING PEAK SHIFTS FOR 20/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 20, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0.1	0.5	1.1	1.6	1.9	1.7	1.3	0.7	0.2	-0.1	-0.3	-0.3	-0.3	-0.3	-0.2
0.2	0	0.1	1.0	2.3	3.4	4.1	3.9	2.9	1.6	0.4	-0.3	-0.7	-0.8	-0.7	-0.6	-0.4
0.3	0	0.2	1.4	3.4	5.3	6.6	6.3	4.9	2.7	0.7	-0.6	-1.3	-1.4	-1.3	-1.0	-0.7
0.4	0	0.2	1.9	4.5	7.2	9.2	9.1	7.2	4.1	1.1	-1.0	-2.0	-2.2	-1.9	-1.5	-1.0
0.5	0	0.3	2.3	5.6	9.2	11.9	12.2	10.1	6.1	1.6	-1.6	-3.1	-3.2	-2.7	-2.1	-1.4
0.6	0	0.3	2.7	6.6	11.1	14.0	15.5	13.4	8.6	2.4	-2.6	-4.6	-4.5	-3.7	-2.7	-1.8
0.7	0	0.4	3.1	7.6	12.9	17.0	18.8	17.0	11.8	3.5	-4.5	-7.0	-6.2	-4.8	-3.5	-2.3
0.8	0	0.4	3.5	8.5	14.6	19.0	22.0	20.8	15.9	5.3	-8.5	-9.9	-8.0	-6.0	-4.2	-2.8
0.9	0	0.4	3.8	9.4	16.1	22.0	24.9	24.5	20.4	8.2	-14.1	-12.1	-9.4	-6.9	-4.9	-3.2
1	0	0.5	4.1	10.1	17.5	24.2	27.4	27.6	24.4	11.6	-16.4	-13.2	-10.1	-7.5	-5.4	-3.6

TABLE B.6: THE RESULTING SHIFTS IN PEAK PERCEPTION FOR 40/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 40, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

max. adaptation \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0	0	0	0	-0.1	-0.4	-0.6	-0.9	-0.9	-0.8	-0.7	-0.5	-0.4	-0.3	-0.2
0.2	0	0	0	0	-0.1	-0.3	-0.9	-1.5	-2.0	-2.0	-1.8	-1.5	-1.1	-0.8	-0.6	-0.3
0.3	0	0	0	0	-0.1	-0.5	-1.7	-2.8	-3.4	-3.4	-3.0	-2.4	-1.8	-1.3	-1.0	-0.6
0.4	0	0	0	0	-0.1	-0.9	-2.8	-4.5	-5.3	-5.2	-4.4	-3.5	-2.6	-1.9	-1.3	-0.8
0.5	0	0	0	0	-0.2	-1.7	-4.7	-7.0	-7.8	-7.3	-6.2	-4.8	-3.5	-2.5	-1.7	-1.1
0.6	0	0	0	0	-0.5	-3.6	-8.2	-10.7	-11.0	-9.9	-8.1	-6.2	-4.5	-3.2	-2.2	-1.3
0.7	-1.5	-2.1	-3.1	-4.1	-6.2	-10.5	-14.0	-15.1	-14.4	-12.5	-10.1	-7.7	-5.6	-3.9	-2.6	-1.6
0.8	-4.2	9.1	-10.6	-13.5	-16.1	-18.2	-19.4	-19.0	-17.4	-14.8	-11.9	-9.1	-6.6	-4.6	-3.1	-1.9
0.9	-5.2	12.4	-13.3	-16.7	-19.6	-21.7	-22.5	-21.6	-19.4	-16.5	-13.3	-10.2	-7.5	-5.2	-3.6	-2.2
1	-5.8	14.3	-14.7	-18.3	-21.4	-23.5	-24.2	-23.1	-20.7	-17.6	-14.2	-11.0	-8.2	-5.8	-4.0	-2.5

TABLE B.7: THE RESULTING SHIFTS IN PEAK PERCEPTION FOR 80/40/50

Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 80, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	-0.1	-0.5	-1.1	-1.6	-1.9	-1.9	-1.7	-1.4	-1.2	-0.8	-0.6	-0.4	-0.3	-0.2	-0.1
0.2	0	-0.1	-1.0	-2.1	-3.2	-3.9	-4.0	-3.5	-3.0	-2.4	-1.8	-1.3	-0.9	-0.6	-0.4	-0.2
0.3	0	-0.2	-1.4	-3.1	-4.8	-5.8	-6.1	-5.5	-4.7	-3.7	-2.8	-2.0	-1.4	-0.9	-0.7	-0.4
0.4	0	-0.2	-1.8	-4.0	-6.2	-7.7	-8.2	-7.6	-6.5	-5.1	-3.8	-2.8	-1.9	-1.3	-0.9	-0.5
0.5	0	-0.3	-2.2	-4.9	-7.6	-9.5	-10.3	-9.6	-8.3	-6.6	-5.0	-3.6	-2.5	-1.7	-1.1	-0.6
0.6	0	-0.3	-2.5	-5.7	-8.8	-11.1	-12.2	-11.5	-10	-8.1	-6.1	-4.4	-3.0	-2.0	-1.4	-0.8
0.7	0	-0.4	-2.9	-6.4	-9.9	-12.6	-13.8	-13.3	-11.6	-9.5	-7.2	-5.3	-3.6	-2.4	-1.6	-1.0
0.8	0	-0.4	-3.2	-7.0	-10.8	-13.8	-15.2	-14.8	-13.1	-10.8	-8.3	-6.1	-4.2	-2.9	-1.9	-1.1
0.9	0	-0.4	-3.5	-7.6	-11.6	-14.8	-16.4	-16.0	-14.3	-11.9	-9.3	-6.9	-4.8	-3.3	-2.2	-1.3
1	0	-0.5	-3.7	-8.1	-12.3	-15.7	-17.4	-17.0	-15.3	-12.9	-10.2	-7.6	-5.4	-3.7	-2.4	-1.5

Conditions with *number of neurons* set to 100

TABLE B.8: THE RESULTING SHIFTS IN PEAK PERCEPTION FOR 20/40/100
Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 20, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0.1	0.5	1.1	1.7	1.9	10.8	1.2	0.7	0.2	-0.2	-0.3	-0.4	-0.3	-0.3	-0.2
0.2	0	0.1	1.0	2.3	3.5	4.1	30.8	20.8	1.5	0.4	-0.4	-0.7	-00.8	-0.7	-0.6	-0.4
0.3	0	0.2	1.4	3.4	5.4	6.5	6.2	4.6	2.5	0.6	-0.7	-1.3	-1.4	-1.2	-1.0	-0.7
0.4	0	0.2	1.9	4.5	7.3	9.1	9.0	6.9	3.9	0.9	-1.1	-2.0	-2.2	-10.8	-1.4	-1.0
0.5	0	0.3	2.3	5.6	9.2	110.8	12.0	9.7	5.7	1.3	-10.8	-3.1	-3.2	-2.6	-2.0	-1.3
0.6	0	0.3	2.7	6.6	11.1	14.5	15.2	12.9	8.1	1.9	-2.9	-4.6	-4.5	-3.6	-2.6	-1.8
0.7	0	0.4	3.1	7.6	12.9	17.2	18.5	16.4	11.2	2.8	-4.9	-6.9	-6.1	-4.7	-3.3	-2.2
0.8	0	0.4	3.5	8.5	14.6	19.6	21.6	20.2	15.2	4.3	-8.9	-9.7	-7.9	-5.8	-4.1	-2.7
0.9	0	0.4	3.8	9.4	16.2	21.9	24.5	23.9	19.6	6.5	-14.1	-11.9	-9.2	-6.7	-4.7	-3.1
1	0	0.5	4.1	10.1	17.6	23.9	27.0	26.9	23.7	9.0	-16.2	-12.9	-9.9	-7.3	-5.2	-3.5

TABLE B.9: THE RESULTING SHIFTS IN PEAK PERCEPTION FOR 40/40/100
Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 40, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

max. adaptation \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.0	0.0	0.0	0.0	0.0	-0.1	-0.4	-0.7	-0.8	-0.9	-0.8	-0.6	-0.5	-0.4	-0.3	-0.2
0.2	0.0	0.0	0.0	0.0	0.0	-0.3	-0.9	-1.6	-1.9	-2.0	-1.8	-1.4	-1.1	-0.8	-0.5	-0.3
0.3	0.0	0.0	0.0	0.0	0.0	-0.6	-1.7	-2.8	-3.4	-3.3	-3.0	-2.3	-1.8	-1.3	-0.9	-0.6
0.4	0.0	0.0	0.0	0.0	-0.1	-1.0	-2.9	-4.6	-5.2	-5.1	-4.4	-3.4	-2.6	-1.8	-1.2	-0.8
0.5	0.0	0.0	0.0	0.0	-0.2	-1.9	-4.9	-7.1	-7.7	-7.2	-6.1	-4.7	-3.5	-2.4	-1.6	-1.0
0.6	0.0	0.0	0.0	0.0	-0.5	-4.0	-8.4	-10.7	-10.8	-9.7	-7.9	-6.0	-4.4	-3.0	-2.0	-1.3
0.7	-1.1	2.2	-3.1	-4.1	-6.3	-10.8	-14.0	-15.1	-14.2	-12.2	-9.9	-7.5	-5.4	-3.7	-2.5	-1.6
0.8	-3.9	9.1	-10.6	-13.5	-16.0	-18.1	-19.2	-18.8	-17.0	-14.5	-11.6	-8.8	-6.4	-4.4	-2.9	-1.8
0.9	-5.0	12.4	-13.3	-16.7	-19.5	-21.6	-22.3	-21.3	-19.1	-16.1	-13.0	-9.9	-7.3	-5.1	-3.4	-2.1
1	-5.7	14.3	-14.7	-18.3	-21.3	-23.5	-24.0	-22.8	-20.3	-17.2	-13.9	-10.7	-8.0	-5.6	-3.8	-2.4

TABLE B.10: THE RESULTING SHIFTS IN PEAK PERCEPTION FOR 80/40/100
Per combination of adaptation strength and σ the resulting peak shift in perception is shown. The shifts are shown for the set of log Gaussian fit model simulations in which the parameter adapter is set to numerosity 80, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

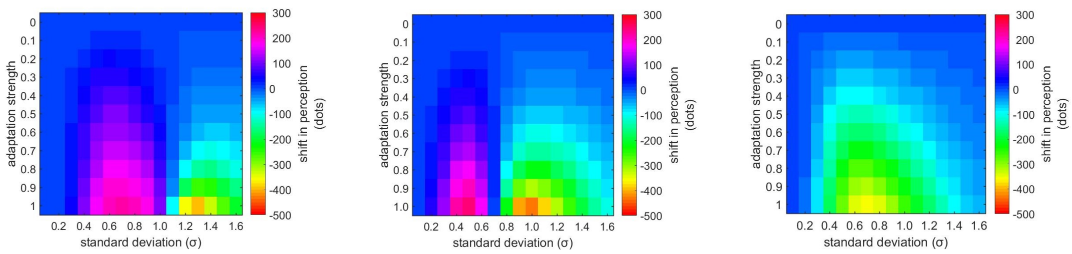
adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	-0.1	-0.5	-1.1	-1.5	-1.9	-1.8	-1.7	-1.3	-1.1	-0.8	-0.6	-0.4	-0.3	-0.2	-0.1
0.2	0	-0.1	-1.0	-2.1	-3.1	-3.8	-3.8	-3.5	-2.9	-2.3	-1.7	-1.2	-0.9	-0.6	-0.4	-0.2
0.3	0	-0.2	-1.4	-3.1	-4.7	-5.8	-5.9	-5.4	-4.5	-3.6	-2.7	-1.9	-1.4	-0.9	-0.6	-0.4
0.4	0	-0.2	-1.8	-4.0	-6.1	-7.7	-8.0	-7.4	-6.2	-4.9	-3.7	-2.6	-1.9	-1.2	-0.8	-0.5
0.5	0	-0.3	-2.2	-4.9	-7.5	-9.4	-10	-9.4	-8.0	-6.4	-4.8	-3.4	-2.4	-1.6	-1.0	-0.6
0.6	0	-0.3	-2.5	-5.7	-8.7	-11.1	-11.9	-11.3	-9.7	-7.8	-5.9	-4.2	-3.0	-2.0	-1.3	-0.8
0.7	0	-0.4	-2.9	-6.4	-9.8	-12.5	-13.5	-13.0	-11.3	-9.1	-7.0	-5.0	-3.5	-2.3	-1.5	-0.9
0.8	0	-0.4	-3.2	-7.0	-10.7	-13.7	-14.9	-14.5	-12.7	-10.4	-8.0	-5.8	-4.1	-2.7	-1.8	-1.1
0.9	0	-0.4	-3.5	-7.6	-11.5	-14.7	-16.1	-15.7	-13.9	-11.5	-9.0	-6.6	-4.7	-3.1	-2.0	-1.2
1	0	-0.5	-3.7	-8.1	-12.2	-15.6	-17.1	-16.7	-14.9	-12.5	-9.9	-7.3	-5.2	-3.5	-2.3	-1.4

C COG Shift in Perception

The center of gravity (COG) relates to the middle point of the area under the population excitation curve. Because of this the COG changes when the shape of the curve changes, also when the peak does remain the same. Adaptation can have an effect on the COG just as it has an effect on the peak of the population excitation curve. This is why a COG shift in perception can be calculated as well. However, in this model the magnitude of the COG shifts were too large to be a meaningful measure.

Congruent with the peak shift in perception, the *number of neurons* does not influence the shift in perception much. However, comparing the result tables for the peak shift in perception (tables B.5 to B.10) and for the COG shift in perception (tables C.1 to C.6) there is a big difference between both: the COG shifts in perception are of a much larger magnitude than the peak shifts in perception. The large magnitude of the COG shifts in perception become more recognizable when looking at figure C.1 and the underlying tables of the color maps (tables C.1 to C.6). Considering the color maps a shift in perception of almost -300 dots is measured. This cannot be a possible psychophysical reaction as the shift would mean that the second stimulus is perceived as negative, less than 0 dots.

Overall, the magnitudes of the shifts in perception found do not in any way match the earlier found psychophysical data (Aagten-Murphy & Burr, 2016; Tsouli et al., 2018). Because of this, the shifts in perception are not included in the results.



(a) Adapter 20 and 50 neurons (b) Adapter 40 and 50 neurons (c) Adapter 80 and 50 neurons

FIGURE C.1: COLOR MAPS INDICATING SHIFTS IN PERCEPTION FOR STIMULUS 40
The figures show the shifts in perception according to the COG found for the three conditions with 50 neurons. The shifts are shown per combination of adaptation strength and σ value. The underlying values of the color maps can be found in tables C.1, C.2 and C.3.

Conditions with *number of neurons* set to 50

TABLE C.1: THE RESULTING SHIFTS IN COG PERCEPTION FOR 20/40/50
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 20, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

standard deviation σ / adaptation strength	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0.69	4.55	10.6	16.13	18.33	16.96	13.21	8.51	3.96	0.23	-2.46	-4.13	-4.96	-5.17	-4.96
0.2	0	1.36	9.06	21.37	32.87	37.79	35.45	28.08	18.39	8.66	0.42	-5.63	-9.4	-11.22	-11.59	-11.01
0.3	0	2.01	13.49	32.23	50.12	58.29	55.53	44.83	29.94	14.29	0.51	-9.84	-16.31	-19.33	-19.76	-18.52
0.4	0	2.65	17.83	43.1	67.74	79.71	77.19	63.67	43.55	21.15	0.42	-15.65	-25.71	-30.16	-30.4	-28.05
0.5	0	3.26	22.05	53.88	85.53	101.82	100.32	84.77	59.67	29.61	-0.08	-24.06	-39.07	-45.18	-44.69	-40.4
0.6	0	3.86	26.13	64.44	103.24	124.29	124.63	108.15	78.78	40.22	-1.42	-37	-59.15	-66.97	-64.55	-56.79
0.7	0	4.44	30.04	74.66	120.56	146.64	149.61	133.58	101.29	53.61	-4.65	-58.63	-91.57	-100.31	-93.15	-79.11
0.8	0	4.99	33.76	84.37	137.12	168.24	174.4	160.26	127.12	70.33	-12.53	-98.85	-148.4	-153.87	-135.49	-110.06
0.9	0	5.53	37.25	93.42	152.48	188.28	197.74	186.49	154.85	89.62	-33.04	-183.16	-253.11	-239.21	-197.05	-152.6
1	0	6.05	40.48	101.63	166.16	205.79	217.84	209.2	179.66	104.71	-89	-350.48	-395.98	-335.75	-269.15	-205.74

TABLE C.2: THE RESULTING SHIFTS IN COG PERCEPTION FOR 40/40/50
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 40, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

standard deviation σ / adaptation strength	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.47	2.04	5.14	9.45	10.71	7.32	2.11	-2.55	-5.77	-7.56	-8.21	-8.08	-7.48	-6.64	-5.72	-4.80
0.2	1.04	4.48	11.29	20.67	23.37	15.97	4.50	-5.87	-13.11	-17.09	-18.46	-18.04	-16.56	-14.56	-12.40	-10.31
0.3	1.73	7.43	18.68	34.10	38.45	26.27	7.22	-10.29	-22.66	-29.40	-31.55	-30.56	-27.75	-24.11	-20.29	-16.67
0.4	2.58	11.02	27.62	50.23	56.51	38.66	10.29	-16.38	-35.46	-45.75	-48.68	-46.61	-41.76	-35.78	-29.69	-24.07
0.5	3.61	15.40	38.48	69.64	78.20	53.63	13.70	-25.02	-53.25	-68.18	-71.70	-67.62	-59.60	-50.21	-41.00	-32.75
0.6	4.87	20.74	51.63	92.92	104.14	71.73	17.33	-37.79	-78.97	-100.08	-103.55	-95.76	-82.68	-68.28	-54.74	-43.00
0.7	6.41	27.18	67.33	120.41	134.74	93.42	20.80	-57.48	-117.90	-147.20	-148.78	-134.06	-112.90	-91.12	-71.55	-55.16
0.8	8.22	34.69	85.50	151.78	169.61	118.55	23.04	-89.17	-179.31	-218.42	-213.29	-186.04	-152.37	-120.02	-92.18	-69.67
0.9	10.22	42.89	105.13	185.17	206.50	145.26	21.47	-141.20	-276.14	-321.40	-299.07	-252.32	-201.97	-155.92	-117.36	-86.97
1	12.14	50.65	123.55	216.00	239.81	167.97	10.68	-220.18	-402.55	-429.32	-379.68	-317.98	-256.16	-197.74	-147.35	-107.45

TABLE C.3: THE RESULTING SHIFTS IN COG PERCEPTION FOR 80/40/50
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 80, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 50 neurons.

standard deviation σ / adaptation strength	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	-1.63	-8.76	-15.36	-18.46	-18.93	-17.9	-16.11	-14.04	-11.96	-10.02	-8.28	-6.76	-5.46	-4.37	-3.47
0.2	0	-3.21	-17.65	-31.98	-39.39	-40.96	-38.98	-35.14	-30.56	-25.93	-21.59	-17.72	-14.36	-11.53	-9.17	-7.23
0.3	0	-4.73	-26.59	-49.85	-63.07	-66.66	-63.93	-57.73	-50.11	-42.32	-35.02	-28.53	-22.96	-18.29	-14.44	-11.31
0.4	0	-6.21	-35.5	-68.86	-89.74	-96.64	-93.53	-84.68	-73.37	-61.66	-50.68	-40.98	-32.71	-25.84	-20.24	-15.74
0.5	0	-7.62	-44.26	-88.83	-119.49	-131.42	-128.59	-116.85	-101.09	-84.54	-69.01	-55.34	-43.79	-34.3	-26.65	-20.56
0.6	0	-8.98	-52.76	-109.39	-152.11	-171.27	-169.75	-154.99	-134	-111.57	-90.46	-71.95	-56.43	-43.8	-33.74	-25.82
0.7	0	-10.27	-60.83	-129.97	-186.89	-215.74	-216.92	-199.35	-172.53	-143.25	-115.47	-91.13	-70.84	-54.49	-41.59	-31.57
0.8	0	-11.49	-68.33	-149.76	-222.25	-262.94	-268.4	-248.77	-216.2	-179.58	-144.28	-113.16	-87.25	-66.49	-50.29	-37.84
0.9	0	-12.64	-75.06	-167.67	-255.47	-308.55	-319.22	-299.02	-262.36	-219.36	-176.6	-138.11	-105.81	-79.97	-59.94	-44.71
1	0	-13.73	-80.84	-182.37	-282.52	-345.01	-359.6	-340.92	-304.45	-259.06	-211	-165.65	-126.57	-95.02	-70.62	-52.22

Conditions with *number of neurons* set to 100

TABLE C.4: THE RESULTING SHIFTS IN COG PERCEPTION FOR 20/40/100
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 20, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	0.69	4.55	10.59	16.04	18.14	16.67	12.87	8.17	3.67	0	-2.62	-4.22	-5	-5.17	-4.94
0.2	0	1.36	9.06	21.34	32.69	37.4	34.88	27.38	17.66	8.01	-0.1	-5.98	-9.6	-11.3	-11.59	-10.95
0.3	0	2.01	13.49	32.18	49.85	57.72	54.68	43.77	28.8	13.22	-0.37	-10.45	-16.66	-19.46	-19.73	-18.41
0.4	0	2.65	17.83	43.04	67.38	78.96	76.08	62.26	41.96	19.57	-0.93	-16.61	-26.24	-30.32	-30.32	-27.84
0.5	0	3.26	22.05	53.8	85.08	100.92	98.98	83.03	57.61	27.42	-2.07	-25.5	-39.84	-45.36	-44.5	-40.02
0.6	0	3.86	26.13	64.35	102.71	123.25	123.11	106.16	76.26	37.27	-4.32	-39.17	-60.22	-67.11	-64.12	-56.13
0.7	0	4.44	30.04	74.55	119.96	145.48	147.96	131.41	98.35	49.73	-8.94	-61.95	-93.01	-100.19	-92.23	-77.96
0.8	0	4.99	33.76	84.24	136.45	166.98	172.69	158.03	123.91	65.31	-19.16	-104.17	-150.09	-152.95	-133.59	-108.07
0.9	0	5.53	37.24	93.28	151.75	186.94	196	184.34	151.55	83.18	-44.13	-192.03	-253.89	-236.18	-193.35	-149.29
1	0	6.05	40.48	101.48	165.37	204.37	216.1	207.16	176.39	96.27	-109.1	-361.67	-391.82	-330.04	-263.75	-200.9

TABLE C.5: THE RESULTING SHIFTS IN COG PERCEPTION FOR 40/40/100
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 40, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0.48	2.04	5.14	9.38	10.42	6.87	1.65	-2.91	-6.00	-7.68	-8.24	-8.06	-7.43	-6.58	-5.65	-4.73
0.2	1.05	4.48	11.28	20.53	22.75	14.97	3.49	-6.67	-13.63	-17.35	-18.53	-17.98	-16.43	-14.40	-12.24	-10.15
0.3	1.74	7.43	18.67	33.86	37.43	24.64	5.52	-11.66	-23.54	-29.83	-31.64	-30.43	-27.50	-23.81	-20.00	-16.40
0.4	2.58	11.02	27.62	49.88	55.03	36.24	7.73	-18.48	-36.83	-46.38	-48.75	-46.33	-41.31	-35.28	-29.22	-23.66
0.5	3.61	15.40	38.47	69.16	76.16	50.28	10.03	-28.12	-55.25	-69.04	-71.68	-67.10	-58.85	-49.42	-40.28	-32.14
0.6	4.87	20.74	51.62	92.28	101.46	67.26	12.22	-42.27	-81.85	-101.17	-103.31	-94.79	-81.45	-67.07	-53.68	-42.12
0.7	6.40	27.18	67.32	119.58	131.34	87.61	13.74	-64.00	-122.02	-148.43	-147.98	-132.31	-110.91	-89.29	-70.01	-53.94
0.8	8.20	34.69	85.48	150.75	165.43	111.18	13.33	-98.82	-185.12	-219.36	-211.27	-182.95	-149.24	-117.29	-89.98	-67.98
0.9	10.17	42.89	105.10	183.92	201.54	136.18	8.13	-155.65	-283.75	-320.74	-294.78	-247.30	-197.29	-151.99	-114.29	-84.68
1	12.05	50.65	123.52	214.54	234.12	157.08	-7.29	-240.73	-409.51	-424.80	-373.11	-311.62	-250.18	-192.53	-143.22	-104.41

TABLE C.6: THE RESULTING SHIFTS IN COG PERCEPTION FOR 80/40/100
Per combination of adaptation strength and σ the resulting COG shift in perception is shown. The shifts are shown for the set of simulations in which the parameter adapter is set to numerosity 80, the parameter stimulus is set to numerosity 40 and the parameter number of neurons is set to 100 neurons.

adaptation strength \ standard deviation σ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0.1	0	-1.63	-8.76	-15.38	-18.49	-18.9	-17.78	-15.94	-13.85	-11.77	-9.84	-8.12	-6.63	-5.35	-4.28	-3.4
0.2	0	-3.21	-17.65	-32.01	-39.43	-40.87	-38.7	-34.73	-30.11	-25.48	-21.18	-17.36	-14.07	-11.29	-8.97	-7.07
0.3	0	-4.73	-26.59	-49.89	-63.13	-66.47	-63.41	-57.01	-49.31	-41.54	-34.31	-27.93	-22.46	-17.88	-14.11	-11.05
0.4	0	-6.21	-35.5	-68.93	-89.79	-96.29	-92.68	-83.52	-72.1	-60.43	-49.59	-40.05	-31.95	-25.24	-19.77	-15.37
0.5	0	-7.62	-44.26	-88.91	-119.52	-130.84	-127.29	-115.08	-99.18	-82.72	-67.41	-54.01	-42.72	-33.46	-26	-20.06
0.6	0	-8.98	-52.76	-109.48	-152.08	-170.36	-167.8	-152.41	-131.25	-109	-88.22	-70.1	-54.96	-42.67	-32.87	-25.17
0.7	0	-10.27	-60.83	-130.06	-186.75	-214.35	-214.13	-195.73	-168.73	-139.71	-112.42	-88.64	-68.89	-52.99	-40.47	-30.73
0.8	0	-11.49	-68.33	-149.85	-221.94	-260.91	-264.55	-243.91	-211.15	-174.88	-140.25	-109.89	-84.71	-64.58	-48.87	-36.8
0.9	0	-12.64	-75.06	-167.75	-254.91	-305.74	-314.24	-292.97	-256.09	-213.47	-171.47	-133.94	-102.57	-77.55	-58.17	-43.43
1	0	-13.73	-80.84	-182.44	-281.64	-341.38	-353.79	-334.26	-297.55	-252.31	-204.87	-160.51	-122.54	-92.02	-68.44	-50.66

Appendix References

Aagten-Murphy, D., & Burr, D. (2016). Adaptation to numerosity requires only brief exposures, and is determined by number of events, nor exposure duration. Journal of Vision, 16(10), 1–14. doi: 10.1167/16.10.22

Tsouli, A., Dumoulin, S., te Pas, S., & van der Smagt, M. (2018). Adaptation reveals unbalanced interaction between numerosity and time. Cortex. doi: 10.1016/j.cortex.2018.02.013