Published in Image Processing On Line on 2016–11–18.
Submitted on 2016–04–27, accepted on 2016–11–03.
ISSN 2105–1232 © 2016 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2016.177

Generation and Detection of Alignments in Gabor Patterns

Samy Blusseau, Rafael Grompone von Gioi

CMLA, ENS Cachan, France
({samy.blusseau, grompone}@cmla.ens-cachan.fr)

Communicated by Gregory Randall *Demo edited by* Samy Blusseau

Abstract

This paper presents a method to be used in psychophysical experiments to compare directly visual perception to an a contrario algorithm, on a straight patterns detection task. The method is composed of two parts. The first part consists in building a stimulus, namely an array of oriented elements, in which an alignment is present with variable salience. The second part focuses on a detection algorithm, based on the a contrario theory, which is designed to predict which alignment will be considered as the most salient in a given stimulus.

Source Code

The source code associated to this manuscript (written in Matlab) and an online demo are available from the web page of this article[1].

Keywords: human and computer vision; a contrario detection

1 Introduction

Since the emergence of the field of Computer Vision [18] about fifty years ago there have been many attempts at formalizing vision theories and especially the one formulated by the Gestalt school of psychology. Its members, Wertheimer, Köhler, Koffka, Kanizsa among others [28, 11, 5, 19, 10], developed from the twenties to the eighties an original modus operandi, leading to the conclusion that the first steps of visual perception are based on a reduced set of geometrical grouping laws [26, 27]. These laws, also called *Gestalts*, describe the configurations in which most human observers can't help interpreting different elements as one shape or group. Figure 1 illustrates the laws of proximity, symmetry and good continuation.

Desolneux et al. [4] designed the *a contrario* theory as one of the attempts to provide mathematical foundations to these laws. Several properties of the *a contrario* framework argue in favor of its suitability for visual perception modeling.

[1] https://doi.org/10.5201/ipol.2016.177

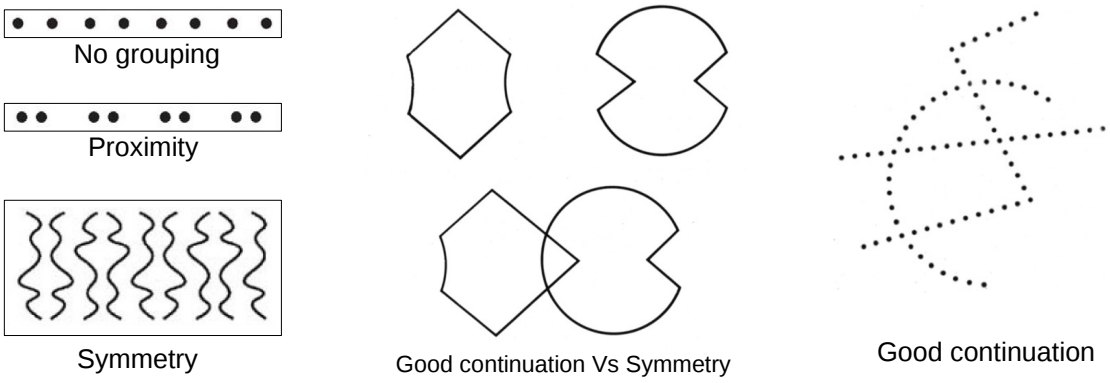


Figure 1: Illustration of some of the grouping laws defined by the Gestalt school of psychology. The left hand image, adapted from [21], shows how proximity and symmetry forces us to group dots and lines two by two. The right hand drawing is a figure from [9], which we interpret as three smooth lines because of the good continuation grouping law. The central image, adapted from [10], illustrates relationships between grouping laws when several of them occur simultaneously. In this case, in spite of the two symmetrical shapes, we can't help seeing two asymmetrical shapes with more regular contours.

First, it applies Attneave's principle, stating that we do not perceive any structure in white noise [1]. Computer vision algorithms based on the *a contrario* theory, like for example the Line Segment Detector [7, 8], typically avoid detections in noise. This idea is also consistent with the more general *non-accidentalness* principle, introduced by Witkin, Tenenbaum and Lowe [14, 29, 30], and of great importance in the study of perception. This principle states that spatial relations are perceptually relevant when their accidental occurrence is unlikely. Therefore, in accordance with Attneave's statement, these meaningful spatial relations should not appear by chance in a random image of noise.

Another strength of the *a contrario* theory is that it allows the design of parameterless detection algorithms. These make as few assumptions as needed to adapt *automatically* to a large variety of inputs. Similarly, our visual system is able to perform equally well in very different conditions, without the need to re-learn, each time, priors on the world.

To link the Gestaltist qualitative description of perceptual grouping laws to a mathematical model, a quantitative gap must be bridged. Human perceptual behavior has been the subject of quantitative experimentation since the times of Fechner, the founder of Psychophysics. This relatively new science investigates the relationship between the stimulus intensity and the perceived sensation [24]. Concerning visual perception, psychophysical studies have added quantitative measurements to the qualitative observations made by the Gestaltists. Among them, one finds experiments investigating the so called contour[2] integration phenomenon [6]. In particular, the role of the Gestaltic *good continuation* grouping law (Figure 1), which can be rephrased in Palmer's words: "All else being equal, elements that can be seen as smooth continuations of each other tend to be grouped together" [22, p. 259].

In this paper, we build on a method introduced in the early 90s [6, 12] and still used in the study of good continuation [13, 20, 15, 16, 25, 23]. It consists in embedding discrete contours in a cluttered background of oriented elements, called Gabor patches, and measuring the ability of human observers to perceive such contours, as a function of their characteristics. Here, we focus on a particular case of good continuation, namely alignments of Gabor patches, like the left hand image in Figure 2.

[2]The word contour is used differently in computer vision and human vision science. It is commonly used in the psychophysics literature to refer to linear structures, tested as targets for visual grouping.

A result that is common to the previously mentioned experiments, and illustrated in Figure 2, is that the visual system is good at detecting smooth paths formed by elements that are roughly oriented like the local tangent of the contour. These studies also reported decreasing detection performance when contours were deviating from this ideal configuration. An important step towards the understanding of the mechanisms underlying these observations is the definition of a model that could *explain* and *predict* these results obtained experimentally. This paper describes a methodology adapted to address that issue.

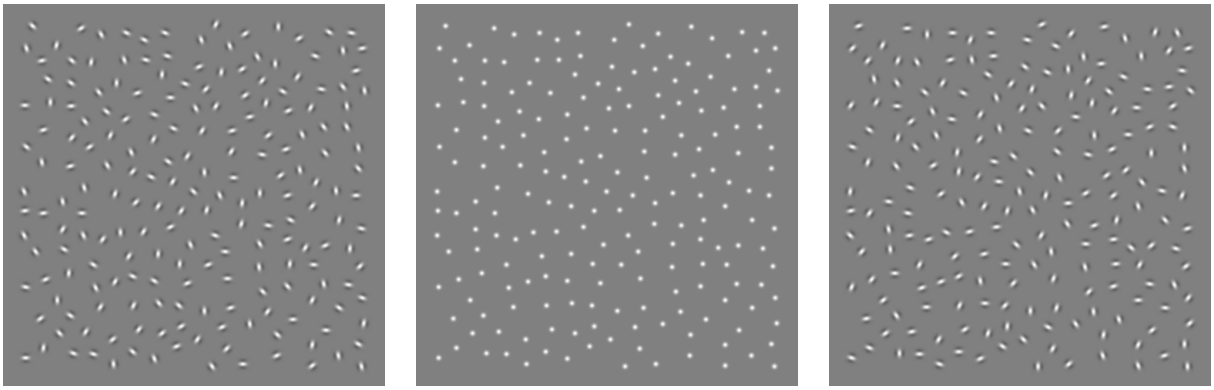


Figure 2: Illustration of the perception of alignments in arrays of Gabor patches. The displayed elements have the same coordinates in all three images. In the left hand image, 10 elements are aligned and have the same orientation as their line, whereas the other elements are randomly oriented. In the right hand image, all elements have random orientations, even the 10 elements that appear aligned on the left hand image. The central image only shows the coordinates of the elements, without any orientation. An alignment is perceptible only in the left hand image, where the aligned elements have the same orientation as their line.

In the first part, we present an algorithm to generate stimuli suited to test the perception of alignments in different conditions. This generation algorithm complies with several requirements. As in Figure 2, in a generated stimulus an alignment should be perceptible only thanks to the orientations of the elements composing it. It should be almost impossible to see without the orientation information, like in the central image of Figure 2, or when the orientations of the aligned elements are unconstrained, like in the right hand image. Furthermore, the algorithm should allow a great variety of stimuli. In particular, we want to test the effect observed in Figure 2 with different quantities of elements per image, several numbers of aligned elements, and in intermediate orientation conditions, that is, ranging from totally constrained orientations for the aligned elements (left hand image) to completely random orientations (right hand image).

In the second part, we describe a detection algorithm based on the *a contrario* theory, designed to detect alignments in the latter stimuli. More precisely, this algorithm relies on the non-accidentalness principle to predict whether a candidate alignment is perceptually meaningful. When the target alignment is detectable for human observers, the algorithm should detect it as well. Otherwise, it should prefer another structure in the background.

2 Generation Algorithm

The algorithm first sets the coordinates of the elements in the image, and then assigns them their orientations. Whereas this last step is straightforward (see Section 2.5), the first one is more technical and is detailed in Sections 2.1- 2.4. The idea is to place randomly, in a square image, a number N of elements, n of which are aligned, and so that the image is filled homogeneously, without clusters and

large empty regions. In short, we do so by first placing n elements on a line and regularly spaced by a distance d_a, and then filling the background with $N-n$ random elements respecting a minimal distance d_b from each other and from the aligned elements.

2.1 Distances between Points

We first need to set the minimal distance between two elements, which we will also take as the minimal distance from an element to an edge of the image. Recall that we want N elements to fit into the image and fill it homogeneously, avoiding clusters and large empty regions. These requirements are somehow the converse to those of the well known problem consisting in filling a domain with as many spheres as possible, given their radius r. Indeed, in our case we know the number of discs to fit into the square image, and we want to set their radius in order to get the most homogeneous layout. This amounts to placing N discs with equal radius r in a square, without overlap but in a compact way, so that the distance between the centers of two neighboring discs, is about $2r$. The most compact configuration fulfilling these requirements consists in a hexagonal lattice. As we see in Figure 3, in such an arrangement the density is $\frac{1}{2\sqrt{3}r^2}$ disc per surface unit. Therefore, the maximal possible radius $r_{hex}(N,S)$ for N discs to fit in a domain of area S verifies

$$r_{hex}^2(N,S) = \frac{S}{2\sqrt{3}N}. \tag{1}$$

The distance $d_{hex}(N,S)$ between two adjacent disc centers is twice this radius. In our case we want to fill a square image of size $I \times I$, with the additional constraint that a disc center be also d_{hex} pixels away from the image's edge. Thus the domain is a square with side $I - 4r_{hex} = I - 2d_{hex}$. Replacing S by $(I - 2d_{hex}(N,I))^2$ in Equation (1), we get the following equation for d_{hex}

$$\begin{aligned} d_{hex}(N,I) &= \frac{2\left(I-2d_{hex}(N,I)\right)}{\sqrt{2\sqrt{3}N}} \\ &= \frac{2I}{\sqrt{2\sqrt{3}N+4}}. \end{aligned} \tag{2}$$

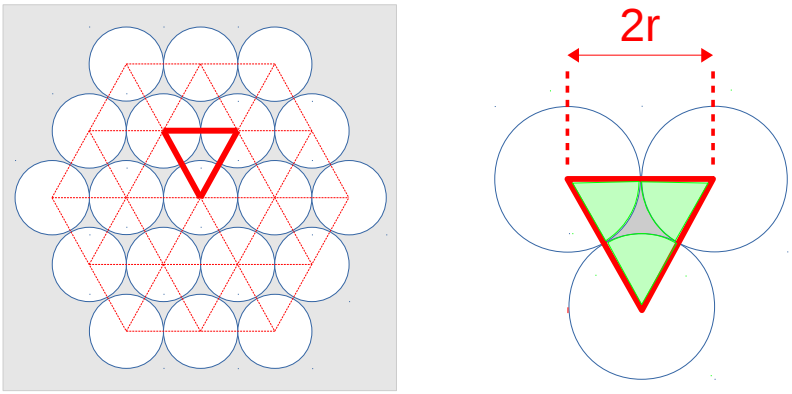


Figure 3: Illustration of the density of a hexagonal arrangement of discs with radius r. The area of a red equilateral triangle is $\sqrt{3}r^2$, and a triangle contains half a disc (each green portion represents one sixth of a disc, and a triangle contains three of these portions, that is $3 \times \frac{1}{6} = \frac{1}{2}$ disc). Thus, there is $\frac{1}{2\sqrt{3}r^2}$ disc per unit area.

In order to achieve a random layout for the elements' coordinates (and not a regular lattice) in our stimuli we need a smaller value for the minimal distance d_b between background points. This is

equivalent to throw discs with smaller radius so that they can fit more loosely in the same square. Therefore, we set[3]

$$\begin{aligned} d_b(N, I) &= \frac{\lambda(I-2d_b(N,I))}{\sqrt{2\sqrt{3}N}} \\ &= \frac{\lambda I}{\sqrt{2\sqrt{3}N}+2\lambda}, \end{aligned} \tag{3}$$

with $\lambda < 2$. For our stimuli, we choose $\lambda = 1.5$ and $I = 500$ pixels. Regarding the distance d_a separating regularly the aligned points, we simply take

$$d_a(N, I) = d_{hex}(N, I) = \frac{2I}{\sqrt{2\sqrt{3}N + 4}}. \tag{4}$$

Indeed, whereas d_b is a *minimal* distance between random points, which is never achieved in practice in the stimulus generation, d_a is the *exact* spacing between aligned elements; thus it is necessary to choose $d_a > d_b$, otherwise the aligned points would always be closer from each other than the background elements, and would therefore be easily detected by an observer from the points coordinates only.

2.2 Placing the Alignment

In this section we describe how we build an alignment of n points, regularly spaced by a distance d_a, in a square image $\mathcal{I}$ of size $I \times I$ pixels, deprived of d_b pixels wide horizontal and vertical margins (Figure 4a). First, assume we have already chosen the center of the alignment $c = (x_c, y_c)$ (that is to say the center of the segment defined by the two extremity points in the alignment), and its orientation $\theta_a \in [0, \pi)$. Then, noting $\overrightarrow{v_a} = (cos(\theta_a), sin(\theta_a))$ a unitary directing vector of this segment, the target alignment is the set points $s_1, s_2, \ldots, s_n$ defined by

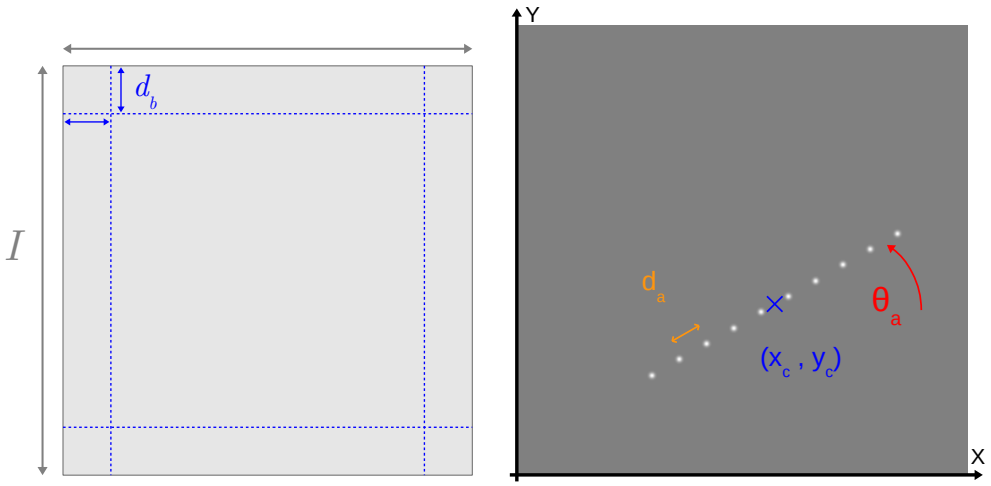
$$\begin{cases} s_1 = c - \frac{(n-1)\times d_a}{2}\overrightarrow{v_a} \\ s_k = s_1 + (k-1)d_a\overrightarrow{v_a} \quad if\ 2 \leq k \leq n. \end{cases} \tag{5}$$

The center c and the orientation θ_a must be chosen so that the n aligned points belong to the allowed image domain, that is, in the image but not in the margins. To ensure this, it is sufficient to test if the two extremity points s_1 and s_n fall in the image. This criterion is formalized by Equation (6).

$$\begin{cases} d_b + \frac{1}{2}(n-1)d_a\,|cos(\theta_a)| \leq x_c \leq I - d_b - \frac{1}{2}(n-1)d_a\,|cos(\theta_a)| \\ d_b + \frac{1}{2}(n-1)d_a\,|sin(\theta_a)| \leq y_c \leq I - d_b - \frac{1}{2}(n-1)d_a\,|sin(\theta_a)|. \end{cases} \tag{6}$$

Thus, in our algorithm, for a given number n of aligned elements, we first choose randomly the orientation θ_a, and then choose randomly the center c from the rectangle defined by Equation (6), if the latter is not empty.

[3]It may seem more intuitive to simply relax the minimal distance by setting $d_b = \lambda d_{hex} = \lambda\frac{2I}{\sqrt{2\sqrt{3}N+4}}$ with $\lambda < 1$ instead of the result of Equation (3) with $\lambda < 2$, but recall that we want the minimal distance between elements to be also the minimal distance from an element to the edge of the image. When relaxing the minimal distance between elements, we also relax the size of the square. Therefore the relaxation needs to be applied to the first line of Equation (2), and not directly to its solution d_{hex}. This is what is done in Equation (3). For large values of N, $d_b = \frac{\lambda_1 I}{\sqrt{2\sqrt{3}N+2\lambda_1}}$ and $d_b = \lambda_2 d_{hex} = \lambda_2\frac{2I}{\sqrt{2\sqrt{3}N+4}}$ give *approximately* the same d_b if $\lambda_1 = 2\lambda_2$.



(a) (b)

Figure 4: (a) The authorized image region for the elements of the stimulus: a $I \times I$ pixels square, deprived of d_b pixels wide horizontal and vertical margins. (b) A discrete homogeneous line-segment, composed of $n = 10$ points, with spacing d_a, center (x_c, y_c), and direction θ_a. In the implementation of the algorithm, we will take the left hand lower corner as the origin of the coordinates.

The algorithm also ensures that this rectangle is not empty by limiting the aligned elements to a maximum number $n_{\max}$. This quantity is defined by Equation (7) as the maximum number of elements than can fit in the allowed image domain, arranged into a central and horizontal (or, equivalently, vertical) alignment with d_a pixels as regular spacing. This constraint guarantees that an alignment composed of no more than $n_{\max}$ elements, can always fit in the center of the image, whatever its orientation θ_a.

$$n_{\max} = \left\lfloor \frac{I - 2d_b}{d_a} \right\rfloor + 1. \tag{7}$$

2.3 Placing the Background Elements

Recall that we want to fill the background of $\mathcal{I}$ with $N - n$ random elements respecting a minimal distance d_b from each other and from the aligned elements. The principle of the algorithm consists in iteratively performing two main steps. The first step is to define a "forbidden region" as the union of discs with radius r_b and centered at the already placed points. The second step is to choose a new point randomly and uniformly from the complement of the forbidden region. In practice, the forbidden region is defined as a discrete dilation of the set of already placed points, with an approximation of a disc of radius r_b as structural element. To set the problem in a discrete frame, we use a binary image $\mathcal{I}'$ of size $I' \times I'$ pixels, in which all points will have integer coordinates. Image $\mathcal{I}'$ will have a different size from $\mathcal{I}$, as will be explained in the next paragraph. The placement of the background elements is performed in this new image $\mathcal{I}'$, and then the coordinates of these new points are mapped to $\mathcal{I}$ with the desired resolution.

Mapping between $\mathcal{I}$ and $\mathcal{I}'$. The size I' of $\mathcal{I}'$ is deduced from the choice of the resolution to which the coordinates of the final points will be set. For example, to achieve a 0.5 pixels resolution, we set $I' = 2I$. More generally, for a resolution of r pixels,

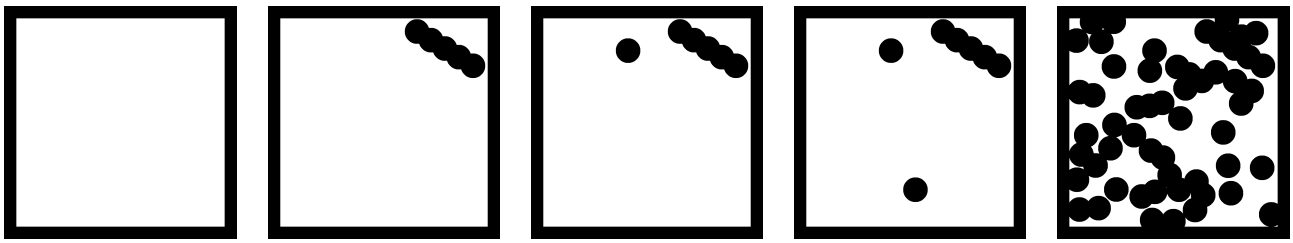
$$I' = \left\lceil \frac{I}{r} \right\rceil \tag{8}$$

where $\lceil \cdot \rceil$ denotes the upper integer part. Similarly, we define in Equation (9) the mapping between the coordinates (x, y) of a point in $\mathcal{I}$ and its coordinates (x', y') in $\mathcal{I}'$. Computing (x', y') from (x, y) is done deterministically, as described by Equation (9).

$$x' = \left\lceil \frac{x}{r} \right\rceil, \; y' = \left\lceil \frac{y}{r} \right\rceil. \tag{9}$$

This mapping from $\mathcal{I}$ to $\mathcal{I}'$ is used only to transform the coordinates of the aligned points prior to set the background points. After defining the coordinates of the background points in $\mathcal{I}'$, we apply an inverse mapping to get their coordinates in $\mathcal{I}$, as described by Equation (10). This inverse mapping ensures the desired resolution but avoids to set the coordinates on the vertices of a lattice, by adding some random noise to the position of the points. Indeed, in Equation (10), u_x and u_y are sampled independently from the uniform distribution on $[0, 1]$.

$$\begin{cases} u_x &= rand([0,1]) \\ u_y &= rand([0,1]) \\ x &= (x' - 1 + u_x) \times r \\ y &= (y' - 1 + u_y) \times r. \end{cases} \tag{10}$$



(a) (b) (c) (d) (e)

Figure 5: Illustration of the procedure to set the coordinates of the background points. Every picture represents $\mathcal{I}'$ at different steps of the procedure. The white pixels represent the authorized region $\mathcal{A}$ at each step. (a) Initialization of $\mathcal{I}'$, only the pixels along the margins are set to $false$, and $\mathcal{A}$ is the inner white square. (b) Update of $\mathcal{A}$ after the dilation of the set $\mathcal{S}$ initialized with the aligned points. (c)-(e) Dilation of $\mathcal{S}$ after definition of one, two and 50 background points.

Setting the coordinates in $\mathcal{I}'$. As we said, $\mathcal{I}'$ is a binary image that we use to define the coordinates of the background points. The coordinates of every new background point in $\mathcal{I}'$ are chosen randomly and uniformly from the authorized region $\mathcal{A}$, defined as the pixels of $\mathcal{I}'$ that are set to the Boolean value "true".

$$\mathcal{A} = \left\{ (x', y') \in [1, I']^2, \mathcal{I}'(x', y') = true \right\}. \tag{11}$$

Therefore, $\mathcal{I}'$ is initialized by setting most of its pixels to "true". The only pixels set to false are those which are close enough to the edges, that is

$$\begin{cases} \mathcal{I}'(x', y') = true & \text{if } d'_b \leq x' \leq I' - d'_b \text{ and } d'_b \leq y' \leq I' - d'_b \\ \mathcal{I}'(x', y') = false & \text{otherwise,} \end{cases} \tag{12}$$

where $d'_b = \left\lceil \frac{d_b}{r} \right\rceil$ is the minimum distance allowed between two points and between a point and the image's edge, in $\mathcal{I}'$.

We will denote $\mathcal{S}$ the set of already existing points. Naturally, $\mathcal{S}$ is initialized with the aligned points defined in section 2.2 and whose coordinates are transformed according to Equation (9). Then, the definition of a new point given the existing points in $\mathcal{S}$, is realized following two steps.

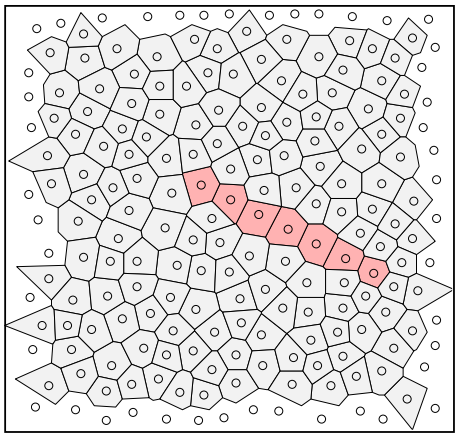


Figure 6: Voronoi cells for a set of points. Only cells that are fully included in the image domain are taken into account. Cells are filled in gray for background points and in light red for the aligned points. The check for density cue is a statistical t-test of the hypothesis that the areas of the target and background cells are samples of two normal distributions with equal means.

- **Step 1: update $\mathcal{A}$.** We compute $\delta_B(\mathcal{S})$, the morphological dilation of $\mathcal{S}$ by the structural element B, which is a discrete disc of radius d'_b (its members comprise all pixels whose centers are no greater than d'_b away from the origin). Then we update $\mathcal{I}'$ by setting to false all pixels in $\delta_B(\mathcal{S})$

$$\forall (x', y') \in \delta_B(\mathcal{S}),\ \mathcal{I}'(x', y') = false. \tag{13}$$

  This update of $\mathcal{I}'$ automatically updates the set $\mathcal{A}$ of authorized points, according to its definition in Equation (11).

- **Step 2: update $\mathcal{S}$.** The new point is chosen randomly according to a uniform distribution over the set $\mathcal{A}$ of authorized points. This new point is then added to $\mathcal{S}$.

These two steps are repeated until $\mathcal{S}$ reaches N points or $\mathcal{A} = \emptyset$. Therefore, the total number of points in the stimulus might be lower than N.

Finally, the coordinates of the background points, that is, the points of $\mathcal{S}$ except the alignment, are transformed following Equation (10) to fit in $\mathcal{I}$. For the alignment in $\mathcal{I}$, we keep the original coordinates, as defined in Section 2.2.

2.4 Checking for Density Cues

Following [3, 17], we check for density cues by comparing the area of the aligned points' Voronoi cells (also called target cells) to those of the background points (also called background cells); see Figure 6. If the areas of the target cells are significantly different from those of the background cells, then we suspect that a density cue may exist in the stimulus.

The set of Voronoi cells. We first compute the Voronoi diagram for the set of points. Then we discard the cells with vertices falling outside the image domain. We will call valid cells the cells which, on the contrary, have all their vertices included in the image domain. Note that among the discarded cells there might be target ones. If strictly more than 20% of the target cells are discarded, which might happen if the alignment is close to an edge, we consider that the remaining target cells are not sufficient to compute reliable statistics. In that case, the stimulus is regarded as suspicious, and might be rejected.

The statistical comparison between target and background cells. If at least 80% of the target cells are kept, then we compare statistically the areas of the valid target cells to those of the valid background cells. We assume that the areas of target and background cells are independent samples of two normal distributions with equal variances, and the tested hypothesis is that these two distributions have equal means. The alternative hypothesis is that one of the two means is greater than the other.

The performed test is a t-test. We note $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_q\}$ respectively the sets of valid Voronoi cells areas for the alignment and the background. Their sample means are respectively noted $\bar{a}$ and $\bar{b}$. The Matlab method *ttest2* computes the sample standard deviations s_a and s_b, from which it deduces a unique pooled sample standard deviation s (Equation (14)).

$$\begin{cases} s_a &= \left(\frac{1}{m-1}\sum_{i=1}^{m}(a_i-\bar{a})^2\right)^{1/2} \\ s_b &= \left(\frac{1}{q-1}\sum_{j=1}^{q}(b_j-\bar{b})^2\right)^{1/2} \\ s &= \left(\frac{(m-1)s_a^2+(q-1)s_b^2}{m+q-2}\right)^{1/2}. \end{cases} \tag{14}$$

Then the value of interest, denoted t, is computed according to Equation (15), and is supposed to follow a Student t-distribution with $m+q-2$ degrees of freedom, if the tested hypothesis is correct.

$$t = \frac{\bar{a}-\bar{b}}{s\sqrt{\frac{1}{m}+\frac{1}{q}}}. \tag{15}$$

If the tested hypothesis holds, t is likely to be close to 0. On the contrary, the greater the absolute value of t, the less likely samples A and B come from distributions with the same means. Noting T_{m+q-2} a random variable following a Student t-distribution with $m+q-2$ degrees of freedom, the associated p-value is therefore

$$p = 1 - \mathbb{P}\Big(-|t| < T_{m+q-2} < |t|\Big) = 2\,\mathbb{P}\Big(T_{m+q-2} \geq |t|\Big). \tag{16}$$

The smaller p, the more confidently the tested hypothesis can be rejected. The user can decide the significance threshold $p_{\min}$ under which the p-value should lead to reject the null hypothesis. A common practice is to set $p_{\min} = 0.05$, but this is only a suggestion. We prefer not to impose a rejection criterion to the user. Indeed, we are aware that the exposed strategy may not be always optimal and could be adapted. Furthermore, we believe it is useful for research purposes to observe the relationship between the statistical p-value and the rendered stimulus. Finally, returning the produced stimulus along with the p-value is the simplest way to guarantee the algorithm ends with an output.

Note that it is also possible to assume that the areas of target and background cells are independent samples of two normal distributions with possibly *different* variances. In that case, if the tested hypothesis of equal means is correct, the computed value t has an approximate Student's t distribution with a number of degrees of freedom given by Satterthwaite's approximation

$$df = \frac{\left(\frac{s_a^2}{m}+\frac{s_b^2}{q}\right)^2}{\frac{1}{m-1}\left(\frac{s_a^2}{m}\right)^2+\frac{1}{q-1}\left(\frac{s_b^2}{q}\right)^2}. \tag{17}$$

This is what was done for the stimuli described in [2]. In the latter method, to produce one stimulus, ten were generated and the one with the greatest p-value was kept. In most cases, the eventual p-value was greater than 0.6.

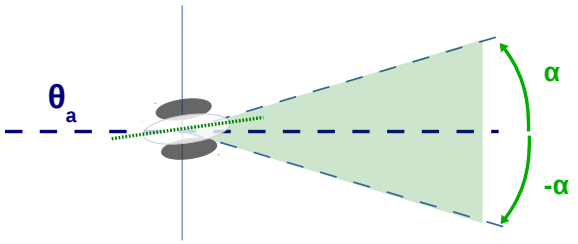


Figure 7: Orientation of a target element, chosen randomly from a uniform distribution over $[\theta_a - \alpha, \theta_a + \alpha]$.

2.5 Setting the Orientations

Target orientations. The orientations of the target elements are randomly chosen from an interval centered on the main direction θ_a of the alignment. The width of this interval is determined by an angle $\alpha \in [0, \frac{\pi}{2}]$, called angular jitter, and which is a parameter of the algorithm. Then the orientations $\theta_1, \theta_2, \ldots, \theta_n$ of the n target elements are independently and uniformly sampled from the interval $[\theta_a - \alpha, \theta_a + \alpha]$.

Background orientations. The orientations of the background elements are sampled independently from a uniform distribution over all possible angles.

This last step completes the definition of the set $\mathcal{G} = \{(x_i, y_i, \theta_i), 1 \leq i \leq N\}$, representing the N oriented elements that will be displayed in the stimulus. Sections 2.6 and 2.7 describe how the actual stimulus is built from $\mathcal{G}$.

2.6 Building a Patch

In our stimuli a patch is a small $h \times h$ pixels square image showing the central part of a symmetrical Gabor function. The family of functions $G_{f,\theta,\sigma}$ we use is defined by

$$\forall (x, y) \in \mathbb{R}^2, \qquad G_{f,\theta,\sigma}(x, y) \;=\; \frac{1}{2}\left(1 + e^{-\frac{x^2+y^2}{2\sigma^2}} \cos\Big(2\pi f(y\cos\theta - x\sin\theta)\Big)\right), \tag{18}$$

where f is the spatial frequency, σ the space constant and θ the orientation of the Gabor element. For our stimuli, we set $h = 15$ pixels, $f = 0.12$ cycles per pixel and $\sigma = \frac{1}{4f}$ pixels. These parameters, as well as the image dimensions ($I = 500$ pixels), were adjusted to produce visually satisfactory stimuli and were also inspired by the literature. However, other values could be used keeping in mind some practical constraints. The patch must be large enough to show the whole part of the Gabor function where it takes values different from the background; and not too large to avoid overlaps among patches. The value of f controls the thickness of this central blob. The smaller f, the thicker the blob and the more blurry the patch looks. Finally, setting σ proportionally to $\frac{1}{f}$ allows to control the number of blobs that are visible in the patch, independently of f. Increasing the coefficient of proportionality tends to let appear more cycles, and therefore more blobs, in the patch.

Given the orientation θ of the patch (that is, the main orientation of the central blob), the value of pixel (i, j) in the patch is given by

$$g(i, j) = G_{f,\theta,\sigma}(h - j + 1 - c, i - c), \tag{19}$$

where $c = \frac{1+h}{2}$ is such that (c, c) are the coordinates of the center of the patch (whether in Cartesian or matricial form).

2.7 Building the Final Stimulus

First, the image $\mathcal{I}$ is initialized to the medium gray value. Then for each element $(x_i, y_i, \theta_i) \in \mathcal{G}$, a patch g_i with orientation θ_i is built as described in Section 2.6. Image $\mathcal{I}$ is actually a matrix, in which the origin of the coordinates is the upper left hand corner, and the first and second coordinates are the row and column indexes respectively. Since (x_i, y_i) are intended as Cartesian coordinates in a classically oriented frame, we need to translate them into image coordinates prior to placing the patch in the image. This is why we define the central pixel of the patch and its top left hand corner as follows

$$\begin{array}{rcl} x_c^{(i)} & = & x_i \\ y_c^{(i)} & = & I - y_i \\ x_{start}^{(i)} & = & round(x_c^{(i)} - \frac{h}{2}) \\ y_{start}^{(i)} & = & round(y_c^{(i)} - \frac{h}{2}), \end{array} \tag{20}$$

where $round(x) = \lfloor x + 0.5 \rfloor$ is the closest integer to x; and finally patch g_i is placed in the image

$$\mathcal{I}(y_{start}^{(i)} : y_{start}^{(i)} + h - 1, x_{start}^{(i)} : x_{start}^{(i)} + h - 1) = g_i. \tag{21}$$

2.8 Examples

Figures 8, 9, 10 and 11 are four examples of stimuli built with the proposed algorithm. The first three examples show satisfactory stimuli in which no significant density cue was found, that is the perceptual grouping only depends on the orientations of the target elements. In Figure 11 however, the alignment is perceptible despite the maximal angular jitter affecting the target elements. The low associated p-value helps as a criterion to reject this stimulus.

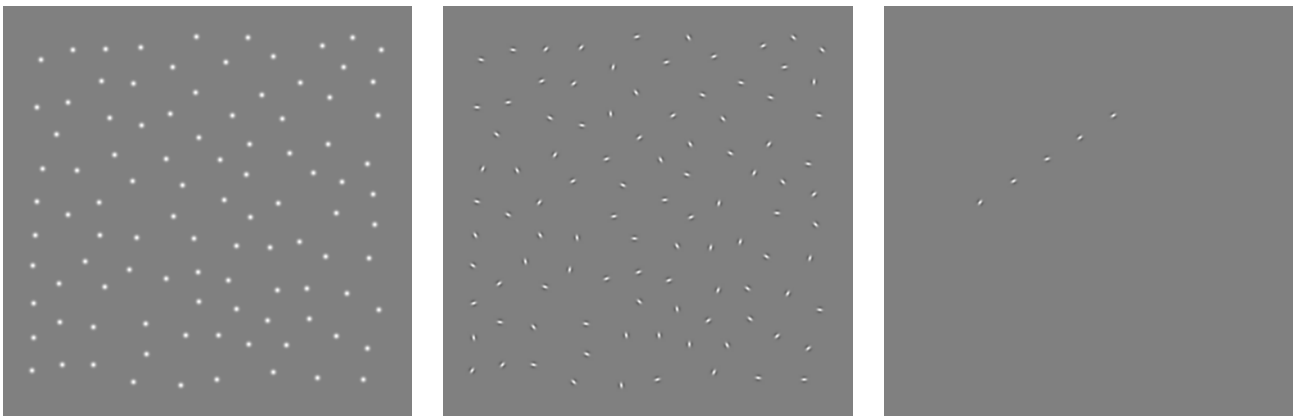


Figure 8: Example of a stimulus with $N = 100$, $n = 5$ and $\alpha = \frac{0.3 \cdot \pi}{2}$. The p-value associated to the t-test is $p = 0.49$, which means that according to the performed test there is no significant difference between the areas of the target Voronoi cells and those of the background cells.

2.9 Formal Description of the Algorithm

The generation algorithm, noted Algorithm 1 is described in detail by its pseudo-code.

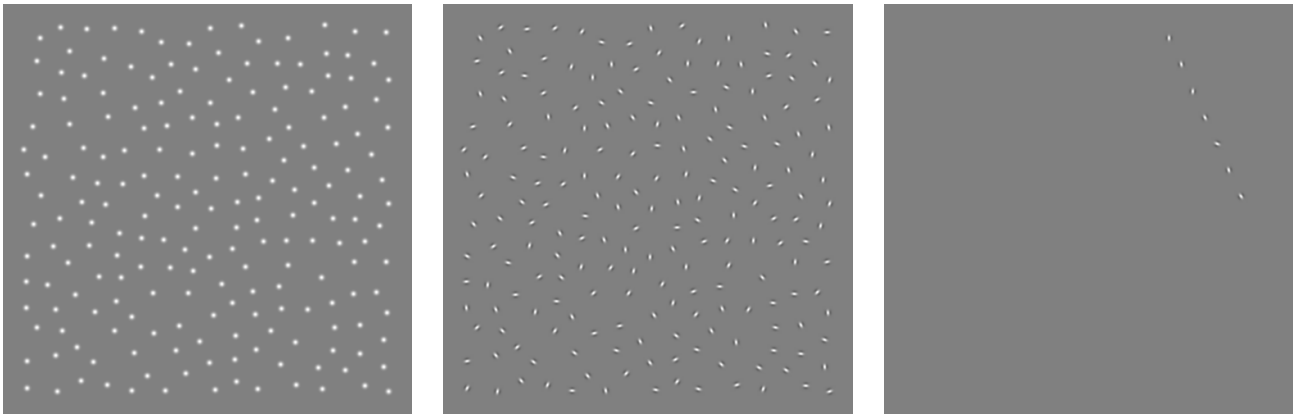


Figure 9: Example of a stimulus with $N = 200$, $n = 7$ and $\alpha = \frac{0.5 \cdot \pi}{2}$. Here the associated p-value is $p = 0.54$. Like in Figure 8, this p-value means that according to the t-test there is no significant difference between the areas of the target and background Voronoi cells.

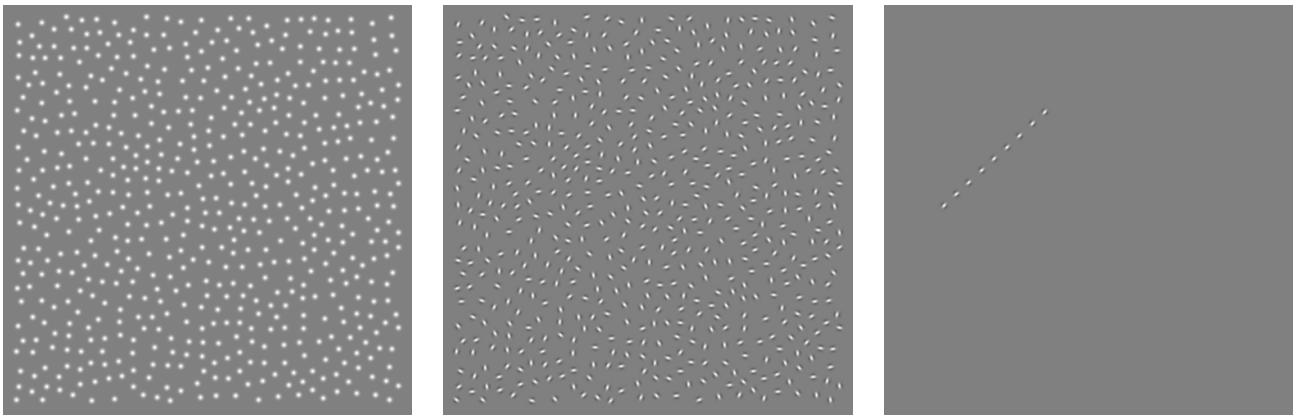


Figure 10: Example of a stimulus with $N = 600$, $n = 9$ and $\alpha = \frac{0.1 \cdot \pi}{2}$. Here the associated p-value is $p = 0.97$. Like in Figures 8 and 9, this p-value means that according to the t-test there is no significant difference between the areas of the target and background Voronoi cells.

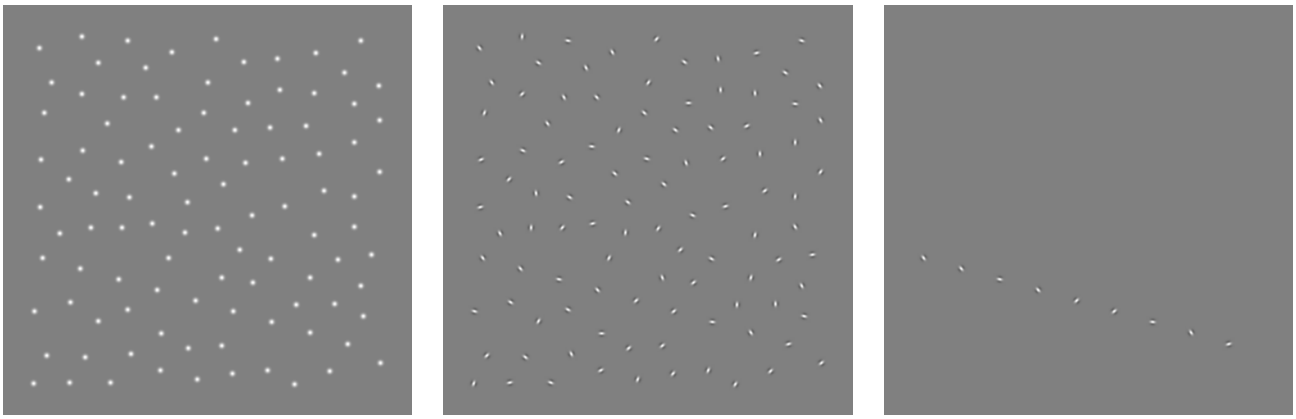


Figure 11: Example of a stimulus with $N = 100$, $n = 9$ and $\alpha = \frac{\pi}{2}$. The p-value associated to the t-test is $p = 0.013$, which indicates, according to the common threshold $p_{\min} = 0.05$, that there might be a significant difference between the areas of the target Voronoi cells and those of the background cells. It is indeed relatively easy to spot the alignment in the left hand image, and in the central image.

Algorithm 1: Generation algorithm

input : The total number N of elements, the number n of aligned points, the level of angular jitter α; $N \in [10, 600]$, $n \geq 3$, $\alpha \in [0, \frac{\pi}{2}]$.

output: The coordinates P and orientations $\mathbf{\Theta}$ of all elements, the indexes of the aligned elements idx_a, the p-value p associated to the statistical check for density cues, the stimulus $\mathcal{I}$.

$I \leftarrow 500$ *// Size of the stimulus' side, in pixels*

$h \leftarrow 15$ *// Size of one patch's side, in pixels*

$f \leftarrow 0.12$ *// Spatial frequency, in cycles per pixel*

$\sigma \leftarrow \frac{1}{4f}$ *// Spatial constant, in pixels*

$(P, idx_a, \theta_a) \leftarrow$ `setCoordinates`(I, N, n)

$actual\_N \leftarrow \#P$*// The actual number of points that could fit in the image - usually equal to* N

$p \leftarrow$ `checkDensityCue`(P, I, idx_a)*// This p-value is not used further in the algorithm. It informs the user about the quality of the stimulus (see Section 2.4 and the pseudo-code of* `checkDensityCue`*).*

$\mathbf{\Theta} \leftarrow$ `setOrientations`$(\theta_a, \alpha, idx_a, actual\_N)$

$\mathcal{I} \leftarrow$ `buildStimulus`$(I, h, P, \mathbf{\Theta}, \sigma, f)$

return $P, \mathbf{\Theta}, p, idx_a, \mathcal{I}$

Function setCoordinates(I, N, n_{in}). This function, called in Algorithm 1, takes as arguments the image side I in pixels, the total number N of elements and the number n_{in} of aligned elements. The function returns a list P of N coordinates, representing the positions of the N elements; the set idx_a of indexes of the aligned elements in P; the orientation θ_a of the alignment.

$\lambda \leftarrow 1.5$

$d_a \leftarrow \frac{2I}{\sqrt{2\sqrt{3}N+4}}$

$d_b \leftarrow \frac{\lambda I}{\sqrt{2\sqrt{3}N+2\lambda}}$

$(S, \theta_a) \leftarrow$ `placeAlignment`(I, n_{in}, d_a, d_b)

$n \leftarrow \#S$

$idx_a \leftarrow (1, \ldots, n)$

$\rho \leftarrow 0.5$

$P \leftarrow$ `placeBackground`$(S, N, \rho, I, d_b, d_b)$

return P, idx_a, θ_a

Function placeAlignment(I, n_{in}, d_a, d_{marg}). This function, called in *setCoordinates*, takes as arguments the image side I in pixels, the number n_{in} of aligned elements, the distance d_a between consecutive aligned elements, in pixels, and the minimal distance d_{marg} of any element from the image's edge, in pixels as well. If n_{in} is too big ($> n_{\max}$), the actual number of aligned elements will be $n_{\max}$. The function returns a set S of coordinates of the n aligned elements, and the direction θ_a of this alignment.

$\theta_a \leftarrow 2\pi \cdot rand([0,1])$
$\mu_x \leftarrow rand([0,1])$
$\mu_y \leftarrow rand([0,1])$
$n_{\max} \leftarrow \left\lfloor \frac{I-2d_{marg}}{d_a} \right\rfloor + 1$ // *Maximal number of aligned elements that can fit in the image for any* θ_a
$n \leftarrow \min(n_{in}, n_{\max})$
$L \leftarrow (n-1)d_a$ // *Length of the alignment*
$x_c^{\min} \leftarrow d_{marg} + \frac{L}{2}|\cos(\theta_a)|$
$x_c^{\max} \leftarrow I - d_{marg} - \frac{L}{2}|\cos(\theta_a)|$
$y_c^{\min} \leftarrow d_{marg} + \frac{L}{2}|\sin(\theta_a)|$
$y_c^{\max} \leftarrow I - d_{marg} - \frac{L}{2}|\sin(\theta_a)|$
$x_c \leftarrow x_c^{\min} + \mu_x \cdot (x_c^{\max} - x_c^{\min})$
$y_c \leftarrow y_c^{\min} + \mu_y \cdot (y_c^{\max} - y_c^{\min})$
$\overrightarrow{v_a} \leftarrow (cos(\theta_a), sin(\theta_a))$
$L \leftarrow (n-1) \times d_a$
$s_1 \leftarrow (x_c, y_c) - \frac{L}{2}\overrightarrow{v_a}$
$S \leftarrow \{s_1\}$
for $k = 2$ **to** n **do**
  $s_k \leftarrow s_1 + (k-1)d_a\overrightarrow{v_a}$
  $S \leftarrow S \cup \{s_k\}$
end
return S, θ_a

Function placeBackground$(S, N, r, I, d_{marg}, d_{\min})$. This function, called in *setCoordinates*, takes as arguments the set S of coordinates of the aligned elements, the total number N of elements, the spatial resolution r for the coordinates of the elements, the image side I in pixels, the minimal distance d_{marg} of any element from the image's edge in pixels, and the minimal distance $d_{\min}$ of any element from its closest neighbor. The function returns a set P of N coordinates, representing the positions of the N elements.

$I' = \left\lceil \frac{I}{r} \right\rceil$
$d'_{\min} = \left\lceil \frac{d_{\min}}{r} \right\rceil$
$d'_{marg} = \left\lceil \frac{d_{marg}}{r} \right\rceil$
$S' \leftarrow \left\lceil \frac{1}{r} \cdot S \right\rceil$
// Initialize $\mathcal{I}'$ according to Equation (12), as illustrated in Figure 5(a)
$\mathcal{I}' \leftarrow false(I', I')$
for $x' = d'_{marg} + 1$ **to** $I' - d'_{marg}$ **do**
  for $y' = d'_{marg} + 1$ **to** $I' - d'_{marg}$ **do**
    $\mathcal{I}'(x', y') \leftarrow true$
  end
end
// Second step of the initialization: turn to false the discs around the aligned elements, as illustrated in Figure 5(b)
$\mathcal{D} \leftarrow \texttt{structuralElement}(disk, d'_{\min})$
$pool \leftarrow false(I', I')$
$pool(S') \leftarrow true$*// Set to true the points in pool with the coordinates contained in S'*
$pool \leftarrow \texttt{dilation}(pool, \mathcal{D})$*// Dilation of the aligned elements*
$\mathcal{I}' \leftarrow \mathcal{I}'$ and $\neg\, pool$*// Update $\mathcal{I}'$ according to Equation (13)*
$\mathcal{A} \leftarrow \{(x', y') \in [1, I']^2, \mathcal{I}'(x', y') = true\}$*// Authorized region as defined by Equation (11)*
// Start the loop to place background elements, as illustrated in Figure 5(c), (d) and (e)
$placed \leftarrow \#S$*// Initialize the number of already placed elements to the number of aligned elements that could be placed*
$B' \leftarrow \emptyset$*// Initialize the set of discrete background points*
while $placed < N$ *and* $\mathcal{A} \neq \emptyset$ **do**
  $newPt \leftarrow rand(\mathcal{A})$*// Choose randomly an element from the authorized region*
  $B' \leftarrow B' \cup newPt$*// Add the new element to the set of discrete background points*
  $pool \leftarrow false(I', I')$
  $pool(newPt) \leftarrow true$
  $pool \leftarrow \texttt{dilation}(pool, \mathcal{D})$*// Dilation of the placed elements*
  $\mathcal{I}' \leftarrow \mathcal{I}'$ and $\neg\, pool$*// Update $\mathcal{I}'$ according to Equation (13)*
  $placed \leftarrow placed + 1$
  $\mathcal{A} \leftarrow \{(x', y') \in [1, I']^2, \mathcal{I}'(x', y') = true\}$*// Authorized region as defined by Equation (11)*
end
// Transform the discrete coordinates of background points back to non-integer coordinates
$B \leftarrow \emptyset$
for $p' \in B'$ **do**
  $p \leftarrow (p' - 1)r + r \times rand([0, 1]^2)$
  $B \leftarrow B \cup p$
end
$P \leftarrow S \cup B$
return P

Function checkDensityCue(P, I, idx_a). This function, called in Algorithm 1, takes as arguments a set P of N coordinates, representing the positions of the N elements, the image side I in pixels, and the list idx_a of indexes of the points in P belonging to the alignment. The function returns a p-value associated to a statistical comparison between the areas of the Voronoi cells of the aligned elements and those of the background elements. The user shall see p as a criterion to decide whether to reject the stimulus or not. If p is smaller than a significance threshold $p_{\min}$ (to be chosen by the user, typically $p_{\min} = .05$), the stimulus should be rejected because the difference between the two sets of areas will be considered as significant. The value $p = 0$ means that the statistical test was not performed, and that the stimulus should be rejected no matter the significance threshold $p_{\min}$.

$V \leftarrow \texttt{voronoi}(P)$
$cellsInside \leftarrow false(\#V)$
for $i = 1$ **to** $\#V$ **do**
  if $V(i) \subset [0, I]^2$ **then**
    $cellsInside(i) \leftarrow true$
    if $i \in idx_a$ **then**
      $cellsAlignmentInside \leftarrow cellsAlignmentInside \cup i$
    else
      $cellsBackgroundInside \leftarrow cellsBackgroundInside \cup i$
    end
  end
end
if $\#cellsAlignmentInside < 0.8 \cdot \#idx_a$ **then**
  $p = 0$
  return p
end
$\mathcal{S}_a \leftarrow \texttt{surface}(V(cellsAlignmentInside))$
$\mathcal{S}_b \leftarrow \texttt{surface}(V(cellsBackgroundInside))$
$p \leftarrow \texttt{ttest}(\mathcal{S}_a, \mathcal{S}_b)$ *// t-test, see Section 2.4*
return p

Function SetOrientations($\theta_a, \alpha, idx_a, N$). This function, called in Algorithm 1, takes as arguments the direction θ_a of the alignment, the angular jitter α, the list idx_a of indexes in $\{1, \ldots, N\}$ corresponding to the aligned elements, and the total number N of elements.

$n \leftarrow \#idx_a$
$\mathbf{\Theta} \leftarrow 2\pi \cdot rand[N]$ *// N random angles*
for $i = 1$ **to** N **do**
  if $i \in idx_a$ **then**
    $\mathbf{\Theta}(i) \leftarrow \theta_a + \alpha \cdot (2 \cdot rand([0, 1]) - 1)$
  else
    $\mathbf{\Theta}(i) \leftarrow 2\pi \cdot rand([0, 1])$
  end
end
return $\mathbf{\Theta}$

Function buildStimulus$(I, h, P, \mathbf{\Theta}, \sigma, f)$. This function, called in Algorithm 1, takes as arguments the dimension I of the image's side, the dimension h of a patch's side, the list P of the patches' coordinates, the list Θ of the patches' orientations, the space constant σ and spatial frequency f that determines the aspect of a patch; the function returns the stimulus itself: a $I \times I$ pixels image representing the set of Gabor patches.

$N \leftarrow \#P$
$\mathcal{I} \leftarrow 0.5 \cdot ones(I, I)$
for $i = 1$ **to** N **do**
  $x_c^{(i)} \leftarrow P_i.x$
  $y_c^{(i)} \leftarrow I - P_i.y$
  $x_{start}^{(i)} \leftarrow round(x_c^{(i)} - \frac{h}{2})$
  $y_{start}^{(i)} \leftarrow round(y_c^{(i)} - \frac{h}{2})$
  $c_y \leftarrow y_{start}^{(i)} + \frac{h-1}{2}$
  $c_x \leftarrow x_{start}^{(i)} + \frac{h-1}{2}$
  for $k = y_{start}^{(i)}$ **to** $y_{start}^{(i)} + h - 1$ **do**
    for $l = x_{start}^{(i)}$ **to** $x_{start}^{(i)} + h - 1$ **do**
      $k_c \leftarrow h - k + 1 - c_y$
      $l_c \leftarrow l - c_x$
      $\mathcal{I}(k, l) \leftarrow G_{f,\theta_i,\sigma}(l_c, k_c)$ // *Equation* (18)
    end
  end
end
return $\mathcal{I}$

3 Detection Algorithm

The detection algorithm described in this section is based on the *a contrario* theory, which is a mathematical formulation of the non-accidentalness principle. An *a contrario* method requires two models. On the one hand, a geometric model, which is deterministic and, on the other hand, a probabilistic model.

The geometric model defines an ideal structure x^* along with a function $d_{x^*}(\cdot)$ measuring a deviation from it. For example, in our case, the ideal structure is a set of aligned Gabor patches with perfectly aligned orientations. The measure of a deviation from this ideal configuration, is detailed in Section 3.1 but we can already provide a hint of it: in Figure 12 (e) for example, the smaller the angles α_i, the closer the depicted chain to a perfect alignment.

The probabilistic model, also called *a contrario* or *background* model, is the statistical hypothesis H_0 of the absence of relevant structure - the so called *null hypothesis*. It represents the most general assumption on the data. Consistently with Attneave's principle stating that we do not perceive any structure in white noise [1], an *a contrario* model generally gives a maximal entropy to the building blocks of the percept. For example if these building blocks are oriented, their orientations must be random, independent, and uniformly distributed for each block. By Attneave's principle the emergence of a percept in a realization of such a model should be unlikely, and in any case purely accidental. Therefore this background model is used to test the significance of an observation, as follows. Let X be a random variable consistent with H_0. Given an observed structure x with a deviation $d_{x^*}(x)$ from the ideal structure, its relevance is measured by the probability $\mathbb{P}_{H_0}(d_{x^*}(X) \leq$

$d_{x^*}(x)$) to be at least as close to x^* under H_0. Small deviations yield small probabilities, and are thus rare events in the background model.

To evaluate the non-accidentalness of an event, an *a contrario* method takes into account the whole set of observations that were necessary to come across that particular event. For example, when looking for alignments in an array of Gabor patches, we define *a priori* a family of sets of patches, that will be tested as candidates to be alignments. This is what is usually called the *family of tests*, noted T, and N_T denotes the number of tests in T. Then, given a tested candidate x that yielded the deviation $d_{x^*}(x)$, denote by $X_1, X_2, \ldots, X_{N_T}$, the random issues to the N_T tests, under the background model H_0. The question is now how common or, on the contrary, how surprising it would be to observe a deviation smaller than $d_{x^*}(x)$ among these N_T tests. An answer is to estimate the expected number of variables X_i that verify $d_{x^*}(X_i) \leq d_{x^*}(x)$. The *a contrario* theory provides as an estimate a quantity called *Number of False Alarms* (NFA), defined as $\text{NFA}(x) = N_T \times \mathbb{P}[d_{x^*}(X) \leq d_{x^*}(x)]$. In general the NFA is an upper-bound of the number of deviations to the ideal model smaller than $d_{x^*}(x)$ that are expected to happen by accident in a set of N_T random tests.

In the following subsections, we detail the geometric and *a contrario* models for our study.

3.1 The Geometric Model

For a given a tuple $(g_1, \ldots, g_n)$ of n Gabor patches, we consider the variables $\alpha_1, \alpha_{2L}, \alpha_{2R}, \ldots, \alpha_{(n-1)L}, \alpha_{(n-1)R}, \alpha_n$, as illustrated in Figure 12(e). Variable α_{iL} is the absolute angle between the orientation of Gabor patch i and the line joining it to the patch $i-1$, while α_{iR} is the same thing changing $i-1$ for $i+1$. Since the first and last patches in the tuple have no previous and next elements respectively, we simply note $\alpha_1 \stackrel{\text{def}}{=} \alpha_{1R}$ and $\alpha_n \stackrel{\text{def}}{=} \alpha_{nL}$. Then we define $\omega_1 \stackrel{\text{def}}{=} \alpha_1$, $\omega_n \stackrel{\text{def}}{=} \alpha_n$ and for $i = 2, \ldots, n-1$, $\omega_i \stackrel{\text{def}}{=} \max(\alpha_{iL}, \alpha_{iR})$; see Figure 12(f).

The ideal alignment that will be our reference structure in the present study, is such a tuple of Gabor patches for which all angles ω_is are equal to 0 rads. In words, it is a set of aligned patches whose orientations are the same as their line's orientation.

Then we measure the deviation of a general tuple from an ideal alignment by its maximum angle $\omega^* \stackrel{\text{def}}{=} \max\{\omega_1, \ldots, \omega_n\}$.

3.2 The A Contrario Model

Now we need to define the *a contrario* model for our stimuli. Figure 13 shows three arrays of Gabor patches with *only background elements* and thus no significant alignment. Formally, we model an array of N Gabor patches by a set $\mathbf{g} = \{(x_i, \theta_i)\}_{i=1\ldots N}$, where $x_i \in \mathbb{R}^2$ represents the coordinates of patch number i, and $\theta_i \in \mathbb{R}$ its orientation. We note $\mathbf{x} \stackrel{\text{def}}{=} \{x_1, \ldots, x_N\}$ and, for any three points $x, y, z \in \mathbf{x}$, ϕ_{xyz} is the angle between vectors $\overrightarrow{xy}$ and $\overrightarrow{yz}$. Finally, the set of the 6 nearest neighbors of a point $x \in \mathbf{x}$ is noted $\mathcal{N}_6(x)$ (see Figures 12 (a) and (b)). Then we define an *a contrario* array of Gabor patches as a random set $\mathbf{G} = \{(X_i, \Theta_i)\}_{i=1\ldots N}$ verifying two properties:

1. The random variables $\Theta_1, \ldots, \Theta_N$ are independent and uniformly distributed in $[0, 2\pi)$.

2. For any $X \in \mathbf{X}$ and $Y \in \mathcal{N}_6(X)$ the angle $\Phi^*_{XY} \stackrel{\text{def}}{=} \min_{Z \in \mathcal{N}_6(Y)} |\Phi_{XYZ}|$ is uniformly distributed in $[0, \frac{\pi}{6})$ (see Figure 12 (d)), and is independent from $\Phi^*_{X'Y'}$ for any $(X', Y') \neq (X, Y)$.

Let's clarify the relation between this definition and the background stimuli of Figure 13. Property 1 corresponds exactly to the rule we used to define the background elements' orientations in our arrays of patches (see Section 2.5). The relevance of property 2 needs more justification. As explained in Sections 2.1 and 2.3, we built each stimulus so that N elements fit in it, that two elements were not too close to each other, and that there were no empty regions in the image. The most compact

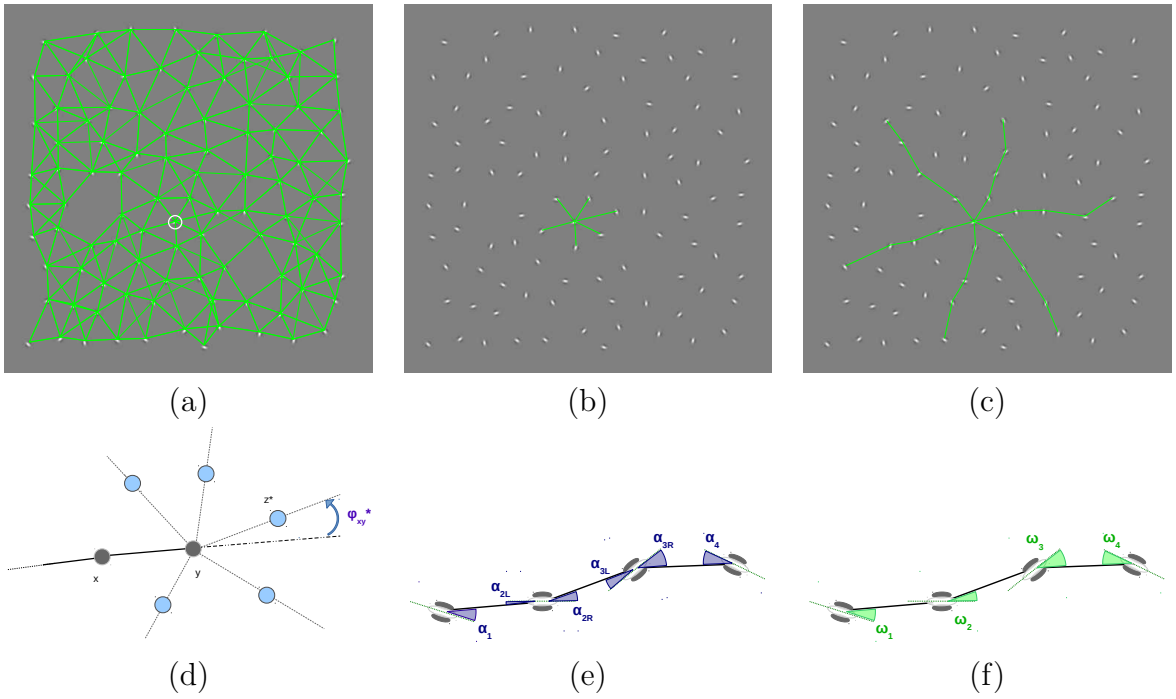


Figure 12: (a) On this array of 100 Gabor patches, we represent the associated graph γ (the orientations of the edges were omitted). In this graph, most of the points are linked to their 6 nearest neighbors, except when a neighbor is too remote, as it often happens for elements that are close to the image border. For a point $x \in \mathbf{x}$ (circled in white in (a)), a chain (x, y) is started for each $y \in \mathcal{N}_\gamma(x)$ in (b), and expanded in (c) into the most rectilinear possible chain. Picture (c) shows the result after 3 iterations of the expansion process. (d) Given two neighbors x and y, we look for the neighbor z of y that minimizes the absolute angle between $\overrightarrow{xy}$ and $\overrightarrow{yz}$. (e)-(f) The variables the algorithm measures when analyzing a chain containing $n = 4$ points.

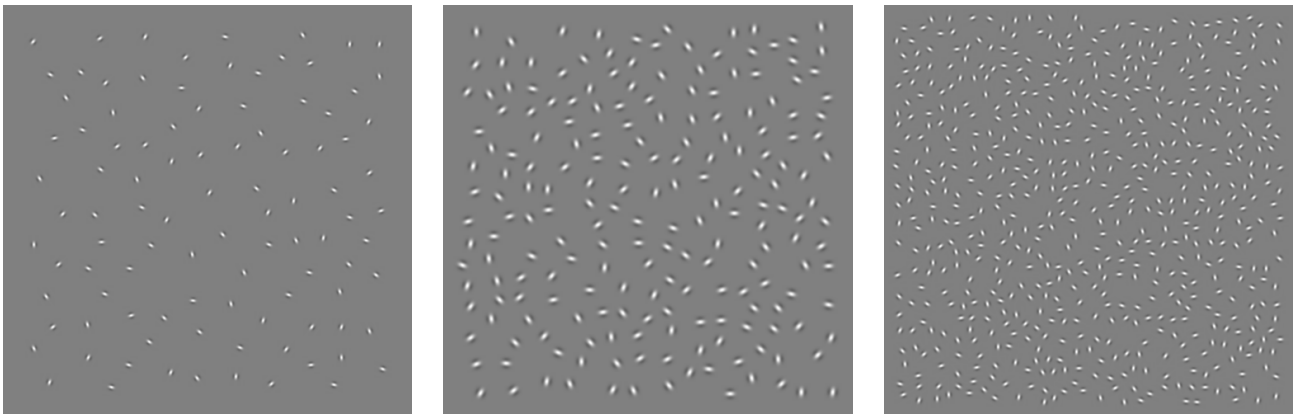


Figure 13: Three arrays of Gabor patches containing only background elements and illustrating the *a contrario* model.

way to fit discs in a given region is to lay them on a hexagonal lattice. Here we chose the minimal distance between elements in order to allow some randomness in their coordinates. The resulting stimuli look like hexagonal lattices affected by some noise, in which property 2 approximately holds (see Figure 12 (a)). As illustrated in Figure 12 (d), in such an approximately hexagonal lattice, we expect the maximal value for the angle Φ^*_{XY} to be roughly $\frac{\pi}{6}$. Since we do not know the actual distribution of Φ^*_{XY}, the simplest assumption is then the uniform distribution in $[0, \frac{\pi}{6})$, which does

not require to set any parameter. We checked that the empirical distribution of Φ^*_{XY} is reasonably close to our assumption.

3.3 Input of the Algorithm

The input of the detection algorithm is a set $\mathbf{g} = \{(x_i, \theta_i)\}_{i=1\ldots N}$, that models a Gabor array composed of N patches (see Section 3.2).

3.4 The Family of Tests

The family of tests is defined by first building a graph γ on the set $\mathbf{x} = \{x_1, \ldots, x_N\}$, and then selecting chains of adjacent edges in this graph.

The graph γ. The oriented graph $\gamma = (\mathbf{x}, \mathbf{e})$, where $\mathbf{x}$ is the set of vertices and $\mathbf{e}$ the set of edges, is defined as follows: $(x_i, x_j) \in \mathbf{e}$ if and only if x_j is one of the 6 nearest neighbors of x_i and $d(x_i, x_j) \leq d_{\max}$, where $d(x_i, x_j)$ is the Euclidean distance between the two points, and $d_{\max}$ the maximal length we accept for an edge (see next paragraph). We will denote by $\mathcal{N}_\gamma(x)$ the set of neighbors of x in γ; see Figure 12 (a).

Maximal edge's length. The maximal length for an edge is based on a statistic of the distances between a point and its nearest neighbor in $\mathbf{x}$. We list the N distances of each point $x_i \in \mathbf{x}$ to its nearest neighbor, and define d_{avg} as the mean value of this list. Then we take as maximal length for an edge $d_{\max} \stackrel{\text{def}}{=} 2d_{avg}$. This is just an empirical choice to avoid too long edges in the graph.

The chains to be tested. A chain $(z_1, z_2, \ldots, z_n)$ is in the family of tests $\mathcal{T}$ if and only if the following conditions are fulfilled

$$\begin{cases} n \geq 3 \\ \forall i \in \{1, \ldots, n-1\}, \quad (z_i, z_{i+1}) \in \mathbf{e} \\ \forall i \in \{3, \ldots, n\}, \quad z_i = \min_{\in \mathcal{N}_\gamma(z_{i-1})} |\phi_{z_{i-2}\, z_{i-1}\, z}|, \end{cases} \tag{22}$$

where $\phi_{z_{i-2}\, z_{i-1}\, z}$ is the angle between vectors $\overrightarrow{z_{i-2}\, z_{i-1}}$ and $\overrightarrow{z_{i-1}\, z}$. In words, the chains to be tested are sequences of adjacent elements in γ that minimize the angles between consecutive edges. Figure 12 (c) shows six examples of chains to be tested, each starting from the same point and composed of five elements.

The number of tests is approximately

$$N_T \stackrel{\text{def}}{=} 6 \times N \times \sqrt{N}. \tag{23}$$

N_T is actually an overestimation of the number of tests, since not all nodes in γ have six neighbors, and we only test chains containing at least three elements.

3.5 The NFA

The NFA of each chain $c = (z_1, \ldots, z_n)$ of $\mathcal{T}$ is defined as follows. Consider the variables $\omega_1, \ldots, \omega_n$, as defined in Section 3.1 and illustrated in Figures 12(e) and (f). Under the assumption that $\mathbf{g}$ is a realization of an *a contrario* array of Gabor patches, the orientations $\omega_1, \omega_2, \ldots, \omega_n$ are samples of n independent random variables $\Omega_1, \Omega_2, \ldots, \Omega_n$. Ω_1 and Ω_n are uniformly distributed in $[0, \frac{\pi}{2}]$,

and a short development, presented in the Appendix, shows that $\Omega_2, \dots, \Omega_{n-1}$ have a cumulative distribution function $F(\omega) = \mathbb{P}(\Omega_2 \leq \omega) = \dots = \mathbb{P}(\Omega_{n-1} \leq \omega)$, defined by

$$F(\omega) = \mathbb{P}(\Omega \leq \omega) = \begin{cases} \frac{12}{\pi^2}\omega^2 & \text{if } 0 \leq \omega < \frac{\pi}{12} \\ \frac{2}{\pi}\omega - \frac{1}{12} & \text{if } \frac{\pi}{12} \leq \omega < \frac{5\pi}{12} \\ \frac{12}{\pi^2}\omega^2 - \frac{8}{\pi}\omega + 2 & \text{if } \frac{5\pi}{12} \leq \omega \leq \frac{\pi}{2}. \end{cases} \tag{24}$$

Note that for a random patch Z_i in a chain, the probability law of $\max(\alpha_{iL}, \alpha_{iR})$ depends on the angle $\Phi^*_{Z_{i-1}Z_i}$. The cumulative distribution function F is obtained by integrating over all possible values for $\Phi^*_{Z_{i-1}Z_i}$, according to the law hypothesized in the *a contrario* model. Noting $\omega^* \overset{\text{def}}{=} \max\{\omega_1, \dots, \omega_n\}$ and $\Omega^* \overset{\text{def}}{=} \max\{\Omega_1, \dots, \Omega_n\}$, the probability that all Ω_i be less than the observed ω^* is

$$\mathbb{P}\left(\Omega^* \leq \omega^*\right) = \left(\frac{2\omega^*}{\pi}\right)^2 \times F(\omega^*)^{n-2}. \tag{25}$$

As usual in the *a contrario* theory, we get a natural definition of the NFA of chain c

$$\begin{aligned} \text{NFA}(c) &\overset{\text{def}}{=} N_T \times \mathbb{P}\left(\Omega^* \leq \omega^*\right) \\ &= N_T \times \left(\tfrac{2\omega^*}{\pi}\right)^2 \times F(\omega^*)^{n-2}. \end{aligned} \tag{26}$$

3.6 Output of the Algorithm

The algorithm returns the lowest NFA and the chain that achieves it. When the *a contrario* theory is applied to computer vision, the common practice is to detect only structures with NFA below a certain threshold (typically, NFA = 1). Like in [2], we want to compare the detection algorithm to visual perception on an objective task, in which both the subject and the algorithm are forced to report the most salient structure in the stimulus. If we allowed the algorithm to report no detection at all, we should give the same possibility to human observers, and this would introduce more subjectivity: the subject could actually *see* something and decide not to consider it as a significant alignment. Moreover, we are also interested in the NFA value of the target alignment, whether it is detected or not (see demo).

3.7 Examples of Detections

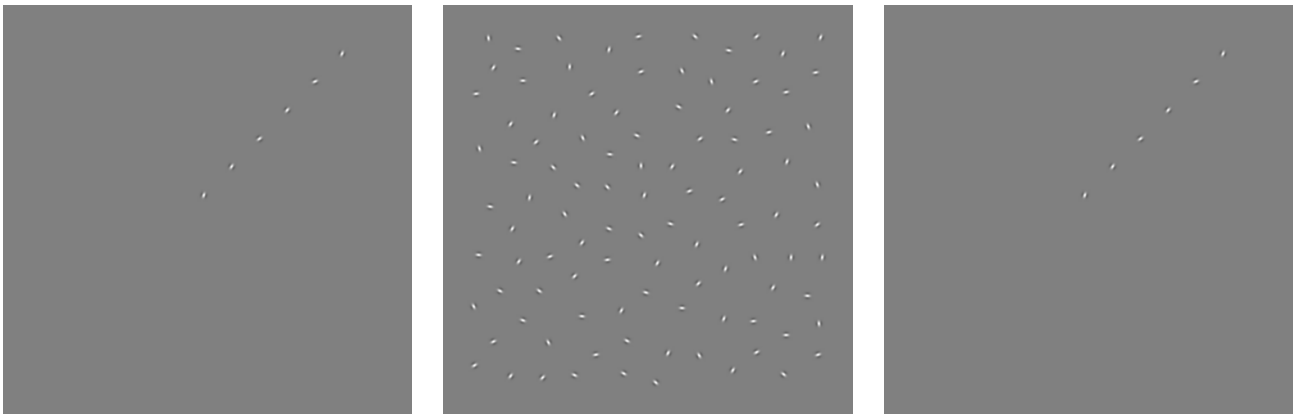
In this section we present a few examples illustrating the behavior of the detection algorithm. Figures 14 and 15 are the most representative cases: the algorithm detects the target alignment when it is conspicuous (Figure 14), but does not when it is perceptually hard to distinguish the alignment from the background (Figure 15).

Figure 16 and 17 show two examples where human observers perceive the target alignment whereas it is not detected by the algorithm. In Figure 16, despite the high level of orientation jitter, human subjects could detect a part of it, composed of the third, fourth and fifth elements starting from the top. The reason is probably that the position masking left a larger space around these three elements. Note that the statistical test on density cue yields a p-value of 0.79, and therefore does not point out the inefficiency of the masking here. Figure 16 also shows the algorithm's tendency to detect curved structures when no salient alignment is present. In Figure 17, the algorithm prefers a short alignment, although the target alignment seems more salient.

On the contrary, Figure 18 illustrates the cases where the algorithm finds the target alignment whereas the latter is not so easy to perceive for a human observer. Because it is quite long and not so affected by angular jitter the target is expected to be easy to spot. However in this stimulus, other alignment-like structures occurred by chance in a more central part of the image, which tend to draw our attention away from the target alignment.

Finally, in Figure 19 we see how the algorithm sometimes detects a chain that includes the target alignment. The latter is not significant enough to be detected alone, but participates in the detection of the bigger set. In the present example the algorithm contradicts our perception, since we tend to see the target and not the longer chain. The fact that the detected structure is curved instead of a straight alignment, shows another limitation of our algorithm. We tried several strategies to constrain the curvature of the chains by including it in the computation of the NFA. In general, this did prevent to detect curves rather than alignments. However, for all the formulations we tried, this also often led to detecting target alignments that were not perceived by human observers. In other words, we have not found the appropriate trade-off between curvature and consistency within orientations. The algorithm presented here is, among the approaches we tried, the one that best matches the human data [2].

The examples presented above are meant to give an idea of the limitations of the algorithm in some particular cases. However, in general, the algorithm provides quite accurate predictions of the perception of alignments in our stimuli. For a detailed presentation of these results, we kindly refer to [2].

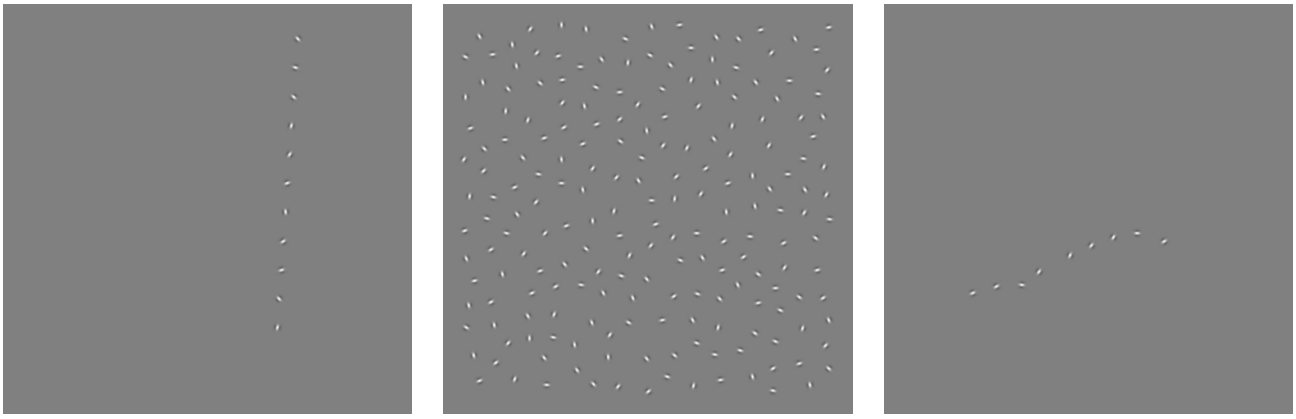


Target alignment | Stimulus (input) | Detection: NFA = 0.91

Figure 14: Example of detection in a stimulus composed of $N = 100$ patches, with $n = 6$ elements affected by a jitter $\alpha = \frac{0.3\pi}{2}$. In general the algorithm detects the target alignment when it is conspicuous.

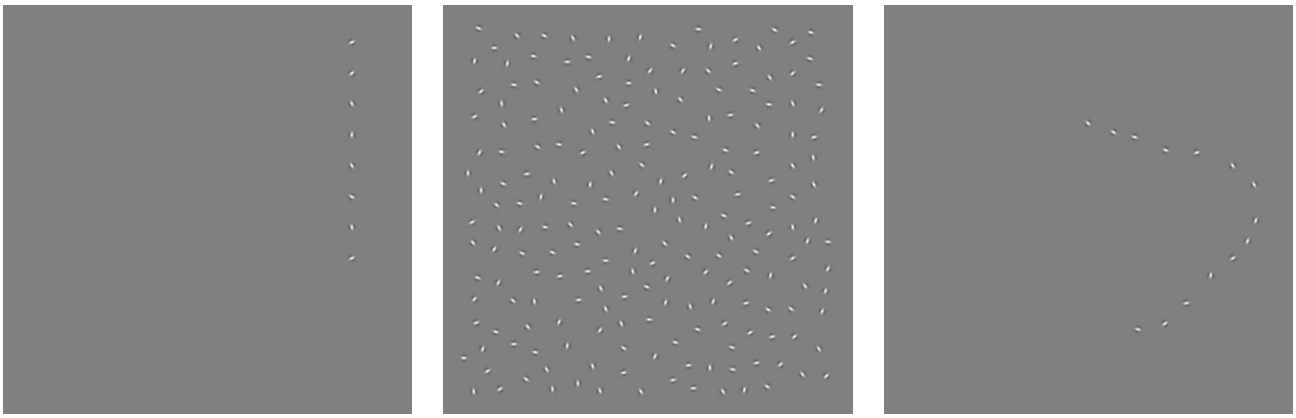
3.8 Formal Description of the Algorithm

The detection algorithm, noted Algorithm 2 is described in detail by its pseudo-code.



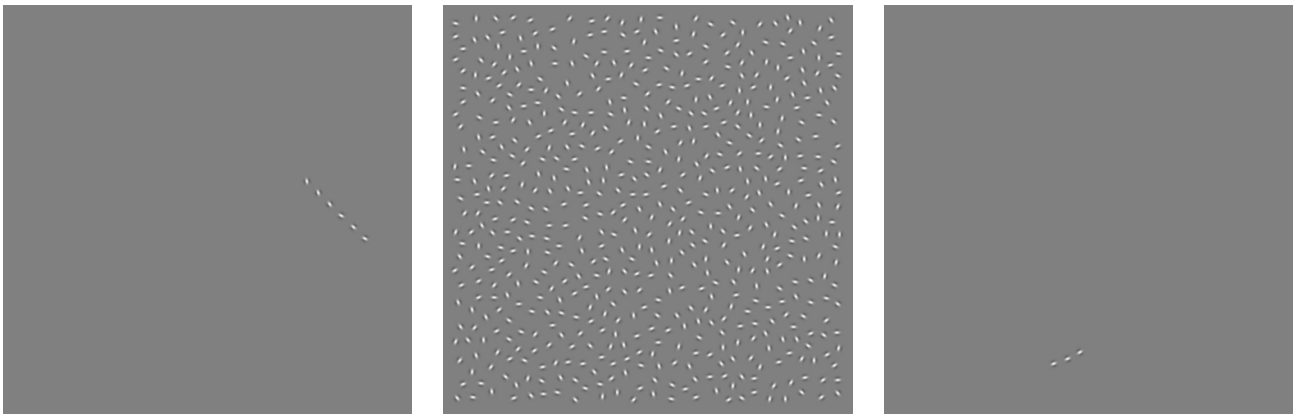
Target alignment | Stimulus (input) | Detection: NFA = 22

Figure 15: Example of detection in a stimulus composed of $N = 200$ patches, with $n = 9$ elements affected by a jitter $\alpha = \frac{\pi}{2}$. In general the algorithm does not detect the target alignment when it is perceptually hard to distinguish from the background.



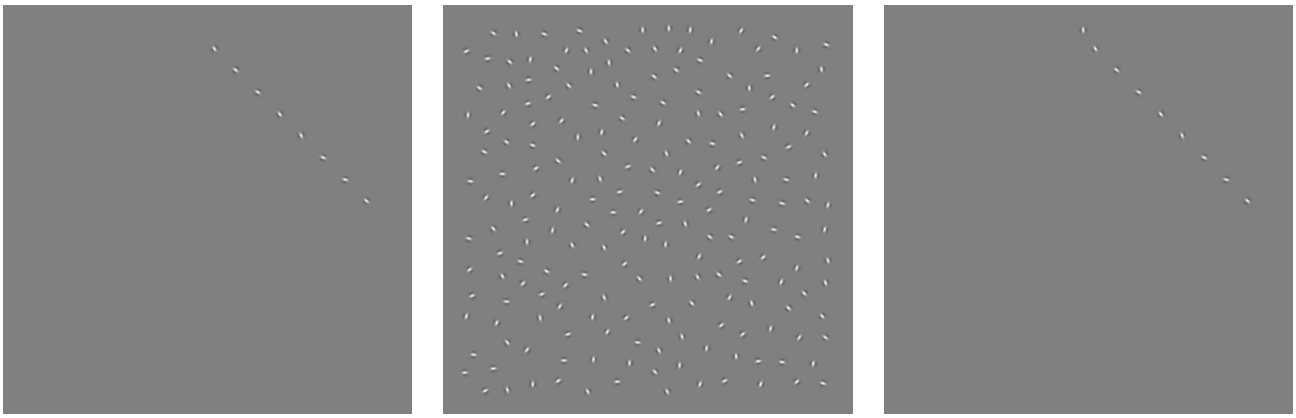
Target alignment | Stimulus (input) | Detection: NFA = 0.017

Figure 16: Another example of detection in a stimulus composed of $N = 200$ patches, with $n = 8$ elements affected by a jitter $\alpha = \frac{\pi}{2}$. Despite the high level of orientation jitter, it is possible for an observer to see a part of the target alignment, composed of the third, fourth and fifth elements starting from the top. The reason is probably that the position masking left a larger space around these three elements - although the statistical test on density cue yields a p-value of 0.79, and therefore does not point out the inefficiency of the masking here. The present figure also shows the algorithm's tendency to detect curved structures when no salient alignment is present.



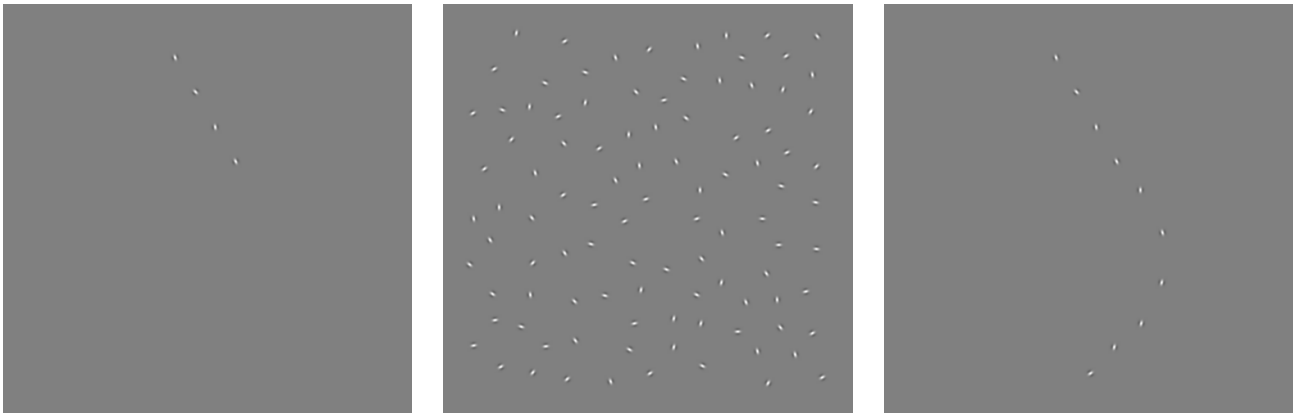
Target alignment | Stimulus (input) | Detection: NFA = 2.0

Figure 17: Example of detection in a stimulus composed of $N = 600$ patches, with $n = 6$ elements affected by a jitter $\alpha = \frac{\pi}{4}$. In this case the algorithm prefers a short alignment, although the target alignment seems more salient.



Target alignment | Stimulus (input) | Detection: NFA = 0.14

Figure 18: Example of detection in a stimulus composed of $N = 200$ patches, with $n = 8$ elements affected by a jitter $\alpha = \frac{\pi}{4}$. Because it is quite long and not so affected by angular jitter the target is expected to be easy to spot. However in this stimulus, other alignment-like structures occurred by chance closer to the image center, which tend to draw our attention away from the target alignment.



Target alignment | Stimulus (input) | Detection: NFA = 0.21

Figure 19: Example of detection in a stimulus composed of $N = 100$ patches, with $n = 4$ elements affected by a jitter $\alpha = \frac{\pi}{5}$. Here the algorithm detected a longer chain than the target. The NFA associated to the target alignment composed of 4 elements is NFA = 7.3.

Algorithm 2: Detection algorithm

input : N points $\mathbf{x} = \{x_i\}_{i=1\dots N}$ in $\mathbb{R}^2$, N angles $\mathbf{\Theta} = \{\theta_i\}_{i=1\dots N}$ in $[0, 2\pi)$. Each angle θ_i is supposed to be the orientation of a Gabor element placed at x_i.

output: The smallest NFA value NFA^*, and the associated set of Gabor elements $\{(x_i, \theta_i), i \in C^*\}$, where C^* is a subset of $\{1, \dots, N\}$

$\gamma \leftarrow$ `computeGamma`$(\mathbf{x})$
$\text{NFA}^* \leftarrow +\infty$
$C^* \leftarrow \emptyset$
for $i = 1$ **to** N **do**
  // $\mathcal{N}_\gamma(i)$ is the set of indexes of x_i's neighbors in γ
  for $j \in \mathcal{N}_\gamma(i)$ **do**
    $C \leftarrow (i, j)$
    $lastElt \leftarrow j$
    while $\#C < \lfloor \sqrt{N} \rfloor$ *and* $lastElt \neq \emptyset$ **do**
      $lastElt \leftarrow$ `findNextPoint`$(\mathbf{x}, \gamma, C)$
      $C \leftarrow C \cup lastElt$
      $\text{NFA} \leftarrow$ `nfa`$(C, \mathbf{x}, \mathbf{\Theta})$
      if $\text{NFA} < \text{NFA}^*$ **then**
        $\text{NFA}^* \leftarrow \text{NFA}$
        $C^* \leftarrow C$
      end
    end
  end
end
return $\text{NFA}^*, \{(x_i, \theta_i), i \in C^*\}$

Function computeGamma($\mathbf{x}$). The input $\mathbf{x} = \{x_i\}_{i=1...N}$ is a set of N points in $\mathbb{R}^2$, and the function returns an adjacency matrix γ such that $\gamma(i,j) = 1$ if there is an arrow from x_i to x_j, and $\gamma(i,j) = 0$ otherwise.

$minDistances \leftarrow \emptyset$
for $i = 1$ **to** N **do**
  $d_i \leftarrow \emptyset$
  for $j = 1$ **to** N **do**
    $d_i \leftarrow d_i \cup d(x_i, x_j)$ // *Euclidean distance*
  end
  $(d_i, idx_i) \leftarrow sort(d_i)$ // *sort in ascending order; idx_i contains the new order of the points indexes*
  $minDistances \leftarrow minDistances \cup d_i(2)$ // *$d_i(2)$ is the second element in d_i*
end
$d_{avg} = mean(minDistances)$ // *mean value of the set minDistances*
$\gamma \leftarrow zeros(N, N)$
for $i = 1$ **to** N **do**
  for $j = 2$ **to** 7 **do**
    if $d_i(j) \leq 2d_{avg}$ **then**
      $\gamma(i, idx_i(j)) = 1$
    end
  end
end
return γ;

Function findNextPoint($\mathbf{x}, \gamma, c$). $\mathbf{x} = \{x_i\}_{i=1\dots N}$ is a set of N points in $\mathbb{R}^2$, γ a $N \times N$ adjacency matrix representing an oriented graph on $\mathbf{x}$, and c an ordered list of indexes, corresponding to a chain of points of $\mathbf{x}$. The function returns an index in $\{1, \dots, N\}$, which corresponds to the point which should expand the current chain c.

$i_1 \leftarrow last(c)$
$i_2 \leftarrow penultimate(c)$
$p_1 \leftarrow x_{i_1}$
$p_2 \leftarrow x_{i_2}$
$i^* \leftarrow \emptyset$
$maxCosine \leftarrow -\infty$
for $i_3 \in \mathcal{N}_\gamma(i_2)$ **do**
  $p_3 \leftarrow x_{i_3}$
  $currentCosine \leftarrow \frac{\overrightarrow{p_1p_2} \cdot \overrightarrow{p_2p_3}}{p_1p_2 \cdot p_2p_3}$
  if $currentCosine > maxCosine$ **then**
    $maxCosine \leftarrow currentCosine$
    $i^* \leftarrow i_3$
  end
end
if $i^* \notin c$ **then**
  $nextPoint \leftarrow i^*$
else
  $nextPoint \leftarrow \emptyset$
end
return $nextPoint$

Function nfa($C, \mathbf{x}, \mathbf{\Theta}$). $\mathbf{x} = \{x_i\}_{i=1\dots N}$ is a set of N points in $\mathbb{R}^2$, $\mathbf{\Theta} = \{\theta_i\}_{i=1\dots N}$ a set of N angles in $[0, 2\pi)$, and $C = (i_1, i_2, \dots, i_n)$ a tuple of n indexes in $\{1, \dots, N\}$.

$p_1 \leftarrow x_{i_1}$
$p_2 \leftarrow x_{i_2}$
$p_{n-1} \leftarrow x_{i_{n-1}}$
$p_n \leftarrow x_n$
$\omega_1 \leftarrow |\texttt{angularError}(p_1, p_2, \theta_{i_1})|$
$\omega_n \leftarrow |\texttt{angularError}(p_{n-1}, p_n, \theta_{i_n})|$
for $k = 2$ **to** $n - 1$ **do**
  $p_1 \leftarrow x_{i_{k-1}}$
  $p_2 \leftarrow x_{i_k}$
  $p_3 \leftarrow x_{i_{k+1}}$
  $\alpha_L \leftarrow |\texttt{angularError}(p_1, p_2, \theta_{i_k})|$
  $\alpha_R \leftarrow |\texttt{angularError}(p_2, p_3, \theta_{i_k})|$
  $\omega_k \leftarrow \max(\alpha_{iL}, \alpha_{iR})$
end
$\omega^* \leftarrow \max\{\omega_1, \omega_2, \dots, \omega_n\}$
$N_T \leftarrow 6 \times N \times \sqrt{N}$ // *(Equation (23))*
$p \leftarrow \left(\frac{\pi\omega^*}{2}\right)^2 \times F(\omega^*)^{n-2}$ // *(Equation (24) and (25))*
$nfa \leftarrow N_T \cdot p$ // *(eq 26)*
return nfa

Function angularError(p_1, p_2, θ). p_1 and p_2 are two points in $\mathbb{R}^2$ whose abscissa and ordinates are noted $p_i.x$ and $p_i.y$; θ is an angle in $[0, 2\pi)$.

// First compute the orientation of line (p_1p_2), noted θ_{12}, in $(-\frac{\pi}{2}, \frac{\pi}{2}]$
if $p_1.x \neq p_2.x$ **then**
  $\theta_{12} \leftarrow \arctan\left(\frac{p_1.y - p_2.y}{p_1.x - p_2.x}\right)$
else
  $\theta_{12} \leftarrow \frac{\pi}{2}$
end
$\Delta_{ang} \leftarrow \arctan(\tan(\theta_{12} - \theta))$ *// With the convention:* $\tan(\pm\frac{\pi}{2}) = \pm\infty$ *and* $\arctan(\pm\infty) = \pm\frac{\pi}{2}$
return Δ_{ang}

4 Conclusion

This paper has presented a method aiming at interpreting and quantifying the perceptual grouping of aligned oriented elements in arrays of Gabor patches. The method is composed of two main algorithms: first, a generation algorithm designed to build stimuli in which perceptual grouping is almost only due to the orientations of the patches; second, a detection algorithm designed to find the target alignments only when they are conspicuous for human visual perception. This parameterless algorithm is based on the *a contrario* theory, a mathematical model of the non-accidentalness principle. These methods were applied and tested in psychophysical experiments [2], where the NFA showed strong correlation with the detectability for human subjects, and the algorithm proved to be an accurate predictor of the average subject's behavior. Therefore, the framework presented here is a tool to test the relevance of the *a contrario* approach to perceptual tasks in image processing and computer vision.

Appendix

To analyze the distribution of the random variables Ω_i we start by analyzing the possible values ω when the chain at point i has an angle between the previous (L) and next (R) element of φ, see Figure 20.

Given that ω is the largest of α_L and α_R, for a given φ the possible orientations for the Gabor element are classified into four sectors, in two of them $\omega = \alpha_L$ and in the other two $\omega = \alpha_R$. Indeed, at the orientation of the bisector of the two lines L and R, $\alpha_L = \alpha_R$. The same happens at the bisector of R and the continuation of L and inversely.

Let us analyze the distribution in one of the sectors; the same distribution is observed in the three others and the reader can fill the details. The minimal possible value for ω is $\frac{\varphi}{2}$, as can be seen at the orientation marked with the letter A in Figure 21. Indeed, that orientation is the bisector of line R and the continuation of line L, which form an angle φ. As the orientation increases clockwise, the corresponding angle ω increases gradually, passing the orientation marked with the letter B, where $\omega = \frac{\pi}{2} - \frac{\varphi}{2}$, until arriving at the orientation C, which is orthogonal to line L and gives the maximal possible value $\omega = \frac{\pi}{2}$. Then, as the orientation goes still clockwise to the orientation D, the ω decreases gradually to angle $\frac{\pi}{2} - \frac{\varphi}{2}$, which is the angle of the bisector of lines L and R.

Since the orientation of the Gabor element is uniformly distributed, the values for ω are equiprobable in $[\frac{\varphi}{2}, \frac{\pi}{2} - \frac{\varphi}{2}]$, and so are the values in $[\frac{\pi}{2} - \frac{\varphi}{2}, \frac{\pi}{2}]$. The probability density function for ω, given a fixed φ can therefore be written $p(\omega|\varphi) = a\mathbb{1}_{[\frac{\varphi}{2}, \frac{\pi}{2} - \frac{\varphi}{2})}(\omega) + b\mathbb{1}_{[\frac{\pi}{2} - \frac{\varphi}{2}, \frac{\pi}{2}]}(\omega)$.

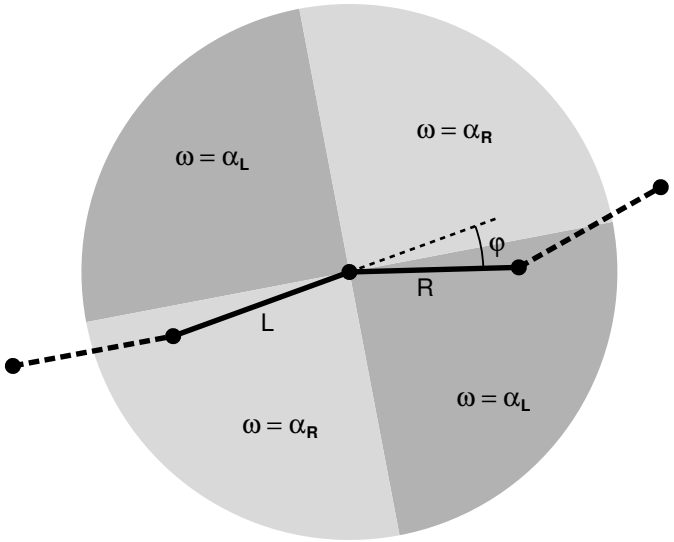


Figure 20: For a fixed φ, depending on the orientation of the Gabor element, the angle $\omega = \max(\alpha_L, \alpha_R)$ is equal to α_L or α_R. Here we represent the different cases according to the orientation of the Gabor element.

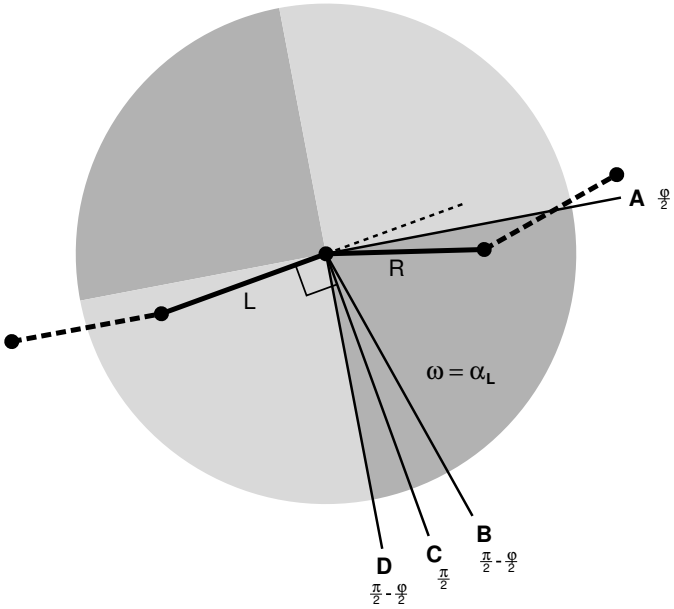


Figure 21: Analysis of the distribution of ω when the Gabor element is oriented in the bottom right sector. The letters A, B, C and D mark four key orientations in the analysis. The angular values reported next to them are the corresponding values for ω, and not the orientation of the Gabor element.

Also, the probability that ω be in the first interval is equal to the probability that the Gabor element be oriented between A and B, that is $\frac{2}{\pi}(\frac{\pi}{2} - \varphi)$; similarly, the probability that ω be in the second interval is equal to the probability that the Gabor element be oriented between B and D, that is $\frac{2\varphi}{\pi}$. This allows to deduce the values of a and b, which gives the following probability density function for ω, given a fixed φ

$$p(\omega|\varphi) = \frac{2}{\pi}\mathbb{1}_{[\frac{\varphi}{2}, \frac{\pi}{2} - \frac{\varphi}{2})}(\omega) + \frac{4}{\pi}\mathbb{1}_{[\frac{\pi}{2} - \frac{\varphi}{2}, \frac{\pi}{2}]}(\omega), \tag{27}$$

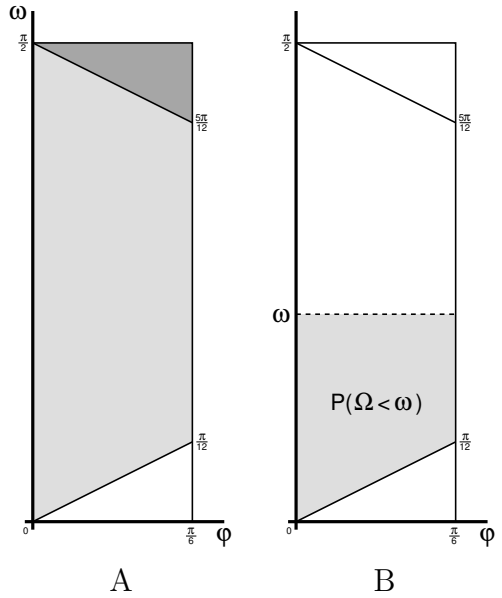
and the corresponding cumulative distribution function is then

$$\mathbb{P}(\Omega \leq \omega|\varphi) = \begin{cases} 0 & \text{if } \omega < \frac{\varphi}{2} \\ \frac{2}{\pi}(\omega - \frac{\varphi}{2}) & \text{if } \frac{\varphi}{2} \leq \omega < \frac{\pi}{2} - \frac{\varphi}{2} \\ \frac{4\omega}{\pi} - 1 & \text{if } \frac{\pi}{2} - \frac{\varphi}{2} \leq \omega \leq \frac{\pi}{2}. \end{cases} \tag{28}$$

Equations (27) and (28) can be represented as a vertical slice at a given value of φ, in Figure 22 A and B respectively.

We are interested in the cumulative distribution $F(\omega) = \mathbb{P}(\Omega \leq \omega)$ which results from integrating Equation (28) with respect to φ between 0 and $\frac{\pi}{6}$. The result is divided into three cases corresponding to the three parts of the distribution as observed in Figure 22 B, for $\omega < \frac{\pi}{12}$, for $\frac{\pi}{12} \leq \omega < \frac{5\pi}{12}$, and for $\frac{5\pi}{12} \leq \omega$. A short computation gives the result

$$F(\omega) = \mathbb{P}(\Omega \leq \omega) = \begin{cases} \frac{12}{\pi^2}\omega^2 & \text{if } 0 \leq \omega < \frac{\pi}{12} \\ \frac{2}{\pi}\omega - \frac{1}{12} & \text{if } \frac{\pi}{12} \leq \omega < \frac{5\pi}{12} \\ \frac{12}{\pi^2}\omega^2 - \frac{8}{\pi}\omega + 2 & \text{if } \frac{5\pi}{12} \leq \omega \leq \frac{\pi}{2}. \end{cases} \tag{29}$$



A B

Figure 22: A: Representation of the joint distribution of φ and ω, given that φ is uniformly distributed in $[0, \frac{\pi}{6}]$. Three regions are defined by the lines of equation $\omega = \frac{\varphi}{2}$ and $\omega = \frac{\pi}{2} - \frac{\varphi}{2}$. B: The corresponding cumulative distribution function.

Acknowledgments

We thank the anonymous reviewers for valuable suggestions. Work partly founded by Centre National d'Etudes Spatiales (CNES, MISS Project), European Research Council (advanced grant Twelve Labours), Office of Naval research (ONR grant N00014-14-1-0023), DGA Stéréo project, ANR-DGA (project ANR-12-ASTR-0035), FUI (project Plein Phare) and Institut Universitaire de France.

References

[1] F. Attneave, *Some informational aspects of visual perception.*, Psychological Review, 61 (1954), pp. 183–193. http://dx.doi.org/10.1037/h0054663.

[2] S. Blusseau, A. Carboni, A. Maiche, J.M. Morel, and R. Grompone von Gioi, *Measuring the visual salience of alignments by their non-accidentalness*, Vision Research, 126 (2016), pp. 192 – 206. Quantitative Approaches in Gestalt Perception. http://dx.doi.org/10.1016/j.visres.2015.08.014.

[3] M. Demeyer and B. Machilsen, *The construction of perceptual grouping displays using GERT.*, Behavior Research Methods, online first., (2011), pp. 1–8. http://dx.doi.org/10.3758/s13428-011-0167-8.

[4] A. Desolneux, L. Moisan, and J.M. Morel, *From Gestalt Theory to Image Analysis, a Probabilistic Approach*, vol. 34 of Interdisciplinary Applied Mathematics, Springer, 2008. ISBN 978-0-387-74378-3, http://dx.doi.org/10.1007/978-0-387-74378-3.

[5] W.D. Ellis, ed., *A Source Book of Gestalt Psychology*, Humanities Press, 1967 (originally 1938). ISBN 093926630X, http://dx.doi.org/10.1037/11496-000.

[6] D. J. Field, A. Hayes, and R. F. Hess, *Contour integration by the human visual system: Evidence for a local association field*, Vision Research, 33 (1993), pp. 173 – 193. http://dx.doi.org/10.1016/0042-6989(93)90156-Q.

[7] R. Grompone von Gioi, J. Jakubowicz, J.-M Morel, and G. Randall, *LSD: A fast line segment detector with a false detection control*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (2010), pp. 722–732. http://dx.doi.org/10.1109/TPAMI.2008.300.

[8] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, *LSD: a line segment detector*, Image Processing On Line, (2012). http://dx.doi.org/10.5201/ipol.2012.gjmr-lsd.

[9] G. Kanizsa, *Organization in vision: Essays on Gestalt perception*, Praeger New York:, 1979. ISBN 978-0275903732.

[10] ——, *Grammatica del vedere*, Il Mulino, 1980. ISBN 978-8815060907.

[11] W. Köhler, *Gestalt Psychology*, Liveright, 1947. ISBN 978-0871402189.

[12] I. Kovacs and B. Julesz, *A closed curve is much more than an incomplete one: Effect of closure in figure-ground segmentation*, Proceedings of the National Academy of Sciences, 90 (1993), pp. 7495–7497.

[13] T. Ledgeway, R.F. Hess, and W. S Geisler, *Grouping local orientation and direction signals to extract spatial contours: Empirical tests of association field models of contour integration*, Vision research, 45 (2005), pp. 2511–2522. http://dx.doi.org/10.1016/j.visres.2005.04.002.

[14] D. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985. ISBN 978-1-4612-9604-1, http://dx.doi.org/10.1007/978-1-4613-2551-2.

[15] B. Machilsen, M. Pauwels, and J. Wagemans, *The role of vertical mirror symmetry in visual shape detection*, Journal of Vision, 9 (2009), p. 11. http://dx.doi.org/10.1167/9.12.11.

[16] B. Machilsen and J. Wagemans, *Integration of contour and surface information in shape detection*, Vision Research, 51 (2011), pp. 179–186. http://dx.doi.org/10.1016/j.visres.2010.11.005.

[17] B. Machilsen, J. Wagemans, and M. Demeyer, *Quantifying density cues in grouping displays*, Vision Research, 126 (2016), pp. 207 – 219. http://dx.doi.org/10.1016/j.visres.2015.06.004.

[18] D. Marr, *Vision*, Freeman and co., 1982. ISBN 9780262514620.

[19] W. Metzger, *Gesetze des Sehens*, Verlag Waldemar Kramer, 3rd ed., 1975. ISBN 3880744920.

[20] G.E. Nygård, T. Van Looy, and J. Wagemans, *The influence of orientation jitter and motion on contour saliency and object identification*, Vision Research, 49 (2009), pp. 2475–2484. http://dx.doi.org/10.1016/j.visres.2009.08.002.

[21] S.E. Palmer, *Perceptual organization in vision*, Stevens' handbook of experimental psychology, (2002). http://dx.doi.org/10.1002/0471214426.pas0105.

[22] S. E. Palmer, *Vision Science: Photons to Phenomenology*, The MIT Press, 1999. ISBN 9780262161831.

[23] M. Sassi, M. Demeyer, and J. Wagemans, *Peripheral contour grouping and saccade targeting: The role of mirror symmetry*, Symmetry, 6 (2014), pp. 1–22. http://dx.doi.org/10.3390/sym6010001.

[24] S.S. Stevens, *Psychophysics*, Transaction Publishers, 1986. ISBN 9781412832335.

[25] K. Vancleef and J. Wagemans, *Component processes in contour integration: A direct comparison between snakes and ladders in a detection and a shape discrimination task*, Vision research, 92 (2013), pp. 39–46. http://dx.doi.org/10.1016/j.visres.2013.09.003.

[26] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, *A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization.*, Psychological bulletin, 138 (2012), pp. 1172–1217. http://dx.doi.org/10.1037/a0029333.

[27] J. Wagemans, J. Feldman, S. Gepshtein, R. Kimchi, J. R. Pomerantz, P. A. van der Helm, and C. van Leeuwen, *A century of Gestalt psychology in visual perception: II. Conceptual and theoretical foundations.*, Psychological Bulletin, 138 (2012), pp. 1218–1252. http://dx.doi.org/10.1037/a0029334.

[28] M. Wertheimer, *Untersuchungen zur Lehre von der Gestalt. II*, Psychologische Forschung, 4 (1923), pp. 301–350. An abridged translation to English is included in [5]. http://dx.doi.org/10.1007/BF00410640.

[29] A.P. Witkin and J.M. Tenenbaum, *What is perceptual organization for?*, IJCAI-83, 2 (1983), pp. 1023–1026. http://dl.acm.org/citation.cfm?id=1623516.1623608.

[30] A.P. Witkin and J. M. Tenenbaum, *Human and Machine Vision*, Academic Press, 1983, ch. On the role of structure in vision, pp. 481–543. ISBN 978-0-12-084320-6.