Chapter 2
A Contrario Detection

At the turn of the century, Desolneux, Moisan, and Morel undertook the task of formalizing the Gestalt theory into a mathematical framework [19, 22]. The motivation for this ambitious project was to provide a foundation for computer vision based, like the Gestalt theory [46, 51, 61, 62], on a small set of fundamental principles. They identified the lack of a principle to guide the selection of detection thresholds and their main contribution was to propose the *a contrario* framework to cover this need. This chapter will introduce the *a contrario* approach and its application to line segment detection.

The framework takes its name from the Latin expression "*a contrario*" which means *by or from contraries*. Accordingly, the meaning of a structure is derived from its *contrary*, the lack of structure: a stochastic model for unstructured data. The theory is actually a formalization of the non-accidentalness principle discussed in Sect. 1.3 (Desolneux et al. refer to it as *Helmholtz principle*). One would like to preclude detections on unstructured data. Yet, this is not possible: even if rarely, any structure *could* be observed in random data. The *a contrario* approach stipulates instead to set the detection thresholds in order to bound the *average* number of detections one would get on unstructured data. Figure 2.1 illustrates the idea.

We will start by an abstract presentation of these ideas, which should become more clear when illustrated in the next section by a simple example. Let us denote by H_0 the stochastic model for unstructured data. In our framework, the quality of a detection method will be evaluated relative to the *a contrario* model H_0. When the method is applied to unstructured data from H_0, every detection is a false detection. The mean number of detection events observed on data from H_0 corresponds then to the false detection rate of the method, and is the quantity to be bounded.

Any method will produce detections for certain configurations in the input data. We will call $\mathscr{E}$ the set of all possible detection events of a method: that is, all possible positions where the particular structure can be observed, together with the corresponding pixel configuration that makes it a valid detection. Every detection method defines, explicitly or implicitly, a set $\mathscr{E}$. The expected number of events from $\mathscr{E}$ in H_0 corresponds then to the expected number of false detections. Thus, a

R. Grompone von Gioi, *A Contrario Line Segment Detection*, SpringerBriefs in Computer Science, DOI 10.1007/978-1-4939-0575-1_2, © The Author(s) 2014

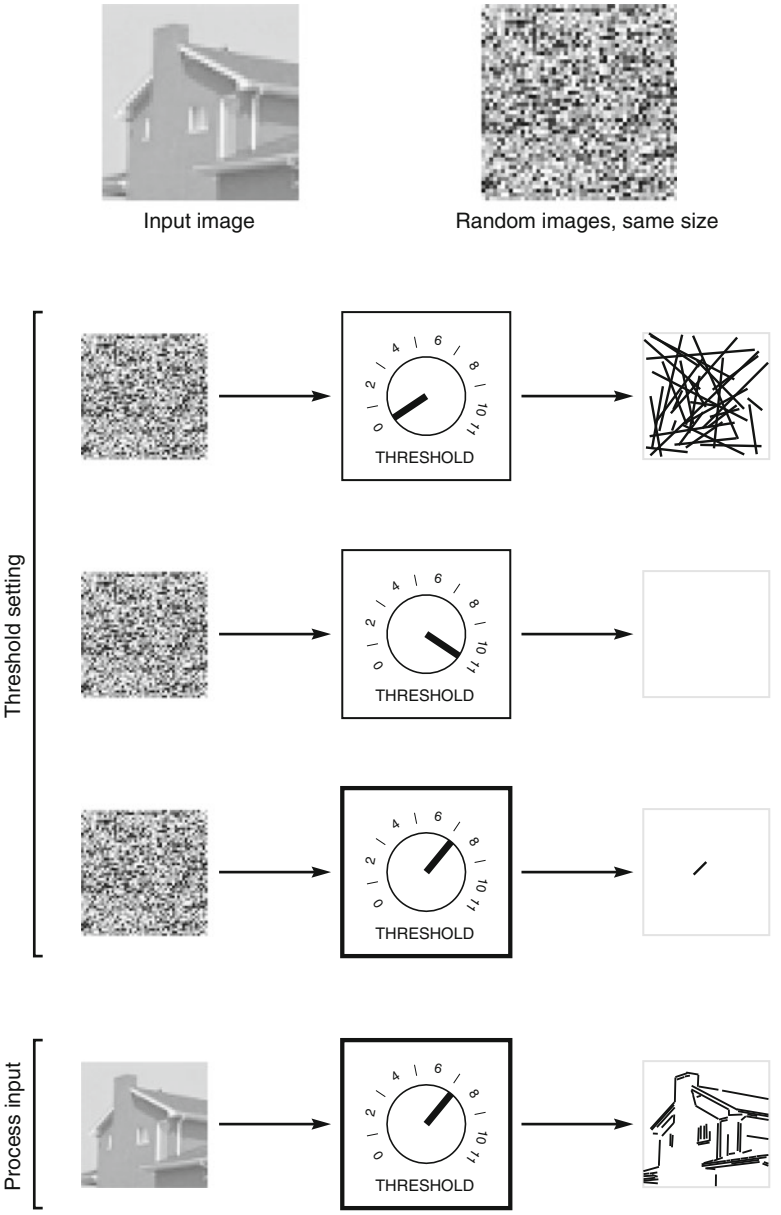

Fig. 2.1 An intuitive illustration of the *a contrario* approach. Different threshold values are tried in random data of the same size as the input. A low detection threshold produces a large number of detections in random data, all of them accidental (*first row*). A high threshold rejects accidental detections, but never produces detections at all (*second row*). The *a contrario* approach suggests selecting thresholds that produce, on average, a fixed number of detections in random data (one in this case, third row). This is restrictive enough to reject most detections by chance, allowing as many good detections as possible. Once the threshold is selected with random data, the algorithm is applied to the input. For simple problems as line segment detection, it is possible to compute the right thresholds analytically, without the need to simulate random data

detection method is said to define ε-meaningful events (or to produce ε-meaningful detections), when the expected number of false detections is bounded by ε.

The *a contrario* statistical framework rests on the definition of ε-meaningful events, relative to a stochastic model H_0 for unstructured data:

Definition 1. The events of a set $\mathscr{E}$ are called ε-meaningful events in H_0 if the *expected* total number of these events observed under H_0 is less than or equal to ε.

As we will see, setting the constant ε to a small fixed value results in an effective control of the number of false detections.

The *a contrario* framework is closely related to the theory of statistical analysis. The problem is formulated as a set of statistical tests in the spirit of Fisherian statistics [16, 41], but including corrections for *multiple comparisons* [38, 39]; see [22, section 15.3.2] and [67, section 4.4] for a comparison. A similar framework was proposed by Dembski [18] for a different problem: the detection of "intelligent causes" relative to "natural causes". This chapter is a brief introduction to the *a contrario* approach; for more details refer to [22].

2.1 A Simple Example

Let us consider a sequence of coin flips: a random sequence of equally likely heads and tails. It is not surprising to observe a sequence of three consecutive heads. Actually, this event should appear once every eight sets of three coin tosses. Now, imagine that a sequence of 100 heads is observed. That would be an extremely unlikely event with a fair coin. Even if everyone in the world made a flip per second, it would take about 1,000 billion years before someone could get that result just by chance. Observing such a sequence would be an overwhelmingly lucky event, and it is more reasonable to look for a causal connection of some kind. However, following the non-accidentalness principle, *only* events that cannot be explained by chance require a causal explanation.

There is another important point to reflect on. Let `1` be heads and `0` tails. Then, the following sequence represents a possible result of 50 coin tosses:

`11111111111111111111111111111111111111111111111111.`

Such a sequence of 50 consecutive heads is very structured and unexpected. The probability of observing this sequence with a fair coin is 2^{-50}. In contrast, the following one seems unstructured:

`01010010001000011000001100110001100111010011001000.`

Actually, the probability of observing *exactly* the latter sequence is the same as before, 2^{-50}, as it is given by the product of the probabilities for each individual equi-probable event: getting a tail in the first toss (1/2), getting a head on the second toss (1/2), a tail in the third one (1/2), and so on. Both sequences are hence equally

likely (or unlikely). What makes the former a relevant event cannot be *only* its small probability. While the first sequence has a certain regularity, a certain order that makes it particular, the second one has no particular reason to be differentiated from, for example,

`00110011100110110100100100010100011010100000111100`.

Dembski [18], discussing the problem of the identification of intelligent causes, states that a relevant event must have a small probability and also be *specified*, meaning that it should conform to a previously defined pattern. To illustrate the point, Dembski suggests imagining an archer shooting arrows at a target painted on the wall; the arrow hitting the center of the target would be a relevant event. However, if the target is painted on the wall around the arrow *after* shooting, the event would not be relevant at all. Analogously, a relevant pattern must be defined in advance—for example, a sequence of n heads—and not be selected as any particular observed pattern that happens with low probability.

Getting back to the *a contrario* approach, two parts need to be defined: an observing method to measure the structure, and a model for unstructured data. The former specifies the structure (the pattern) in advance; in the previous analogy, it corresponds to drawing the target for the archer. The latter tells us what to expect in a random case; think of a blind archer. The *a contrario* approach advises to apply thresholds on the observations and to choose its values to allow, on average, only ε detections on unstructured data. This results in ε-meaningful detections in agreement with definition 1. The smaller the value of ε, the less number of false detections accepted, and the more strict the conditions imposed on the structure to be detected. To continue with the previous analogy, the smaller the ε, the smaller the target where the archer is aiming and the more meaningful if the arrow hits it.

Consider a simple toy detection problem: we want to detect when someone is pressing a certain button. Imagine there is a mechanism that measures the state of the button, giving the value `1` when pressed and `0` when released. In an ideal case, an observed sequence would be as follows,

`00000000000111111111111111000000000000000000000000`,

indicating that the button was first released, then pressed, and finally released again. Unfortunately, our systems is corrupted by noise and a typical observation looks more like:

`01001001000111111111111111100100001011001100011100`.

To simplify things, let us assume that noise only affects released buttons, but a pressed button always produces the value `1`. Judging the button as pressed whenever a `1` is observed would lead to many false detections. However, a long enough run of `1`s would certainly correspond to a pressed button. How much is *long enough*? The *a contrario* approach proposes to define the threshold in order to control the number of false detections due to noise.

Let us formalize the problem. We observe some data x, a binary sequence in our example, and we want to detect a given structure: an exceptional long run of consecutive `1`s. The structure can be located at various positions and each one is a candidate that must be tested; a family of N_C candidates $\mathscr{C} = \{c_1, \ldots, c_{N_C}\}$ is defined. We will call N_C the number of tests. In our case, runs of `1`s can start at any element of the binary sequence; thus, N_C is equal to the length L of the binary sequence x. (We ignore the fact that the run in the last position has a length of at most one element.) We can now define the observing method: a function $k(c_i, x)$ to measure the degree to which the sought structure is present in data x at candidate position c_i. For our problem, the value $k(c_i, x)$ is the length of the run of consecutive `1`s starting at position c_i.

The second part to be defined is a stochastic model H_0 for unstructured binary sequences. The simplest choice for the present example is the classic model of fair coin tosses: independent flips with equal probability for heads and tails.

To complete the detection procedure we need to specify thresholds κ_i so that a candidate c_i is validated as a detection in data x when $k(c_i, x) \geq \kappa_i$. Given an observation x, the total number of detections is then

$$d(x) = \sum_{i=1}^{N_C} \mathbb{1}_{k(c_i,x) \geq \kappa_i} ,$$

where $\mathbb{1}_A$ is the indicator function, taking the value 1 in the set (or event) A and zero otherwise. The *a contrario* approach requires that the thresholds κ_i be set so that the expected number of detections $d(X)$, when X is random data from the model H_0, be less than or equal to ε. Since there are many κ_i and a single ε, there are many ways to achieve this adjustment. A common one is the so-called "Bonferroni correction" [39]. It consists in dividing up ε in equal parts among the tests, which amounts to authorize the same number of false detections per candidate. This comes to setting

$$\kappa_i = \min \left\{ n, \mathbb{P}[k(c_i, X) \geq n] \leq \frac{\varepsilon}{N_C} \right\} . \tag{2.1}$$

With the thresholds κ_i defined this way, we can show that the expected number of detections in H_0 is controlled by ε and thus the detections are ε-meaningful events:

Proposition 1. *The family $\mathscr{E}$ of N_C events $\{k(c_i, X) \geq \kappa_i\}$, with κ_i defined as in (2.1), are ε-meaningful events when X is data from H_0.*

Proof. The expected number of events observed is

$$\mathbb{E}[d(X)] = \mathbb{E}\left[\sum_{i=1}^{N_C} \mathbb{1}_{k(c_i,X) \geq \kappa_i} \right] = \sum_{i=1}^{N_C} \mathbb{P}[k(c_i, X) \geq \kappa_i] ,$$

where $\mathbb{E}$ is the expectation operator and X is random data drawn from H_0. But by definition of κ_i we know that

$$\mathbb{P}[k(c_i, X) \geq \kappa_i] \leq \frac{\varepsilon}{N_C} .$$

Then,

$$\mathbb{E}[d(X)] \leq \sum_{i=1}^{N_C} \frac{\varepsilon}{N_C} = \varepsilon \ ,$$

which concludes the proof. □

In the *a contrario* framework a specific quantity, $\mathrm{NFA}(c_i,x)$, is associated to the candidate c_i observed in data x. It is defined as the number of tests considered, times the probability of observing events at least as meaningful in the *a contrario* model H_0:

$$\mathrm{NFA}(c_i,x) = N_C \cdot \mathbb{P}[k(c_i,X) \geq k(c_i,x)] \ . \tag{2.2}$$

Note that the larger the observed measure $k(c_i,x)$, the lower the probability that $k(c_i,X) \geq k(c_i,x)$ in H_0; therefore, the smaller the NFA value, and the more meaningful the observation is. This quantity was devised to provide a handy test to validate events: an observation (c_i,x) is ε-meaningful if and only if $\mathrm{NFA}(c_i,x) \leq \varepsilon$. It is easy to check that this is the case since this test is equivalent to

$$\mathbb{P}[k(c_i,X) \geq k(c_i,x)] \leq \frac{\varepsilon}{N_C} \ ,$$

which implies, in turn, that $k(c_i,x) \geq \kappa_i$, as is required for ε-meaningfulness.

The name NFA stands for *Number of False Alarms* and is justified by the following fact: a candidate c_i in data x with $\mathrm{NFA}(c_i,x) = \alpha$ will be validated as a detection only if ε is larger than or equal to α, and in such case the method will get at least α false detections. In other words, $\mathrm{NFA}(c_i,x)$ corresponds to the false detection level needed to accept the event.

The constant ε corresponds to the expected number of false detections that one is ready to accept, which depends on the task being handled. If we are trying to detect an event that is rarely observed, accepting one false detection on average would make the detector useless: In such a case, the expected number of detections would be, numerically, very similar to the probability of a false detection, and the *a contrario* framework would be less useful. This framework is well adapted for the case when many detections are made in a typical case and thus accepting a few false detections is not a problem. In pictures of artificial environments, for example, one would expect many line segment detections, as it is shown in Fig. 1.1; to get *one* false detection among them is not a real problem. It is important, however, that the detector keeps the number of false detections controlled to a fixed value ε for different data sets, preventing the explosion of false detections when the data size grows. What value should be used for ε? The thresholds κ_i show a slow dependency on ε, which implies that the resulting method is not very dependent on the particular ε value selected. Any small value is usually fine. As a simple convention, Desolneux et al. [19, 22] suggest using $\varepsilon = 1$. We will see some examples in the next section.

Let us complete the formulation for our example. We need to compute the probability $\mathbb{P}[k(c_i,X) \geq n]$ of observing a run of at least n consecutive `1`s starting at position c_i. Given the independence of the digits in binary sequences under H_0, that

quantity is equal to the probability that the first n digits, starting at c_i, are all `1`. (The following digits could include more consecutive `1`s, increasing the length of the run, or not.) We have

$$\mathbb{P}[k(c_i,X) \geq n] = p^n \ ,$$

where p is the probability of getting a `1` under H_0. As computed before, the number of tests N_C is the length L of the binary sequence. Then, the NFA value of an observed run of length $k(c_i,x)$ is given by

$$\mathrm{NFA}(c_i,x) = L\,p^{k(c_i,x)} \ .$$

For sequences of total length 50, as the previous ones, and with $p = 1/2$, the NFA value of a run of length n is given by 50×2^{-n}. Table 2.1 shows some numerical values. As one could anticipate, we will observe on average 25 runs of at least one element, which is not a meaningful event. A similar conclusion is made for runs of two, three, or four elements. Runs of at least six consecutive `1`s are more meaningful, as it is not expected to be present in every data set from H_0. The meaningfulness grows with the length of the run. With $\varepsilon = 1$, runs of length six or more are considered detections. This threshold will produce no detection in a typical random binary sequence of length 50,

`11101111101000111001000110100101111011010100001000`,

whereas when meaningful events are present, they will be detected,

`00101111111111111010000100110100111111111100100011`.

Setting $\varepsilon = 1$ implies getting, on average, one false detection per sequence; that is, one detection per random binary sequence from H_0. It may seem futile to accept one false detection for such a short sequence, but the strength of the method lies in how it adapts the thresholds automatically to larger data sets. If we now observe a sequence of one million digits, $L = 10^6$, the limit case for $\varepsilon = 1$ is runs of 20 heads with an NFA value of 0.95. We will get all the meaningful runs and still get at most *one* false detection in sequences of one million digits. The limit run length is given by

$$n_{\mathrm{th}} = \left\lceil \frac{\log \varepsilon - \log N_C}{\log p} \right\rceil \ ,$$

where $\lceil x \rceil$ is the smallest integer not less than x. The dependencies on ε and on the number of tests N_C are opposite and both logarithmic, which implies a slight dependency. This is usually the case for *a contrario* formulations.

Table 2.1 NFA values for runs of n consecutive `1`s on a binary sequence of total length 50

n	NFA	n	NFA	n	NFA	n	NFA
1	25	6	0.78	11	$2 \cdot 10^{-2}$	16	$8 \cdot 10^{-4}$
2	12.5	7	0.39	12	$1 \cdot 10^{-2}$	17	$4 \cdot 10^{-4}$
3	6.25	8	0.195	13	$6 \cdot 10^{-3}$	18	$2 \cdot 10^{-4}$
4	3.13	9	0.098	14	$3 \cdot 10^{-3}$	19	$9 \cdot 10^{-5}$
5	1.56	10	0.049	15	$1 \cdot 10^{-3}$	20	$5 \cdot 10^{-5}$

2.2 *A Contrario* Line Segment Detection

Let us see now how these ideas apply to a detection problem in computer vision. We will follow Desolneux, Moisan, and Morel [19] and introduce their line segment detector. We are looking for locally straight edges in images. This can be done by testing all possible line segments and for each one checking if there is an edge that roughly follows it. We will use the simplest of image indicators that provides information about edges: the image gradient.

A digital image x is defined as a grid of $N \times M$ pixels, each one with a numerical value representing its brightness: small values correspond to dark pixels and large values to bright ones. In a common representation, pixels take values in the range $[0, 255]$, zero corresponding to black and 255 to white. The gradient of an image at a certain pixel is a vector pointing in the direction of maximum increase of image brightness, see Fig. 2.2. The gradient is actually locally orthogonal to the image level-lines; that is, lines where the image brightness is constant. When an edge is present at a given pixel, the level-line angle roughly follows the edge and the gradient vector is roughly orthogonal to the edge, see Fig. 2.2. Computing the local level-line angle at each pixel results in a *level-line field*, see Fig. 2.3, whose elements are well structured along edges. Notice that the level-line elements are oriented, implicitly indicating which side of the level-line is darker. In what follows, $\lambda(u, v)$ will denote the (oriented) level-line angle at coordinates (u, v). The gradient magnitude gives additional information about the strength of the edge, but this information will not be used here. The precise definition of the image gradient to be used will be described in Sect. 2.3.

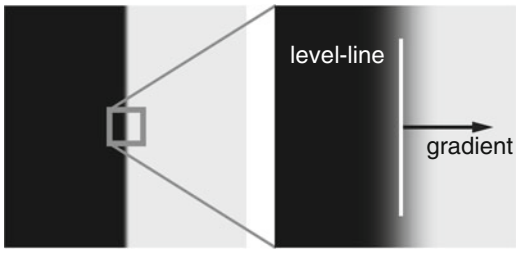

Fig. 2.2 An edge, the image gradient, and a level-line

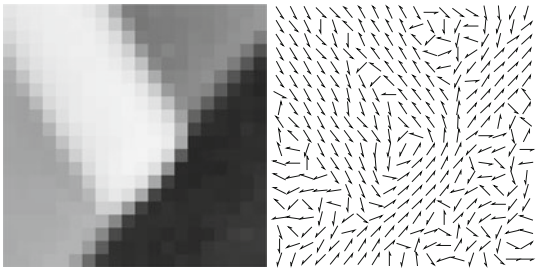

Fig. 2.3 An image (*left*) and its level-line field (*right*). Notice that the level-line elements are oriented, implicitly indicating which side of the level-line is darker

The proposal of Desolneux et al. [19, 22] for detecting line segments is simple: for each candidate, count the number of pixels along it where the level-line angle is compatible with the line segment. *Compatible* means that they have the same direction up to a tolerance τ, see Fig. 2.4. These pixels define the τ-*aligned pixels*, or *aligned* pixels for short:

Definition 2 (τ-aligned pixel). A pixel at coordinates (u,v) is called a τ-*aligned pixel* with a line segment s when the level-line angle at that pixel, $\lambda(u,v)$, shares the same orientation as the line segment up to a precision τ:

$$\left|\text{Angle}\big(\lambda(u,v),s\big)\right| \leq \tau\,.$$

Depending on the number of *aligned* pixels, it will be decided whether a line segment is present or not.

Here is where the *a contrario* approach and the non-accidentalness principle come into play. A candidate will be validated only when its observation corresponds to an unlikely event in the absence of a causal interpretation, that is, in an unstructured image. As in the example of the last section, we need to define a family $\mathscr{S}$ of N_{tests} candidates s_i to test, an observed variable $k(s_i,x)$, and an *a contrario* model H_0. With these elements we will define the NFA of an observation and the constant ε will complete the method.

The family $\mathscr{S}$ includes all the line segments whose end points are centered in pixels of the image. In an $N \times M$ image we have NM possible initial points and NM possible end points; thus $N_{\text{tests}} = (NM)^2$. Notice that the line segments are *oriented*, meaning that the order of their starting and ending points is not arbitrary: an edge is a transition from dark to bright, and the orientation of a line segment indicates which is the dark side. The resolution of digital images is limited by its discrete nature; for that reason, this family of $(NM)^2$ candidates is rich enough to contain all the relevant line segments.[1]

[1] The family $\mathscr{S}$ does not include all the possible line segments in the image; it does not include all the *discrete* line segments, either. There is an extensive literature on the subject of *digital straightness* [47, 73]. Consider an $N \times N$ grid of pixel centers. A *digital straight line* is a set of points in the grid that corresponds to the digitalization of a line. For example, the line $y = ax + b$, has a

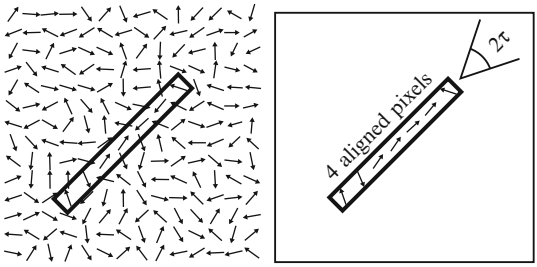

Fig. 2.4 *Left*: a line segment candidate shown over the level-line field. *Right*: the number of aligned pixels, up to an angular tolerance of τ, is counted. The candidate shown has four aligned pixels among seven

As stated before, the observed variable $k(s,x)$ for candidate s in image x is the number of *aligned* pixels in the line segment. For this count, the line segment is defined as a rectangle of width equal to one pixel, see Fig. 2.4. Given a candidate line segment s, we will denote by $n(s)$ the total number of pixels in the corresponding rectangle. The validation then takes into account that $k(s,x)$ aligned pixels were observed among a total of $n(s)$ pixels.

The only information measured from the image are the level-line angles. Hence, Desolneux et al. [22] proposed a stochastic model H_0 for unstructured level-line fields satisfying the following properties:

- $\{\lambda(u,v)\}$ is a family of independent random variables,
- $\lambda(u,v)$ is uniformly distributed over $[0,2\pi]$,

where $\lambda(u,v)$ is the level-line angle at pixel (u,v). Notice that this is a stochastic model for a *level-line field* and not for an image. Strictly speaking, the data x should be defined as this field; to simplify notation and because no real risk of confusion is present, we will use x when referring to an image or to its level-line field.

This model corresponds indeed to unstructured level-line fields: there is no relation between the values at different pixels and the values show an isotropic distribution. The choice can also be justified in the fact that, under certain sampling conditions, this model corresponds to the level-line field of white noise images, see [22, p. 67]. As shown by Attneave [8], white noise images produce an impression of homogeneity, which corresponds to no detection in human vision. Moreover, this model represents well isotropic zones, while straight edges are exactly the opposite: highly anisotropic zones. Thus, in practice a set of pixels will not be accepted as a line segment if it could have been formed by an isotropic process. A good example is shown in Fig. 1.3. The foliage of the tree is far from being a white noise process, but it is an isotropic structure; as a consequence, no line segment will be validated by the *a contrario* approach.

---

corresponding digital line $\{(n,y_n) : n = 1,2,\ldots,N,\ y_n = \lfloor an+b \rfloor,\ 1 \le y_n \le N\}$. Just the digital straight *lines* on an $N \times N$ grid are $O(N^4)$, see [52]. Moreover, in most cases there are multiple digital straight line *segments* defined by the same end points; for example, $\{(1,1),(1,2),(1,3),(2,4)\}$ and $\{(1,1),(2,2),(2,3),(2,4)\}$ are both digital straight line segments from $(1,1)$ to $(2,4)$.

We are now in a position to define the NFA for a candidate line segment s in an image x using the general formulation of the previous section:

$$\mathrm{NFA}(s,x) = N_{\text{tests}} \cdot \mathbb{P}[k(s,X) \geq k(s,x)] \ ,$$

where X is a level-line field according to the model H_0.

Given the isotropic angle distribution under H_0, the probability that any individual pixel is τ-*aligned* with a line segment s is $p = \frac{\tau}{\pi}$. The number of aligned pixels in the *a contrario* model, $k(s,X)$, can be expressed as the sum $A_1 + A_2 + \cdots + A_{n(s)}$, where A_j are Bernoulli variables taking the value 1 if the pixel j is aligned to the line segment s or zero otherwise. Due to the independence at different pixels in H_0, the variables A_j are independent and identically distributed; thus, its sum $k(s,X)$ follows a binomial distribution and,

$$\mathbb{P}[k(s,X) \geq k(s,x)] = B\big(n(s), k(s,x), p\big) \ ,$$

where $B(n,k,p)$ is the tail of the binomial distribution:

$$B(n,k,p) = \sum_{t=k}^{n} \binom{n}{t} p^t (1-p)^{n-t} \ .$$

Putting all together we get the NFA value associated to an observation:

$$\mathrm{NFA}(s,x) = (NM)^2 \cdot B\big(n(s), k(s,x), p\big) \ . \tag{2.3}$$

As in the last section, the NFA value corresponds to the expected number of line segments in H_0 which are as good as s. A large NFA value means that similar events appear often by chance and the structure is not relevant; inversely, a small NFA value reflects an unlikely and meaningful event. Line segment candidates with $\mathrm{NFA}(s,x) \leq \varepsilon$ are kept as detections and are called ε-*meaningful line segments*. The formulation is exactly the same as in the previous section and, as before, we can show that the method satisfies the non-accidentalness principle to a level ε:

Proposition 2. *Line segments with* $\mathrm{NFA} \leq \varepsilon$ *are* ε*-meaningful events. Equivalently,*

$$\mathbb{E}\left[\sum_{s \in \mathscr{S}} \mathbb{1}_{\mathrm{NFA}(s,X) \leq \varepsilon}\right] \leq \varepsilon$$

where $\mathbb{E}$ *is the expectation operator,* $\mathbb{1}$ *is the indicator function,* $\mathscr{S}$ *is the family of line segments considered, and* X *is a random image on* H_0*.*

Proof. We define $\hat{k}(s)$ as

$$\hat{k}(s) = \min\left\{ \kappa \in \mathbb{N},\ \mathbb{P}\big[k(s,X) \geq \kappa\big] \leq \frac{\varepsilon}{N_{\text{tests}}} \right\} \ .$$

Then, $\text{NFA}(s,x) \leq \varepsilon$ is equivalent to $k(s,x) \geq \hat{k}(s)$. Now,

$$\mathbb{E}\left[\sum_{s\in\mathscr{S}} \mathbb{1}_{\text{NFA}(s,X)\leq\varepsilon}\right] = \sum_{s\in\mathscr{S}} \mathbb{P}\left[\text{NFA}(s,X) \leq \varepsilon\right] = \sum_{s\in\mathscr{S}} \mathbb{P}\left[k(s,X) \geq \hat{k}(s)\right] .$$

But, by definition of $\hat{k}(s)$ we know that

$$\mathbb{P}\left[k(s,X) \geq \hat{k}(s)\right] \leq \frac{\varepsilon}{N_{\text{tests}}} ,$$

and using that $\#\mathscr{S} = N_{\text{tests}}$ we get

$$\mathbb{E}\left[\sum_{s\in\mathscr{S}} \mathbb{1}_{\text{NFA}(s,X)\leq\varepsilon}\right] \leq \sum_{s\in\mathscr{S}} \frac{\varepsilon}{N_{\text{tests}}} = \varepsilon ,$$

which concludes the proof. □

Proposition 2 guarantees that false detections are controlled in H_0. For the same reasons as before, we set $\varepsilon = 1$, once and for all.

Figure 2.5 shows an example of applying the described method on a simple noisy image. The validation step did a good job: detections are made *only* where edges are present in the image. However, the detection result is not completely satisfactory. Each straight edge in the image produces many redundant line segment detections, differing in the exact localization, angle and length. Also, many of them are markedly longer than the corresponding edges: any line segment with a significant overlap with an image edge will produce a detection. In addition, the exhaustive test of $(NM)^2$ line segments requires considerable computing time.

The next three sections will discuss in detail several aspects of this basic algorithm. Then, Sect. 2.6 will discuss further improvements to the basic algorithm to cope with the redundancy disadvantages just mentioned.

2.3 Gradient Computation and Sampling

The theory of *a contrario* line segment detection presented in this chapter entails a control of the number of false detections in white noise. As shown, the number of false detections can be bounded assuming independent pixel gradient orientations (orthogonal to level-line orientations). This independence relies on a trade-off between the support of the discrete operator chosen to approximate the gradient and the discretization resolution. If the gradient is computed using a 2×2 support,

$$\nabla x(u,v) = \begin{pmatrix} \frac{x(u+1,v)+x(u+1,v+1)-x(u,v)-x(u,v+1)}{2} \\ \frac{x(u,v+1)+x(u+1,v+1)-x(u,v)-x(u+1,v)}{2} \end{pmatrix} , \tag{2.4}$$

the algorithm should only consider pixels at a distance larger than two, in order to guarantee the potential independence of their gradient orientation.

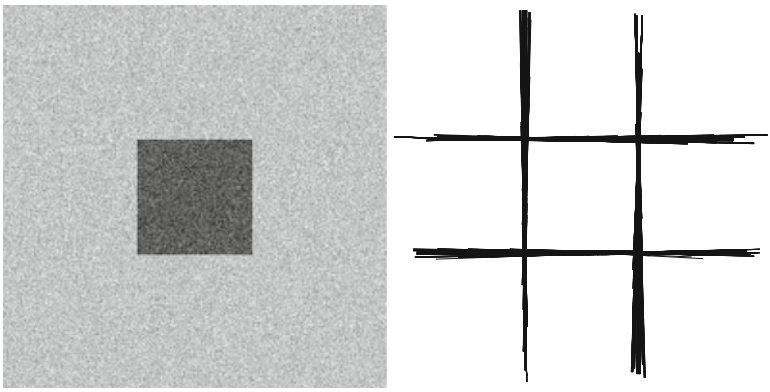

Fig. 2.5 An example of line segment detection by Desolneux, Moisan, and Morel's method [19]

One way to do this is to drop one pixel out of two along the end points of the line segment. But with the pixels, some information is dropped too. If all pixels are kept, the unstructured model H_0 of Sect. 2.2 is not valid as adjacent pixels necessarily have dependent gradient angles. It is still possible to build an *a contrario* method, but a theoretical bound for the number of false detections under the dependency model is much harder to get. However, numerical experiments on white noise show that the number of false detections does not blow up when all pixels in the discretization are kept, and we assume independence. This is an empirical verification that the non-accidentalness principle is still valid without any down-sampling. The advantage of using all pixels instead of one out of two is the ability to detect finer structures; Fig. 2.6 shows two examples.

To test experimentally the number of false detections, two sets of $N \times N$ level-line fields were generated [34]. In the first one, the theoretical conditions of H_0 are satisfied; that is, all level-line angles are independent and isotropic. The second set was generated by directly computing the level-line field on $N \times N$ white noise images using the operator in (2.4); the angles obtained are uniformly distributed in $[0, 2\pi]$ as shown in [22, p. 67], but adjacent angles are dependent. Let us call these sets "independent gradient" and "dependent gradient", respectively. Desolneux et al.'s algorithm as described in Sect. 2.2 was applied to each of the level-line fields in each of the two sets, and for various values of ε. Table 2.2 shows the mean number of detections for each of the two sets. We want to compare the number of false detections; we set ε to large values to see the effect more clearly.

Note that in the "independent gradient" set that corresponds to the theoretical *a contrario* model, the mean number of false detections is smaller than ε, about the half. This offset is due to the discrete nature of the binomial distribution: The number of *aligned* pixels k in a line segment is an integer quantity; thus, the detection threshold κ will be set to the largest *integer* for which

$$\mathrm{NFA} = (NM)^2 \cdot B\big(n, \kappa, p\big) \leq \varepsilon \ .$$

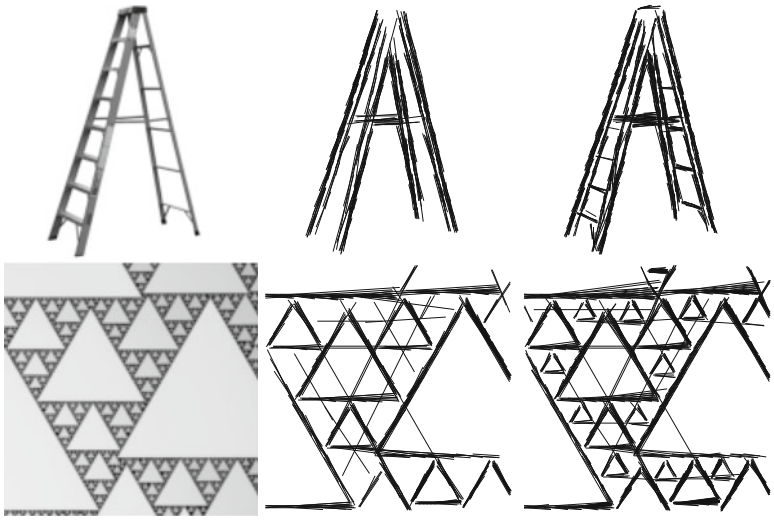

Fig. 2.6 Two examples of the improved resolution obtained by using all pixels instead of one out of two. *Left:* input images, 144×151 and 130×130 pixels. *Center:* meaningful line segments using one-out-of-two pixels. *Right:* meaningful line segments using all pixels. The steps of the ladder are only detected when all pixels are used. In the second case, smaller triangles are detected when using all pixels. No false detection occurs

In a typical case, the NFA value obtained for κ will not be exactly ε, but considerably smaller. This effect, acting on the whole family of tests, leads to offset observed in Table 2.2, see [33] for more details.

The number of false detections observed in Table 2.2 has the same order of magnitude for the independent and dependent cases. The numbers obtained are very similar, suggesting that the same detection thresholds can be used for the dependent case without a degradation of the number of false detections. In fact it can be checked that gradient orientations at neighboring pixels computed with a 2×2 mask are fairly decorrelated in white noise (the correlation between neighbor angles is about 0.2). Moreover, a study performed by Pătrăucean [67] using Markov process models suggests that this dependency may actually *decrease* the number of false detections. In such a case, the *a contrario* framework is still satisfied and the control of false detections is still achieved.

To sum up, we do not know the exact detection thresholds to take into consideration the dependency induced by the gradient computation mask, but the empirical evidence is strong enough to support the use of the same thresholds as for the independent case.

Table 2.2 A measure of the sensitivity of Desolneux, Moisan, and Morel's algorithm to the dependency of pixels. The table shows the mean number of detections in 100 white noise images of size 50×50 for different values of ε. The offset between the mean number of detections in H_0 and ε observed in both cases is due to the discrete nature of the binomial distribution, see [33] for details

ε	100,000	10,000	1,000	100	10	1
Independent gradient	50,590	4,291	395.9	40.06	3.96	0.41
Dependent gradient	50,976	4,300	388.2	37.51	3.61	0.25

2.4 Angular Precision

Desolneux et al.'s algorithm depends on the choice for the angular precision p, related to the angular tolerance τ by $p = \frac{\tau}{\pi}$. The *a contrario* approach will control the number of false detections independently of the precision p used, but the usefulness of the method as an line segment detector can vary with its value. Extreme values would render the method useless: a value $p = 1$ would never produce detections and with $p = 0$ only perfect straight edges would be detected, rendering the method too sensitive to noise.

What value should be used? Considering that our aim is to detect straight edges, a reasonable requirement is $p \leq 1/4$, which corresponds to angles of ± 45 degrees relative to the perfect orientation. For angles larger than 45 degrees, the meaning of being *aligned* to the edge dissolves and *orthogonal* to the edge becomes a better description. Due to image noise induced by acquisition and quantization, and due to imperfection of edges, it is rare to observe precision finer than $p = 1/64$. In practice, the range of $1/4$ to $1/32$ is good enough: coarser precision improves little the robustness to noise and finer precision seldom occurs [22, p. 66].

The selection of p also implies a compromise between robustness to noise and the ability to detect short line segments. Consider that the level-line elements shown in Fig. 2.7 are observed in a 100×100 image. With a precision $p = 1/32$ we get six aligned pixels for A, but *zero* aligned pixels in the noisier case B; the latter will not produce a detection while the former, with $\mathrm{NFA}_A^{p=1/32} = 100^4 \cdot 32^{-6} \approx 0.09$, will be considered 1-meaningful. Inversely, with $p = 1/8$ we get six aligned pixels in A and nine aligned pixels in B. Now, $\mathrm{NFA}_A^{p=1/8} = 100^4 \cdot 8^{-6} \approx 381.5$ and $\mathrm{NFA}_B^{p=1/8} = 100^4 \cdot 8^{-9} \approx 0.75$ so B is 1-meaningful, but A is not.

No fixed precision will get all detections at the same time: fine precisions are able to detect short line segments but are sensitive to noise, large precisions are robust to noise, but cannot detect short line segments. Values like $p = 1/16$ or $1/8$ are usually the best choices.

Even better is to use a multi-precision p approach: apply the same method repeatedly, but for different values of p, for example $1/64$, $1/32$, $1/16$, $1/8$, and $1/4$. In such a case, we need to make some adjustments so that the resulting method still controls the number of false detections. Indeed, if we apply γ times the original method but with different values of p, we would get ε false detections for each one,

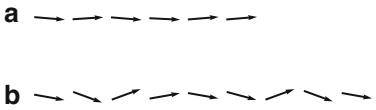

Fig. 2.7 Two sets of level-line elements. *A*: six pixels with precision $1/32$, (± 5.6 degrees). *B*: nine pixels with precision $1/8$, (± 22.5 degrees)

a total of $\gamma\varepsilon$ false detections and γ times more than desired. There are different ways to correct the method to obtain again an average of ε false detections. One is to count in N_{tests} all the tests for different precisions. But the simpler one is to *divide* ε among the γ different detectors applied. Each detector would have an allowance of ε/γ false detections; summing all together, the multi-p method will produce, on average, ε false detections.

2.5 Continuous Angular Measurement

The last section made clear the need to use different values for the precision p to have a good compromise between accuracy and robustness to noise. It is natural to attempt to handle all precisions at the same time, using continuous statistics instead of the binomial ones [33].

In the original Desolneux et al.'s method, an observation consists of a binary sequence $(a_1,\ldots,a_n)$, where a_j take the values zero or one according to whether the j-th pixel is τ-aligned or not. In other words, the angle information is quantized with precision τ. Alternatively, we can keep the level-line angle information as is. Let us normalize the angles between 0 and 1, zero meaning a perfect alignment and one meaning the level-line is opposed to the line segment. Now a_j take values in the full interval $[0,1]$.

The space of possible configurations $(a_1,\ldots,a_n)$ defines a unit hypercube of n dimensions with one vertex at the origin, see Fig. 2.8. The perfect line segment corresponds to the configuration $(0,0,\ldots,0)$ and the worst one to $(1,1,\ldots,1)$. But now, instead of restricting ourselves to the vertices of the hypercube, as in Desolneux et al.'s detector, any point of it is a valid configuration. Given a candidate c in an image x, a configuration $(a_1,\ldots,a_n)$ is determined, and the measure $k(c,x)$ that evaluates the quality of line segments is a real-valued function defined on the hypercube. This function defines a family of iso-$k(c,x)$ surfaces inside the hypercube that we will call *frontiers*. For example, a value α gives the frontier surface $k(c,x)=\alpha$.

Given the function $k(c,x)$, the *a contrario* approach does the rest. The unstructured model H_0 defined in Sect. 2.2, with independent and uniformly distributed level-line angles, induces a uniform distribution on the hypercube of configurations. The probability of an observed event, $\mathbb{P}[k(c,X)\geq k(c,x)]$, is then obtained by integrating the uniform distribution under the frontier surface defined by the value of $k(c,x)$.

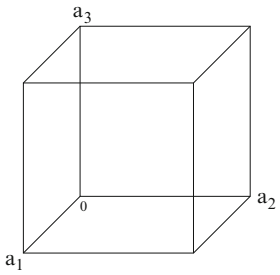

Fig. 2.8 Hypercube of possible configurations, shown for $n = 3$

The selection of the function k, or equivalently of the frontier family, needs some discussion. A simple idea is to use functions of the form $k(c,x) = \sum_j f(a_j)$, where a_j is the normalized angle at pixel j in the candidate and f is some function on the normalized angles. A natural choice for f is $f_A(a) = -a$, that defines an *additive* family of frontiers of the form $a_1 + a_2 + \cdots + a_n = \alpha$, see Fig. 2.9 left. Notice the minus sign to obtain the largest value for the perfect alignment. A second option is the *multiplicative* frontiers family $a_1 \cdot a_2 \cdots a_n = \alpha$, see Fig. 2.9 right. This corresponds to using $f_M(a) = -\log(a)$. These two frontier families were proposed in [33] and a similar formulation but with a different class of f functions was proposed before in [42].

The multiplicative family presents a small drawback: The presence of *one* perfectly aligned pixel, $a_j = 0$, is enough to produce a meaningful line segment, regardless of the angle direction of the other pixels. Nevertheless, this problem is not new. In the binomial formulation, if the value of p is chosen small enough, one aligned pixel will also lead to a meaningful line segment. On the other hand, $a_j = 0$ should never happen, except in synthetic images. Unfortunately, due to the discretization of pixel values this can happen in practice. The problem has a simple (although not elegant) solution by imposing a minimum attainable precision in the measurement, so a_j can never be zero.

We need to compute

$$\mathrm{NFA}(c,x) = N_{\text{tests}} \cdot \mathbb{P}[k(c,X) \geq k(c,x)] \,.$$

The probability term is determined by the volume of the hypercube under the frontier defined by $k(c,x)$.

For the additive case, the frontier is the hyperplane $a_1 + \cdots + a_n = -k_A(c,x)$ and the sought probability is determined by the intersection of the unit hypercube with the half-plane $a_1 + \cdots + a_n \leq -k_A(c,x)$. This computation is straightforward when $\alpha = a_1 + \cdots + a_n \leq 1$ because the intersection is the standard simplex of size α. The volume of this simplex is $\frac{\alpha^n}{n!}$, see [78]. Then,

$$\mathbb{P}[k_A(c,X) \geq k_A(c,x)] = \frac{[-k_A(c,x)]^n}{n!}, \qquad \text{valid for } -k_A(c,x) \leq 1 \,, \tag{2.5}$$

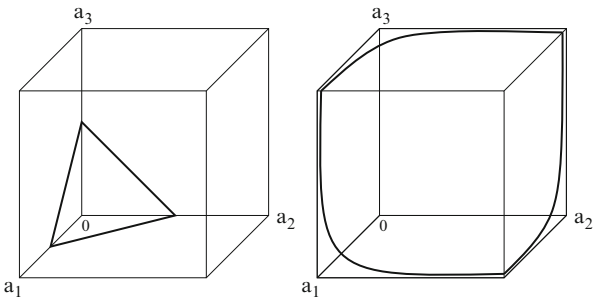

Fig. 2.9 Two examples of frontiers. *Left*: the additive frontier defined by $\alpha = a_1 + \cdots + a_n$. *Right*: the multiplicative frontier defined by $\alpha = a_1 \cdot a_2 \cdots a_n$

where n is the number of pixels in c. Notice that when $-k_A(c,x) = a_1 + \cdots + a_n > 1$ the previous formula is no longer valid because part of the simplex lays outside the hypercube, overestimating the volume; for this same reason, (2.5) is always an upper bound to the probability term and

$$N_{\text{tests}} \frac{(a_1 + \cdots + a_n)^n}{n!} \leq \varepsilon$$

implies an ε-meaningful detection, even if it is not a necessary condition. A closed-form expression valid for any value of $k_A(c,x)$ can be derived from the general case of hypercube and half-plane intersections given in [59]:

$$\mathbb{P}[k_A(c,X) \geq k_A(c,x)] = \frac{1}{n!} \sum_{i=0}^{\lfloor -k_A(c,x) \rfloor} (-1)^i \binom{n}{i} \big[-k_A(c,x) - i \big]^n ,$$

where $\lfloor -k_A(c,x) \rfloor$ is the largest integer not greater than $-k_A(c,x)$. Notice that (2.5) corresponds to the first term of this sum.

For the multiplicative case we need to observe that since the values a_j are uniformly distributed under H_0, $-\log(a_j)$ follows an exponential distribution under H_0 and $k_M(c,X)$ a Γ distribution with parameters $(n,1)$. There is a simple closed form for the associated cumulative distribution function $\mathbb{P}[k_M(c,X) \geq k_M(c,x)]$:

Proposition 3.

$$\mathbb{P}[k_M(c,X) \geq k_M(c,x)] = \exp\big(-k_M(c,x) \big) e_{n-1}\big(k_M(c,x) \big) ,$$

where $e_m(z) = 1 + z + \frac{z^2}{2} + \cdots + \frac{z^m}{m!}$.

Proof. The density of a Γ distribution of parameters $(n,1)$ has the form, $\frac{t^{n-1}}{(n-1)!} e^{-t}$. Hence

$$\mathbb{P}[k_M(c,X) \geq k_M(c,x)] = \frac{1}{(n-1)!} \int_{k_M(c,x)}^{+\infty} t^{n-1} e^{-t} dt .$$

Integration by parts leads to the sought formula. □

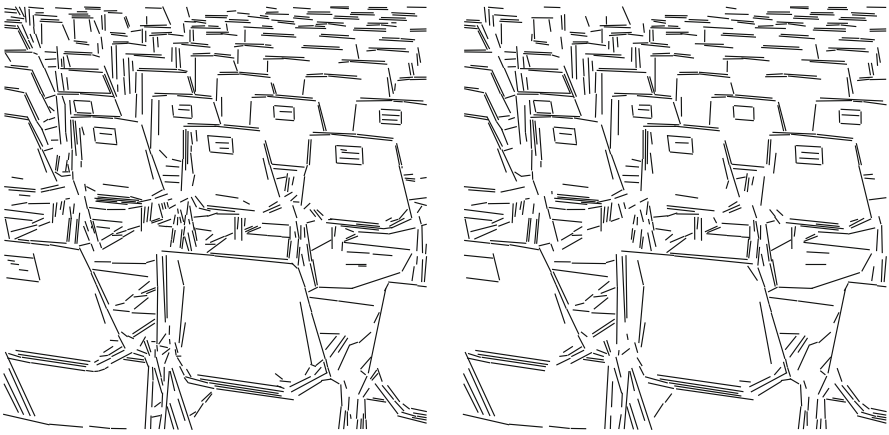

Fig. 2.10 Line segment detection for the image shown in Fig. 1.1, using continuous angular precision with additive (*left*) and multiplicative (*right*) NFA. Both results are very similar to the one obtained with a multi-precision p approach (Fig. 1.1)

For both, the additive and the multiplicative methods, the candidates that satisfy the test $\text{NFA}(c,x) \leq \varepsilon$ are selected as detections and we set $\varepsilon = 1$ as usual. Figure 2.10 shows sample detections for both criteria. Comparing with Fig. 1.1, one can see that the discrete formulation produces comparable results, but requires a multi-precision p approach. Among the continuous formulations, the additive criterion seems to capture slightly finer details.

2.6 Exclusion Principle and Further Improvements

As illustrated in Fig. 2.5, an exhaustive search for line segments produces many redundant detections. Two divergent approaches were proposed to cope with redundancy and inaccuracy. The first one uses a criterion to select the best detections and remove the others; this is done *after* the exhaustive search and validation. The second one uses the *heuristic search plus validation* strategy where a reduced number of candidates is selected *previously* to validation.

Desolneux, Moisan, Morel and Almansa proposed various criteria to cope with redundancy [4, 5, 19–22]. Only the *exclusion principle* [22] will be described here, the most general of them. With a name borrowed from *quantum mechanics*, the principle is defined as follows:

Definition 3 (Exclusion Principle). Each measurement can support (or vote for) at most one detection.

In the case of line segment detection, *measurements* refer to level-line angles; in other words, each pixel of the image can vote for only one final line segment. The previous general definition does not indicate how a measurement should choose the candidate to vote. An iterative algorithm is generally used:

Algorithm 1: Exclusion principle

input: An set of candidates $\mathscr{C}$.
output: A list $\mathscr{O}$ of detections.

```
Mark all pixels as Available.
Select the candidate c ∈ 𝒞 with best (smallest) NFA(c) value.
while NFA(c) ≤ ε do
    Add c to 𝒪 and remove it from 𝒞.
    Mark the aligned pixels in c as Unavailable.
    Recompute all NFA values counting only Available pixels.
    Select c ∈ 𝒞 with best (smallest) NFA(c) value.
end
```

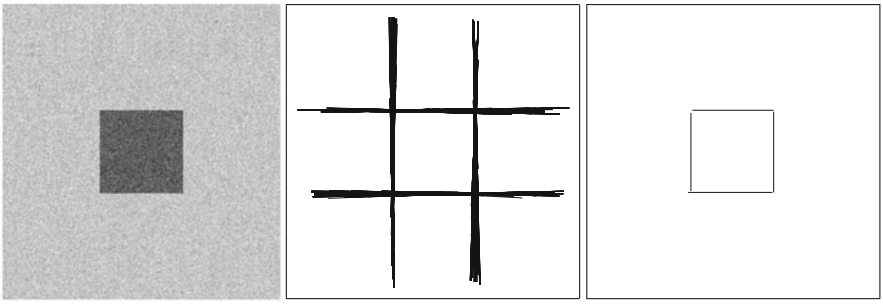

Fig. 2.11 Redundancy reduction by Exclusion Principle. *Left*: input image. *Center*: all meaningful line segments. *Right*: meaningful line segments using the exclusion principle

The line segment with the smallest NFA value will be selected first and it will usually correspond well to a straight edge on the image. Then, all its aligned pixels are *excluded* from being used by the remaining candidates, whose NFA values are re-computed without using them. Redundant detections share most of their aligned pixels. Thus, when the best candidate appropriates its pixels, it leaves few or no aligned pixels on redundant line segments; the new NFA values are large and the redundant line segments are no longer meaningful. Figure 2.11 shows an example.

Unfortunately, the previous approach does not solve all the problems. When two or more aligned line segments are present, see Fig. 2.12, the exclusion principle can provide an erroneous interpretation: a longer structure covering all the aligned ones may be preferred to individual detections. For example, imagine that the level-line field shown in Fig. 2.13 appears in a 100×100 image. With a precision $p = 1/32$, the NFA value for each line segment of six elements is $100^4 \cdot 32^{-6} \approx 0.09$. The longer line segment covering the full 18 pixels has 12 aligned pixels, and its NFA

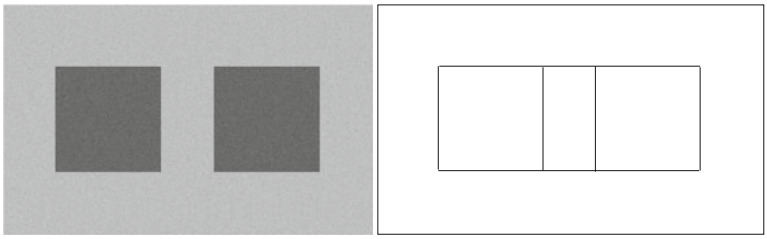

Fig. 2.12 The problem of aligned structures. *Left*: an image of two aligned squares. *Right*: meaningful line segments using the exclusion principle

Fig. 2.13 Which is the best interpretation for these level-line elements: *two* line segments, each of six elements, or *one* longer line segment of 18 elements?

values is $100^4 \cdot B(18, 12, 1/32) \approx 10^{-5.9}$. When comparing individual line segments, the long one has a better (smaller) NFA value than each of the shorter ones; thus one long line segment results from the exclusion principle.

Now, the two interpretations were not compared fairly. The data sets were of different size: 6 against 18 observed pixels. More reasonable would be to compare the interpretation of the entire data as *one* long line segment against the interpretation of *two* aligned line segments. This idea is used in the *multisegment detector* [31, 34]. The input digital image is analyzed line by line in all directions, and for each line the best interpretation in terms of aligned line segments is computed. A set of aligned and disjoint line segments will be called a multisegment, see Fig. 2.14.

Each line of an image is analyzed as a binary sequence of aligned or not aligned pixels (given a precision p). Then, every binary segmentation of the sequence is considered and its quality is evaluated by the NFA of an n-multisegment $(s_1, \ldots, s_n)$ supported by the line L:

$$\mathrm{NFA}(s_1, \ldots, s_n) = \#\mathscr{L} \cdot \binom{l(L)}{2n} \cdot \prod_{i=1}^{n} \big(l(s_i)+1\big) B\Big(l(s_i), k(s_i), p\Big) \,,$$

where $\#\mathscr{L}$ stands for the total number of lines considered on the image, $l(L)$ and $l(s_i)$ stand for the length of the line L and of the line segment s_i, respectively, and $k(s_i)$ is the number of aligned pixels on the line segment s_i. A multisegment $(s_1, \ldots, s_n)$ is called ε-meaningful whenever $\mathrm{NFA}(s_1, \ldots, s_n) \leq \varepsilon$. The multisegment with the lowest NFA value gives the best interpretation for that line. Actually, the core of the multisegment detector, including the NFA definition stated before, works as a general-purpose binary sequence segmentation algorithm and can be used for other problems as well.

A multi-precision p approach as described before can also be applied to the multisegment detector and a dynamic programming algorithm allows some speed improvements. Refer to [31, 34] for a thorough description of the multisegment detector.

Figure 2.15 shows the result of the multisegment detector for the image in Fig. 2.12. As one can see, the right interpretation was found. Some more examples will be commented in Chap. 4. As we will see, the multisegment detector produces, in general, satisfactory results. The algorithm computes the best interpretation for each line. Nevertheless, the best interpretation *line by line* does not always correspond to the best *global* interpretation of the image. Figure 2.16 shows an example. The method can be extended even further in this sense, preferring the best interpretation in terms of line segments for the whole image. The formulation would be necessarily much more complex, carrying the implicit suggestion of further extensions to incorporate other geometrical structures in the global interpretation.

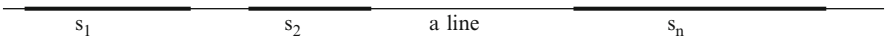

Fig. 2.14 An n-multisegment is an n-tuple $(s_1, \ldots, s_n)$ of n *disjoint* and *collinear* line segments

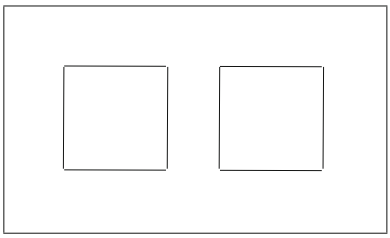

Fig. 2.15 The result of the multisegment detector on the example of Fig. 2.12

The multisegment detector is so far the more sophisticated line segment detector in the family of *a contrario* exhaustive methods. The *exhaustive* approach leads nonetheless to inefficient algorithms: the original method has an algorithmic complexity of $O(N^4)$, for an $N \times N$ image, whereas the multisegment detector has a complexity of $O(N^5)$. These algorithms could hardly produce *real time* results and their computation time is prohibitive even for medium size images.

The next chapter describes the alternative approach, the heuristic search plus validation used in the LSD algorithm.

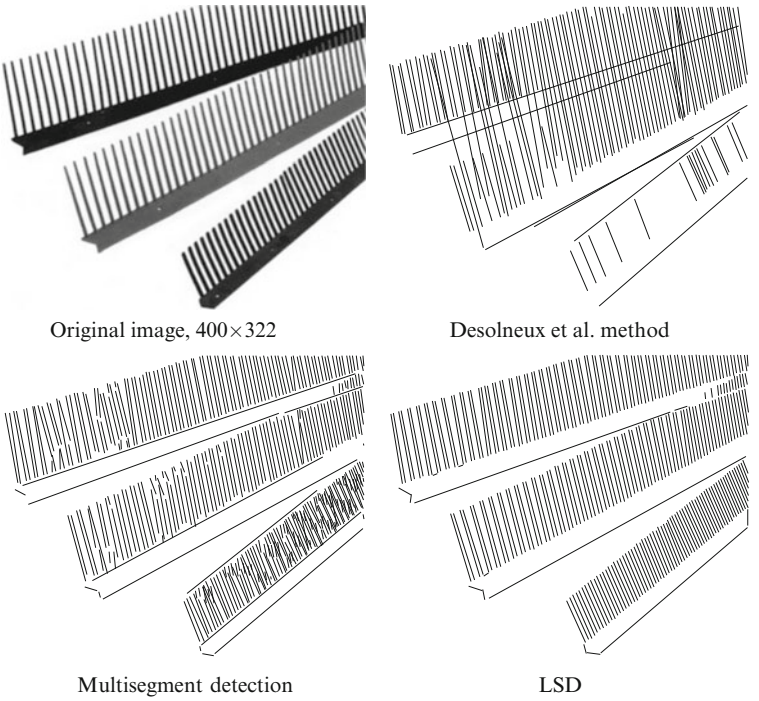

Fig. 2.16 A comparison between three *a contrario* line segment detection methods. Desolneux et al. and the multisegment detector, both hallucinate global structures as a result of the aligned combs. Desolneux et al. produces slanted and longer detections. The multisegment detector makes erroneous cuts in line segments and some slanted ones

http://www.springer.com/978-1-4939-0574-4

A Contrario Line Segment Detection
von Gioi, R.G.
2014, VIII, 90 p. 60 illus., Softcover
ISBN: 978-1-4939-0574-4