Int J Comput Vis
DOI 10.1007/s11263-012-0543-6

Beyond Independence: An Extension of the A *Contrario* Decision Procedure

Artiom Myaskouvskey · Yann Gousseau · Michael Lindenbaum

Received: 16 May 2011 / Accepted: 5 June 2012

Abstract The *a contrario* approach is a principled method for making algorithmic decisions that has been applied successfully to many tasks in image analysis. The method is based on a *background model* (or null hypothesis) for the image. This model relies on independence assumptions and characterizes images in which no detection should be made. It is often image dependent, relying on statistics gathered from the image, and therefore adaptive. In this paper we propose a generalization for background models which relaxes the independence assumption and instead uses image dependent second order properties. The second order properties are accounted for thanks to graphical models. The modified *a contrario* technique is applied to two tasks: line segment detection and part-based object detection, and its advantages are demonstrated. In particular, we show that the proposed method enables reasonably accurate prediction of the false detection rate with no need for training data.

Keywords A contrario decision · Object recognition · Significance test · Background model · Number of false alarms · Meaningful matches

A. Myaskouvskey (✉) · M. Lindenbaum
Computer Science Dept., Technion, Haifa 32000, Israel
e-mail: artiom@cs.technion.ac.il

M. Lindenbaum
e-mail: mic@cs.technion.ac.il

Y. Gousseau
Telecom ParisTech, LTCI CNRS, Paris 75634, France
e-mail: gousseau@telecom-paristech.fr

Published online: 29 June 2012

1 Introduction

Common algorithms in computer vision, such as, edge detection and object recognition, involve decisions. These decisions typically calculate some scalar function and test it against a threshold. The quality of the decision is usually very sensitive to the chosen function and the threshold value. A common approach to such decision procedures follows the Bayesian methodology; see, e.g., Weber et al. (2000), Konishi et al. (2003). The decision is optimized so that it minimizes a cost function related to two types of errors: false alarms and misses. Statistical models, required for this minimization, are usually learned in a training phase. Costly to collect and label, the training data often does not best represent the true distribution of the test data. Moreover, even if it does, the optimized parameters are best only on the average and are not necessarily optimal for the particular image at hand.

An alternative, general, non-Bayesian approach for decision making and parameter tuning, suggested several years ago (Desolneux et al. 2000, 2008), has already been applied successfully to diverse tasks; see e.g., edge detection (Desolneux et al. 2001), histogram mode selection (Desolneux et al. 2008), robust point matching (Moisan and Stival 2004) or local feature matching (Rabin et al. 2009). Denoted *a contrario*, this powerful methodology quantifies the Helmholtz principle that "we do not perceive any structure in a uniform random image" (Desolneux et al. 2008). See Fig. 1 for an illustration of this principle.

In *a contrario* decisions, the parameters (e.g., the thresholds) are set so that a decision algorithm would not accidentally detect too many visual events (such as edges, lines, familiar objects, etc.) in a "random" image. The random image and the implied false detection rate are specified by a probabilistic background model. Unlike a common approach in

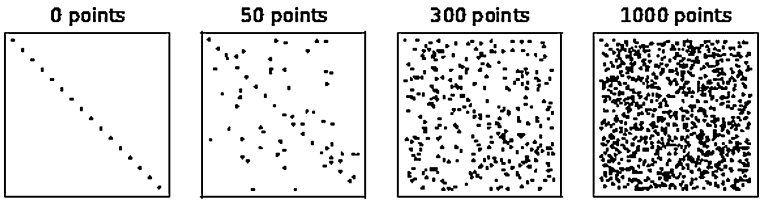


Fig. 1 A line is perceived only when it cannot be considered an accidental event. Every one of the 4 subplots contains one 16-point line and some uniformly distributed random points. The *line* is perceived only when the additional point density is low and the accidental line formation hypothesis can be disqualified

computer vision, this background model is not determined by an off-line training phase, but instead relies on crucial statistical independence assumptions, as well as, in many cases simple data distributions inferred from the image itself. A detection decision is associated with some false detection rate in this model. A decision is accepted if this rate, denoted NFA (number of false alarms), is low. The detected event is called meaningful precisely because it would not be detected in a noise image. A meaningful event is therefore consistent with the Helmholtz principle.

A contrario decisions have several advantages. First, because the background model depends on the image, it follows that the decision parameters (e.g., thresholds) are tuned adaptively to the image at hand and not by optimizing average performance over some training set, which may be non-representative for specific images. Second, every *a contrario* decision is accompanied by a reliability estimate, the NFA. Third, the algorithmic process becomes very flexible. There is a uniform criterion for all decisions, even if made with different features and different models (Desolneux et al. 2008).

A common practice in computer vision is to use image data for normalizing the cost function on which the decision relies. The best example would perhaps be the decision used for SIFT based matching (Lowe 2004). This decision matches a point A to a point B if the appearance (SIFT) distance between A and B is lower than all other distances of points from A, but also requires that the ratio between the minimal distance and the second-smallest distance of some point from A be small. This ratio is in fact a crude indicator of false alarm possibility. Thresholding it is nevertheless extremely effective (Lowe 2004). Unlike *a contrario*, this criterion does not rely on quantitative analysis and therefore does not allow the error probability to be bounded.

The reliability estimate, NFA, in *a contrario* approaches corresponds to the background model and not to the real image. Somewhat surprisingly, the NFA is often indicative of the number of false detections in the actual image. In principle, however, an image is not a white noise realization, and the independence assumption is a simplification: it may lead to decisions which are meaningful for the model, but are not accurate predictions of failures in the real image. This is indeed the case for several common tasks.

The goal of this study is to extend the *a contrario* approach to a more general scope. We develop here statistical background models which do not rely on independence assumptions, and evaluate their power to make meaningful and effective decisions. After discussing the proposed extension in a general context, we focus on two tasks, which demonstrate both the need for the more complex model and the effectiveness of the proposed solution: line segment detection in natural texture and object detection using part based methods. In the first case, which was the first task analyzed using the *a contrario* approach (Desolneux et al. 2000), the support to a line segment comes from pixels along the line with gradient directions consistent with it. Many images (or image parts) do not contain apparent lines, and yet, the gradient directions in nearby pixels cannot be considered independent. In the second case, which is a new application of *a contrario* decisions, objects are detected by comparing image patches to corresponding model parts. The patches may be close and even overlapping and hence their similarities to the model parts are not independent.

In a sense, our approach challenges and generalizes the interpretation of Helmholtz principle. The aforementioned "uniform random image" is usually taken to be white noise (Desolneux et al. 2008), but we argue that many moderately correlated random images are still considered uniformly random by a human observer.

Detecting objects in correlated noise was considered mostly in the signal processing community and in maximum likelihood or Bayesian decision contexts; see Van Trees (1965, p. 287) for typical examples. These decision approaches differ from the proposed approaches in many respects. First, they model the object to be detected as an ideal clean image. Then they model the distortion explicitly by additive colored and usually Gaussian noise. Then they specify some image statistics and analytically calculate their distribution, with and without the object. The decision rule and the implied error rate follow. This approach is rigorous but works mostly for detection methods based on linear filtering (i.e., correlation detector), which is very limited in many computer vision tasks.

We are aware of only one paper discussing *a contrario* detection using a background model that explicitly models dependence (Grosjean and Moisan 2009). The approach

in this paper differs from ours and is related to the signal processing approaches discussed above. It provides stronger results on a more limited scope. In Grosjean and Moisan (2009), the goal, motivated by mammography analysis, is to detect spots over textured regions. The considered background model is colored noise, obtained as the convolution of white noise with some kernel. Thus the background model completely specifies the type of images where no detection is expected. Spots are detected by thresholding a linearly filtered version of the image, which makes the thresholds computable. In contrast, the background model considered in the present work only specifies second order dependencies between some sub-events to be grouped, but it permits the prediction of performances of non-linear detection tasks, such as alignment detection.

Other methods take an indirect approach and transform the dependent image data into another representation where the components are more independent. Comparing image patches using PCA components, for example, lends itself better to *a contrario* decisions (Sabater et al. 2012) than comparing them using pixel values directly. Another recent work replaces the independence assumption with the exchangeability assumption (which allows, for example, to sample without replacement) and thus induces a dependence possibility (Flenner and Hewer 2011).

The paper continues as follows: The next section discusses the *a contrario* decision method briefly and proposes the extension beyond independence based assumptions. The next two sections discuss the two previously mentioned tasks, and appropriate solutions relying on second order dependencies. The proposed generalizations are experimentally tested in the context of the two tasks and compared to the original method. Some suggestions for future research are considered in the last section.

2 Independence in *A Contrario* Decisions

2.1 The Classic *A Contrario* Approach

As discussed above, the *a contrario* methodology proposes to make decisions only by considering false detections in random images. More precisely, decisions are made by controlling the number of false detections in a *background model*, which is the core of the *a contrario* approach. This background model is based on independence assumptions and some marginal distributions extracted from the test image, on which the decision is made. These empirical distributions make the model related to the image at hand, and the decision adaptive to it.

To calculate the probability for false detection and the implied expected number of false detections, the NFA, the detection event is usually specified as a function of simpler events which we call sub-events. The key assumption here is that these sub-events are independent. As an example, consider the task of straight segment detection (alignment) (Desolneux et al. 2000, 2008). A segment hypothesis l is supported by a pixel on it if the gradient orientation in this pixel is approximately orthogonal to the hypothesized segment. The algorithm accepts the hypothesis (i.e., a detection event occurs) if at least k pixels support the line segment. The approximate orientation orthogonality of a single pixel is a sub-event in our notation. The background model considered in Desolneux et al. (2000) is based on the assumption that these orientations are independent. This model is justified in a white noise image when the pixels considered are sufficiently far from each other (typically two pixels apart).

Other *a contrario* decisions may rely on different criteria for accepting a hypothesis, and may use, for example, the sum of continuous random variables associated with the sub-events (Rabin et al. 2009). Almost always, they rely on independence to estimate the probability of detection.

2.2 The Price Paid for Assuming Independence

The independence based probability model is surprisingly effective but nonetheless has several drawbacks:

1. The independence assumption may not match the image and the particular sub-events considered for the detection. Often, sub-events are positively correlated, which makes the independence based NFA estimate overly optimistic. That is, while the expected number of false detections in white noise may be indeed lower than the chosen ϵ, the number of false detections in real images may be far higher.
2. In order to comply with the independence assumption, the designer of *a contrario* procedures has to choose the sub-events so that they are unrelated enough. In the alignment detection described above, for example, the gradient directions are estimated using $s \times s$ masks. Then the gradient direction is sampled every s pixels, which guarantees that the directions are independent (in white noise). Choosing large masks implies that the information in many pixels is ignored. Choosing small masks, on the other hand, may lead to inaccurate direction estimates, especially when the image is binary or of high contrast. In some cases, as in curve (Muse et al. 2006) or SIFT (Rabin et al. 2009) matching, the choice of the sub-events involves a non-trivial tradeoff between independence and discriminative power.

2.3 *A Contrario* Without Independence

In principle, the background model could be more complex and could allow some dependence between the sub-events.

This necessitates a background model specifying the joint distribution of the sub-events, and not only their marginals. The rest of the procedure can then remain unchanged: the decision is made by calculating the expected number of false alarms, NFA, using the modified background model and comparing it to a threshold.

As done in several *a contrario* applications, we shall estimate the more complex model from the given image. Note that the enhanced background model does not need and should not reflect the true probability model of the given image. Such a theoretically perfect model would consider any visual event present in the image, whether it is of interest to the application or not, as "background", and would therefore be unable to detect anything in this image. Care should thus be taken not to make the model overly powerful. Besides, estimating a very high order model from a single image would be unfeasible due to the curse of dimensionality.

We propose a model that is just a little more complex than the independence based model. In this paper we shall consider simple graphical models that are based only on a subset of first and second order statistics. That is, let X_i be a binary random event which gets the value '1' when the ith sub-event succeeds. Then the models we use here are parameterized by the probabilities $\{P(X_i), P(X_i, X_j)\}$.

Rather than giving a general formulation and then reducing it for the particular tasks, we prefer a more concrete description of the formulation in the context of two tasks. The next section describes a relatively simple variation on the alignment task, which uses a Markov model to specify the background model. Then, Sect. 4 describes a more elaborate part-based object detection algorithm, which uses a tree-based graphical model to specify the background model.

3 Line Segment Detection Using Second Order Statistics

As already said, most *a contrario* methods validate events that are unlikely under the hypothesis of i.i.d. sub-events. In particular, the original line segment detection method introduced in Desolneux et al. (2000) relies on the following background model: orientations of the gradient at each pixel are independent and uniform random variables. It is shown in Desolneux et al. (2000) that, for pixels that are 2 pixels apart, this hypothesis is exactly satisfied in Gaussian white noise and is almost true in some other white noise signals such as uniformly distributed white noise.

While the i.i.d. model works surprisingly well for detecting line segments and edges in natural images, it is obviously not justified for some images in which lines should not be detected. A simple example would be micro-textures that can be modeled as colored noise (Galerne et al. 2010). When the original segment detection procedure, runs on a colored noise image, many lines are erroneously detected; see Fig. 5 for a 512×512 image obtained as the convolution of white noise with a Gaussian kernel (s.d. $= 5$), where about 3500 segments are detected. Similarly, lines are detected in natural textures; see Fig. 6.

The validity of the independence assumption in natural images may be challenged by performing a χ^2 test on the empirical joint probability of pixel alignments; see Sect. 5.1.1. Such a test clearly concludes to non-independence, even in the absence of alignments in images.

Our hypothesis is that the excessive detections reported above arise from the use of an inadequate background model. It is therefore of interest to investigate more elaborate background models that would preclude detection in these arguably structureless situations. In this section, after reiterating the original segment detection procedure, we explain how to refine the detection by modeling the dependencies between adjacent pixels.

Original Segment Detection For a segment S having length l, let $X_1, \ldots, X_l$ be the functions taking value one when the gradient orientation at the ith pixel is perpendicular to the orientation of S, up to a precision angle $p\pi$ (with $0 < p < 0.5$), and zero otherwise. Then, $L(S) = \sum_{i=1}^{l} X_i$ is the number of aligned pixels in S.

The principle of *a contrario* detection is to consider a random segment S' and to set a threshold T on $L(S')$ so that $P(L(S') \geq T)$ is small enough. The random segment is assumed to follow the structureless background model already mentioned. In the original formulation (Desolneux et al. 2000), the background model specifies the variables X_is as i.i.d. Bernoulli variables with parameter p. This is the case if the gradient orientations at different pixels are i.i.d. uniformly distributed variables on $[0, 2\pi]$. Therefore, for a segment S' of length l, following the background model implies that the number of aligned pixels follows a binomial distribution. That is,

$$P\big(L\big(S'\big) \geq k\big) = \sum_{i=k}^{l} \binom{l}{i} p^i (1-p)^{l-i}.$$

The NFA of a given (deterministic) segment S is then defined as

$$\begin{aligned} NFA(S) &:= N^4 P\big(L\big(S'\big) \geq L(S)\big) \\ &= N^4 \sum_{i=L(S)}^{l} \binom{l}{i} p^i (1-p)^{l-i}, \end{aligned} \tag{1}$$

where the image is of size $N \times N$. The segment is validated if $NFA(S) \leq \varepsilon$, where ε is a constant. This definition is valid because, by using this decision step, the mathematical expectation of the number of detected segments in the background model (white noise) is less than ε. This result relies

on the linearity of mathematical expectation, and on the fact that N^4 is the number of segments in the image (N^2 choices for each extremity of a segment). It is usually reasonable to set $\varepsilon = 1$, therefore ensuring no more than one detection in white noise, on the average.

Other *a contrario* line detection methods such as the LSD (Grompone von Gioi et al. 2010) or the multi-segment detector (Grompone von Gioi et al. 2008) rely on the independence assumption as well.

Detection with Second Order Modeling In this section, we consider a background model in which some low order dependencies between sub-events are modeled, and propose to base the setting of the decision function $L(S)$ on it. We specify the background model, determining the joint distribution of the variables $X_1, \ldots, X_l$ by a Markov chain of order one. That is, we specify that for all $1 < i \leq l$,

$$P(X_i = x_i | X_{i-1} = x_{i-1}, \ldots, X_1 = x_1) = P(X_i = x_i | X_{i-1} = x_{i-1}).$$

The background model is isotropic (identical for all segment orientations) and homogeneous (stationary) so that for all lines it is characterized by the transition probabilities, $P(X_1 = x_1 | X_0 = x_0)$, for $x_1, x_0 \in \{0, 1\}$. For brevity we sometimes refer to the values of these probabilities as $P(1|1)$, $P(1|0)$, etc. These values are learned from the image at hand, simply by computing empirical frequencies, as further detailed in the experiments in Sect. 5. Observe that segments are not oriented, so that the Markov chain is in fact assumed to be reversible. This is however not a limitation, since it is easily seen that a two-state Markov chain is always reversible.

The probability under the background model to observe a segment with length larger than or equal to k is then computed as

$$P\big(L\big(S'\big) \geq k\big) = \sum_{x_1 + \cdots + x_l \geq k} P(X_1 = x_1) \times \prod_{i=2}^{l} P(X_i = x_i | X_{i-1} = x_{i-1}). \tag{2}$$

Then, segments are detected exactly as before. That is, a segment S with $L(S)$ aligned pixels is kept if

$$NFA(S) := N^4 P\big(L\big(S'\big) \geq L(S)\big) \leq \epsilon,$$

where S' is a segment following the background model. Observe that, as in the original NFA, this quantity only depends on $L(S)$, the number of aligned points in S. Moreover, if $P(X_1 = x_1 | X_0 = x_0) = P(X_1 = x_1 | X_0 = 1 - x_0)$ for $x_1, x_0 = 0, 1$ then both NFAs (in the dependent and independent cases) are identical, as is easily seen from Formula (1) and (2). Another implication of this formula is the minimal length a segment should have to be detected, obtained in a way similar to the independent case. For both cases, the minimal segment length is easily obtained from the NFA associated with segments where all pixels are aligned. This minimal length is equal to $(\log(\epsilon) - 4\log(N))/\log(p)$ in the independent approach, as may be seen from Formula (1), and equal to $(\log(\epsilon) - 4\log(N) - \log(P(1)))/\log(P(1|1))$ in the dependent approach, as may be seen from Formula (2). Observe that this second quantity involves a conditional probability that is learned from the image. We will see in the experimental section that statistics on a typical natural image yield larger minimum lengths in the dependent case than in the independent one, and more generally that detection using the dependent approach is more conservative, that is, yields less detections.

Practical Computation of the NFA The expression (2) for the probability $P(L(S) \geq k)$ seems computationally hard to evaluate. Note that because it depends on the transition probabilities, which, in turn, depend on the specific tested image, it cannot be precomputed. We now show that it may be computed efficiently, in polynomial time, using a dynamic programming algorithm (Cormen et al. 1990). In order to compute $P(L(S') \geq k)$, we write $Y_i = \sum_{j=i}^{l} X_j$ and compute $P(Y_i \geq k)$ (for all $k \in [0, l - i + 1]$) by a descending induction on i. Indeed, we first observe that for $i \leq l - 1$,

$$P(Y_i \geq k) = P(Y_{i+1} \geq k | X_i = 0) P(X_i = 0) + P(Y_{i+1} \geq k - 1 | X_i = 1) P(X_i = 1).$$

Next, we have that, for $x \in \{0, 1\}$ and $k' \geq 1$,

$$\begin{aligned} &P\big(Y_{i+1} \geq k' | X_i = x\big) \\ &= \sum_{y \in \{0,1\}} P\big(Y_{i+2} \geq k' - y, X_{i+1} = y | X_i = x\big) \\ &= P\big(Y_{i+2} \geq k' | X_{i+1} = 0\big) P\big(X_{i+1} = 0 | X_i = x\big) \\ &\quad + P\big(Y_{i+2} \geq k' - 1 | X_{i+1} = 1\big) P\big(X_{i+1} = 1 | X_i = x\big). \end{aligned} \tag{3}$$

Note that $P(Y_l \geq k' | X_{l-1} = x)$ may be easily computed: $P(Y_l \geq 0 | X_{l-1} = x) = 1$, $P(Y_l \geq 1 | X_{l-1} = x) = P(x_i = 1 | X_{i-1} = x)$, $P(Y_l \geq 2, 3, \ldots | X_{l-1} = x) = 0$. Therefore, $P(Y_i \geq k)$, for all k, can be computed by induction. Note that this computation, which takes $O(l^2)$ time and $O(l^2)$ space, is performed only once using $l = N\sqrt{2}$ (maximum length for a segment in the given image) in order to allow the computation of the NFA for all possible segments.

Experimental results using the second order modeling of segments are presented and discussed in Sect. 5. The interested reader may skip the next section, devoted to part-based object detection, and go directly to the experiments.

4 *A Contrario* Based Object Detection

4.1 The Detection Task

We now consider the task of object detection, where a decision is made as to whether an object belongs to a given category. The common approaches to this well-studied task use part-based representations that can handle the high variability of objects in the same category and detect them accurately; see some examples in Vidal-Naquet and Ullman (2003), Fergus et al. (2007), Zhang et al. (2007), Sivic et al. (2005).

Due to the difficulty in modeling the appearance of objects, detection algorithms are usually constructed from examples, using learning techniques. They are trained as classifiers, using training sets containing images of objects from the category to be detected (positive examples, or *targets*) and images of objects from other categories (negative examples, or *non-targets*). The detection function is specified up to a set of parameters, and these are optimized by minimizing some combination of miss and false detection training errors. The detection reliability is evaluated by testing the algorithm on a set of validation data. Naturally, when the test conditions differ from the training conditions, both the detection performance and its predictability may deteriorate.

The detection algorithm considered in this paper uses very limited training data and the information in the given image (inside and outside the subimage candidate window) to decide whether an object belongs to a given category. The part-based model is specified from a training set containing only positive examples. Important parameters, such as the appearance similarity threshold and the actual number of parts, are not specified in advance (i.e., in the training phase) but determined using the *a contrario* tools, from the (test) image statistics.

In practice, we found that using an independence assumption, while it indeed adapts to the image and performs reasonably well, results in many more false detections than it should. Investigating this problem was actually the motivation for our proposed non-independence based *a contrario* approach. With the independence assumption removed and replaced with a simple second order statistics based model, predictions of the number of false detections became fairly accurate.

4.2 A Part-Based Model

The detection algorithm compares the description of the object category (model) to subimage candidates and decides independently, for every subimage, whether it belongs to the model category.

We consider a relatively simple part-based approach. Both the model and the subimage candidates are represented in the same way, by a set of local descriptors, which

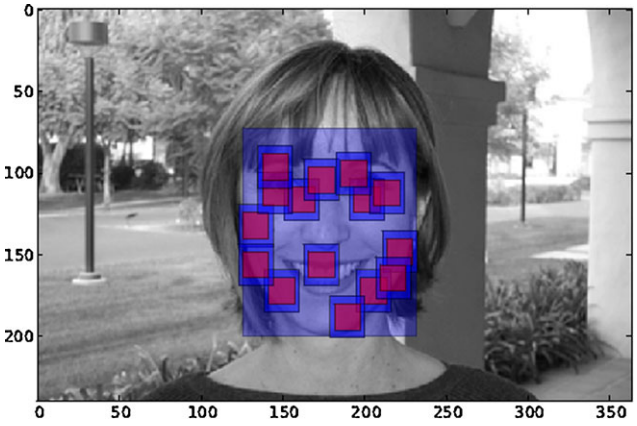


Fig. 2 An example of the part-based model. *Red rectangles* correspond to the image patches, which are used for the calculation of the appearance part descriptors. *Blue rectangles* are the part location regions $\mathcal{S}_k^{\mathcal{M}}$ (Color figure online)

are specified by locations and appearances. Specifically, the model $\mathcal{M}$ consists of K parts, contained in a bounding box: $\mathcal{M} = \{\mathcal{M}_k = \{\mathcal{S}_k^{\mathcal{M}}, \mathcal{A}_k^{\mathcal{M}}\}\ k = 1, 2, \ldots, K\}$, where $\mathcal{S}_k^{\mathcal{M}}$ is the part location and $\mathcal{A}_k^{\mathcal{M}}$ is the part appearance. The dimensions of the bounding box are part of the model as well.

The part location $\mathcal{S}_k^{\mathcal{M}}$ describes the location of the part center in normalized coordinates: every bounding box is mapped to a 1×1 square and every point is associated with normalized coordinates in this square. See Fig. 2 for an example of the model part locations.

The appearance descriptor is calculated from an image patch corresponding to a part. We used the popular histogram of local intensity gradient orientations (HOG) (Dalal and Triggs 2005), which is a simplified version of the SIFT descriptor (Lowe 2004). Following Dalal and Triggs (2005), the descriptors are constructed as follows: A 16×16 patch describing the part is divided into 4 smaller spatial regions, each of size 8×8, denoted *cells*. After smoothing, each cell is described by a local 1D weighted (9 bin) histogram of gradient orientations. The histograms of the four cells are concatenated into a one-dimensional vector. The 36-dimensional vector is normalized to make it more robust to illumination changes and shadows. This vector is the appearance descriptor associated with every part and its corresponding 16×16 image patch. For more details on HOG descriptors, see Dalal and Triggs (2005).

The parts and parameters characterizing the model can be specified in many ways. Both the detection process proposed below and the accompanying analysis would be similar for different model choices. In particular, we could specify the model using a semi-supervised algorithm (see, e.g. Fergus et al. 2007) and optimize its choice so that it would perform best on training data. However, achieving competitive performance was not our goal here, and may not be possible with this relatively simple detection method. Moreover,

following the *a contrario* spirit, we preferred to specify the model with as little training data as possible.

To specify the model, we used only positive example images, with the objects marked by a bounding box. One of these images was arbitrarily selected to be the source of all parts, and all others served as a validation set. Intuition tells us that a good part should be close, in location and in appearance, to regions describing the same part in many images of the same category. All 16×16 patches in the bounding box of the selected image were considered, one by one (in arbitrary order), as candidates to be a model part. A tested patch was selected to be a model part if the following two requirements were met: a. In at least two-thirds of the (30) validation images, there was a patch that was close to the tested patch in appearance and in space. b. The tested patch was not close in space to an already selected model part. Patches were considered to be close in appeareance if Euclidean distance between appeareance descriptors was smaller than 0.05. Patches were considered to be close in space if the maximal difference between both patch center coordinates was less than 0.1.

4.3 The Detection Process

The detection process follows a standard procedure and tests the presence of the object at multiple locations in a given test image. Every location specifies a subimage candidate. Each subimage is described by a set of features, C_n, which, similarly to the model parts, consists of spatial (location) and appearance (histogram of gradients) descriptors.

The set of local descriptors associated with the subimage is compared to the model. The following notation will be useful. We say that the kth part is δ-detected for a given subimage candidate if there is a feature in C_n satisfying two demands: a. its location is close enough (closer than a given threshold ΔS) to the model part's center and b. the distance between its appearance and that of the kth part is smaller than δ. (The metric and the threshold δ are specified below.) When appropriate we omit the explicit reference to the threshold δ.

Detecting many parts provides strong evidence that the subimage candidate is indeed an instance of the model. In practice, some parts are often occluded or significantly deformed. Therefore, we adopt the following simple decision rule:

Decision rule Accept the subimage candidate as an object instance if a sufficient number (K_{min}) of parts are δ-detected. Otherwise, reject this hypothesis.

To use this simple decision rule effectively, we should have a method for setting the threshold δ. The *a contrario* analysis provides this threshold indirectly as explained before. Essentially, given a subimage candidate, the analysis uses the minimal δ value for which this candidate is accepted (most conservative decision), and calculates the expected number of false detections (NFA) associated with this threshold in a background model. The subimage candidate is accepted if the NFA is small enough. Because the NFA depends on the given image, the uniform threshold on the NFA is an image adaptive threshold on the appearance distance. We now give a complete pseudo-code for the detection process.

Formal description of the detection process

Input: an image I, a model with K parts, and a threshold ϵ.

1. Sample the image in a stride of 4 pixels, extract a set of features $\{c_l = (\mathcal{S}_l^c, \mathcal{A}_l^c,)\}_{l=1}^{L}$. $\mathcal{A}_l^c$ is the HOG appearance descriptor calculated over a 16×16 pixel region and $\mathcal{S}_l^c$ is the corresponding location (center of the region).
2. Specify a set of subimage candidates (denoted $\{R_n\}_{n=1}^{N}$) using some dense spatial grid; see the note below. Describe every subimage candidate by the corresponding set of features C_n.

   $$C_n = \left\{c_l : \mathcal{S}_l^c \in R_n\right\}.$$

   Normalize the feature coordinates relative to the candidate subimage so that all normalized spatial coordinates are in [0, 1]. The notation $\mathcal{S}_{l,n}^c$ is used for the normalized coordinates.
3. For every subimage candidate R_n described by C_n ($1 \leq n \leq N$):
   (a) For every model part $\mathcal{M}_k$ ($1 \leq k \leq K$) of the model $\mathcal{M}$, find the distance $d(C_n, \mathcal{M}_k)$ between the subimage candidate and the kth part as follows:
      (i) Find all candidate features that satisfy the following spatial constraints for this part, and denote them by $C_{n,k}$

      $$C_{n,k} \triangleq \left\{c_l : c_l \in C_n \text{ and } \left\|\mathcal{S}_{l,n}^c, \mathcal{S}_k^{\mathcal{M}}\right\|_\infty \leq \Delta S\right\},$$

      where $\|.\|_\infty$ is the L^∞ norm (maximal entry) on the 2D plane.
      (ii) The distance between the subimage candidate and the kth model part is the minimal distance between the appearance descriptors in $C_{n,k}$ and the model part appearance,

      $$d(C_n, \mathcal{M}_k) \triangleq \begin{cases} \infty & C_{n,k} = \emptyset \\ \min_{c \in C_{n,k}} d^A(\mathcal{A}^c, \mathcal{A}_k^{\mathcal{M}}) & \text{otherwise} \end{cases} \quad (4)$$

      where d^A is the L_1-norm distance between appearance descriptors.

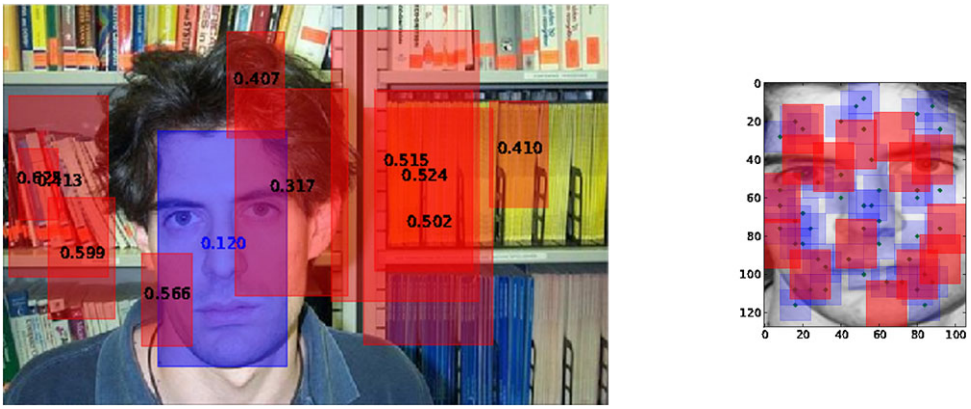


Fig. 3 An example of the detection process. (**a**) A few of the model candidates (*subimages*) extracted from the image. The *number inside the rectangle* is the appearance distance between the model and the candidate. The *blue rectangle* is the best candidate found in this image. (**b**) A closer look at a single candidate. The *blue rectangles* correspond to (some of) the features extracted from the candidate. The *green diamond* markers denote the corresponding feature location. A feature may match a model part if its center is in the corresponding (*red*) region. Note that more than one feature corresponds to every model part (Color figure online)

(b) *Normalize* $d(C_n, \mathcal{M}_k)$ to get $d^{norm}(C_n, \mathcal{M}_k)$ as described in Sect. 4.4.2.

(c) Calculate the distance between the model and the subimage candidate as

$$d_{(K_{min})}(C_n, \mathcal{M}) \triangleq K_{min}\text{th smallest distance of } \left\{d^{norm}(C_n, \mathcal{M}_k)\right\}_{k=1}^{K}. \quad (5)$$

See Fig. 3 for an example of the detection process.

(d) Calculate the expected number of false detections, $NFA(C_n, d_{(K_{min})}(C_n, \mathcal{M}))$, that would occur if we apply a decision rule that accepts this subimage candidate in a random image; see Sect. 4.4. Decide to accept the candidate as a model instance if the NFA is smaller than a threshold ϵ.

Notes

1. For multiscale detection, we repeat this process with a set of scaled versions of the input image. We used $N_s = 8$ scaled versions of the image I with a scaling step of $s = 1.08$ (multiplicative). This modification only makes a difference in calculating the NFA; see Sect. 4.4.4. Otherwise the algorithm is unchanged.
2. As described above, the algorithm is specified up to the distance normalization and the NFA estimation, which are described below.
3. The threshold ϵ is meaningful: it is the expected number of false detections. It can be specified according to the task requirements but does not depend on the model or the image. Therefore it does not need tuning.
4. In our implementation, the candidate subimages have a size of 128 × *width* so that each subimage has the same height/width ratio as the bounding box used for the model. The centers of the candidate subimages are on a 4×4 grid. The other parameters of the algorithm are the spatial uncertainty in part location ΔS, always set as $\Delta S = 0.1$, and the number of parts K_{min}, to which the detection is not very sensitive; see Sect. 5.2.
5. A more elaborate version of the algorithm, not prespecifying the number of parts K_{min}, is possible as well. In this version we simply repeat the detection process with different values of K_{min}. Like the multiscale version, this modification only makes a difference in calculating the NFA; see Sect. 4.4.4. Otherwise the algorithm is unchanged.

4.4 Predicting the NFA with Independent Parts

4.4.1 The Background Image Model

An arbitrarily chosen subimage is usually not similar to the model, and its appearance distance from the model is therefore large. Accepting subimage candidates associated with a large appearance distance to the model, $d_{(K_{min})}(C_n, \mathcal{M})$, implies, intuitively, that candidates which do not correspond to the model still have a high probability of acceptance.

To estimate the false detection probability, we need the distribution of such distances for randomly picked candidates. This distribution is, however, both too complex and unknown. Using the *a contrario* methodology, we replace it with a well-defined probability of a "background model," which is still related to the given image.

This background model does not provide an explicit distribution for full images, and is rather a model for a subimage candidate. Moreover, this model is specified only partially, by the set of the distances $\{d(C', \mathcal{M}_k)\}_{k=1}^{K}$ between an

object model part and the random image candidate C'. These distances, which are regarded as random variables, are assumed to be characterized by a simple joint distribution, satisfying that the K distances are either independent random variables or weakly dependent variables. We start with the classic and simpler case, where the variables are supposed to be independent, and then consider, in Sect. 4.5, the generalized, more complicated, one. For the simpler case, the marginal probability distributions $\{P(d(C', \mathcal{M}_k) \leq \alpha)\}_{k=1}^{K}$ are estimated empirically from a set of subimage candidates $\mathcal{C}$:

$$P\big(d\big(C', \mathcal{M}_k\big) \leq \alpha\big) = \frac{1}{\|\mathcal{C}\|} \#\big\{C'' \in \mathcal{C}, d\big(C'', \mathcal{M}_k\big) \leq \alpha\big\}. \tag{6}$$

The set $\mathcal{C}$ is usually the set of all subimage candidates in the given image. When assuming independence between parts, the (estimated) marginal probabilities completely specify the joint distribution, and therefore suffice for the analysis.

4.4.2 Normalizing $d(C_n, \mathcal{M}_k)$

Calculating the distance between a model and a subimage candidate, as defined in (5), requires that the distances associated with the model parts be sorted. To compare the distances meaningfully, they should be brought to the same scale or, in other words, *normalized*. As in Muse et al. (2006), the *normalized* distance $d^{norm}(C_n, \mathcal{M}_k)$, corresponding to the un-normalized distance $d(C_n, \mathcal{M}_k)$, is defined as:

$$d^{norm}(C_n, \mathcal{M}_k) = P\big(d\big(C', \mathcal{M}_k\big) \leq d(C_n, \mathcal{M}_k)\big), \tag{7}$$

where the probability function P is taken from (6).

An additional advantage of this normalization is that the distance becomes meaningful by itself: it is the empirical probability that we get a distance of at most $d(C_n, \mathcal{M}_k)$ if we pick a random candidate following the background model. Using the notation introduced in Sect. 4.3, it is the probability that the kth part is $d(C_n, \mathcal{M}_k)$-detected in a random candidate. Note that the normalized distance distribution is uniform in $[0, 1]$.

4.4.3 Calculating the Probability of a False Alarm for a Specific Subimage Candidate

Consider a subimage candidate R_n, represented by its set of features C_n. Recall that $d_{(K_{min})}(C_n, \mathcal{M})$ is the K_{min}th smallest distance from a part of the model to C_n, as defined by Formula (5). By the Helmholtz principle, this subimage should be accepted only if its distance from the model is small enough to be very unlikely in a random situation (the background model). Let us write $PFA(C_n, K_{min}, \mathcal{M})$ for the probability that, under the background model, a candidate subimage contains at least K_{min} parts at distance $d_{(K_{min})}(C_n, \mathcal{M})$ from the model. Observing that $P(d^{norm}(C', \mathcal{M}_k) \leq \alpha) = \alpha$, we get,

$$PFA(C_n, K_{min}, \mathcal{M}) = \sum_{l=K_{min}}^{K} \binom{K}{l} \big(d_{(K_{min})}(C_n, \mathcal{M})\big)^l \times \big(1 - d_{(K_{min})}(C_n, \mathcal{M})\big)^{K-l}. \tag{8}$$

4.4.4 NFA Calculation

After computing the probability of false detection for a particular subimage candidate (in the previous section), the next step is to control the number of false detections when the detection procedure is run on the whole image. Calculating the probability of one false detection or more in the image is complex due to the dependencies between the detection events corresponding to the different candidates. Therefore, the *a contrario* methodology uses a simpler criterion: the expected number of false detections in a random image. Recall that N is the number of candidates in a single image associated with the basic (coarsest) scale and with a specific K_{min} value. Then, the expected number of false detections is simply

$$NFA(C_n, K_{min}, \mathcal{M}) = N \cdot PFA(C_n, K_{min}, \mathcal{M}).$$

For the multiscale version let N_s be the number of scaled versions of the input image that are used. These versions are scaled by a multiplicative scale factor s (set as 1.08 in our implementation). The number of candidates decreases by s^2 when scaling down the image. Therefore the total number of candidates is $N(1 + s^2 + s^4 + \cdots + s^{2N_s})$, which by using geometric progression, gives

$$NFA(C_n, K_{min}, \mathcal{M}) = N \frac{1 - s^{2N_s}}{1 - s} \cdot PFA(C_n, K_{min}, \mathcal{M}).$$

Running the detection process with more than one value of K_{min} would also be possible. Let N_k be the number of K_{min} values that are used. Then the number of candidates that are tested is simply $N_k N$ and

$$NFA(C_n, K_{min}, \mathcal{M}) = N_k N \cdot PFA(C_n, K_{min}, \mathcal{M}).$$

The actual decision whether to accept the sub-image candidate as a model instance is made by comparing the NFA to a threshold ϵ, and accepting the sub-image if the NFA is lower than ϵ.

4.5 Predicting the NFA with Dependent Parts

The background model described in the previous section is based on the assumptions that the distances between the can-

didate and the various model parts are i.i.d. random variables. This assumption can naturally be challenged, especially as the parts may be spatially close or even overlapping, which makes their content related. We found indeed that predicting the false detection rate using this simple background model gives inaccurate results. The actual number of false detections in most non-target images is much larger than ϵ, implying that the independence based model is not a good background model for the real image; see Sect. 5.2. Therefore, we now propose a generalized background model, which better fits the task of part-based detection in natural images.

Note that the detection process is almost the same. The only change is in the decision step, where the NFA is calculated in a different way.

4.5.1 A Background Model from a Tree-Based Distribution

As in the independence based analysis, we regard part detection as a random event and calculate the probability of object detection in the background model from the joint detection probabilities of the different parts.

Unlike the classic *a contrario* approach, we consider the case where the detection of different parts may be correlated. We know intuitively that the detections of two parts are more correlated when the spatial distance between them is smaller. This observation is indirectly supported by our experiments.

We describe this correlation model using a graph $G = (V, E)$, where the vertices, $V = \{v_k\}_{k=1}^{K}$, correspond to the model parts, and the weights of the edges E_{ij} are spatial distances between the parts in the model. A set of random binary variables $X = \{X_k\}_{k=1}^{K}$ is associated with V, with X_k taking the '1' value when the kth part is detected and '0' otherwise.

Every pair of variables (X_i, X_j) is, in principle, correlated. To handle the dependencies in a tractable way, we approximate the joint distribution specified by all the correlations (which is a second order approximation of the true distribution) by a simpler one which depends only on some of the stronger correlations. Specifically, we describe the joint probability of part detections $P(X_1, \ldots, X_K)$ by a graphical model where the graph is the minimum spanning tree (MST) of G. That is, the joint probability depends only on the correlations associated with the edges of this MST. Note that while the tree topology is specified by the distances, the joint probability is specified by the actual (estimated) correlations associated with the edges. Assuming that the correlations are monotonically non-increasing with the distance, choosing the distribution specified by this tree (see below) is better than choosing a distribution specified by any other tree, in the sense that the resulting distribution is the best approximation of the true distribution according to the Kullback-Leibler divergence (Chow and Liu 1968). See Fig. 4 for an illustration of the MST model, corresponding to one model image.

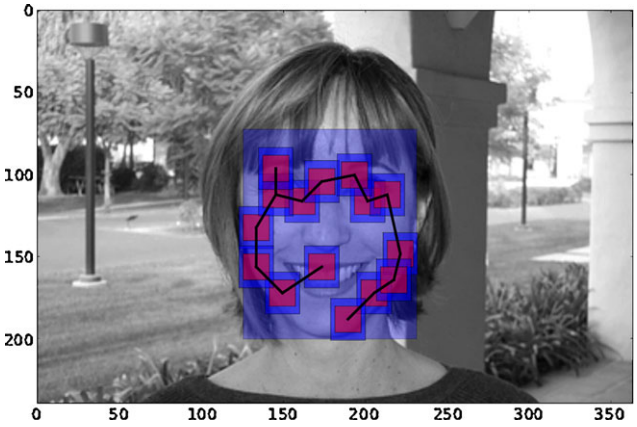


Fig. 4 Minimum spanning tree for the model parts

Thus, the joint probability of the part detection events is specified by:

$$P(X_1, \ldots, X_K) = P(X_{root}) \prod_{k=1, k \neq root}^{K} P_C(X_k), \tag{9}$$

where

$$P_C(X_k) = P(X_k | X_{parent(k)}), \tag{10}$$

and where *root* is some vertex of the MST that is chosen to be the root, and *parent*(k) is a parent of a vertex (part) k in the MST. Recall that the joint probability of a directed graphical model does not depend on the choice of the root (Pearl 1988).

4.5.2 Estimating the Empirical Distributions $P(X_k | X_{parent(k)})$

The variables $\{X_k\}$ are binary. Therefore every distribution $P_C(X_k)$ is specified by 2 probabilities, $P(X_k = 1 | X_{parent(k)} = 0)$, $P(X_k = 1 | X_{parent(k)} = 1)$. For brevity we sometimes refer to the values of these probabilities as $P(1|0)$, $P(1|1)$, etc. Both the $P_C(X_k)$ and the $P(X_{root})$ distributions are empirically estimated from the image itself. Note that the MST specifying the graphical model does not depend on the threshold δ but the detections do. Therefore, the distributions are estimated for a range of δ values. Let $X_{k,m}$ be the indicator

$$X_{k,m} = \begin{cases} 0 & d^{norm}(C_m, \mathcal{M}_k) > \delta \\ 1 & d^{norm}(C_m, \mathcal{M}_k) \leq \delta \end{cases} \tag{11}$$

corresponding to the threshold δ, the kth part, and the candidate C_m $(1 \leq m \leq N)$. We use the index m (and not n) here

to emphasize that these candidates are used to construct the empirical distribution, and not to test a particular candidate.

The empirical probabilities, $P_\delta(X_k|X_{parent(k)})$ and $P_\delta(X_{root})$, are estimated by:

$$\begin{aligned} &P_\delta\big(X_k = x_k|X_{parent(k)} = x_{parent(k)}\big) \\ &= \frac{|\{m : X_{k,m}{=}x_k,\ X_{parent(k),m} = x_{parent(k)}, 1 \le m \le N\}|}{|\{m : X_{parent(k),m} = x_{parent(k)}, 1 \le m \le N\}|}, \\ &P_\delta(X_{root} = x_{root}) = \frac{|\{m : X_{root,m} = x_{root}, 1 \le m \le N\}|}{N}, \end{aligned} \tag{12}$$

where $x_k, x_{parent(k)}, x_{root} \in \{0, 1\}$.

Note that the MST structure (connectivity) depends on distances and is therefore completely identical for evaluating different hypotheses in the same image and also in different images. The MST correlations and the implied conditional probabilities depend on the particular image. They will be the same for all hypotheses in the same image but may be different from image to image, which is an advantage, because this way the decision process adapts itself to image similarities.

4.5.3 PFA Calculation with Dependent Parts

Suppose that the joint distribution, $P(X_1, \ldots, X_K)$, is calculated as described above. Recall that the model is detected when at least K_{min} parts of the model are detected. Then $PFA(C_n, K_{min}, \mathcal{M})$ is the sum of joint assignments of $\{X_k\}_{k=1}^K$ satisfying $\sum_{k=1}^K X_k \ge K_{min}$. From (9), it follows that

$$\begin{aligned} &PFA(C_n, K_{min}, \mathcal{M}) \\ &= \sum_{\sum_{k=1}^K X_k \ge K_{min}} P(X_1, \ldots, X_K) \\ &= \sum_{\sum_{k=1}^K X_k \ge K_{min}} P_{d_{(K_{min})}(C_n,\mathcal{M})}(X_{root}) \\ &\quad \times \prod_{k=1, k\ne root}^{K} P_{d_{(K_{min})}(C_n,\mathcal{M})}(X_k|X_{parent(k)}), \end{aligned} \tag{13}$$

where the sum is over all vector assignments satisfying $\sum_{k=1}^K X_k \ge K_{min}$. As before, $NFA(C_n, K_{min}, \mathcal{M}) = N \cdot PFA(C_n, K_{min}, \mathcal{M})$. We will see in the experiments (Sect. 5.2) that using the MST background image model gives much more accurate predictions than when assuming independent parts.

4.5.4 Fast $PFA(C_n, K_{min}, \mathcal{M})$ Calculation

Straightforward calculation of $PFA(C_n, K_{min}, \mathcal{M})$ requires evaluating the sum of approximately 2^K different detection combinations and might be prohibitive for a large number of parts. Note that $PFA(C_n, K_{min}, \mathcal{M})$ must be calculated online, as part of the detection phase, because it is based on the given image. We now describe an efficient yet precise algorithm for PFA calculation in the MST model.

Our goal is to calculate the probability of detecting at least K_{min} parts, $PFA(C_n, K_{min}, \mathcal{M})$. We solve this problem by solving another problem: calculating the probability of detecting exactly k parts, where $0 \le k \le K$. The latter problem is solved for all k's simultaneously. Similarly to the simpler model described in Sect. 3, the solution is recursive. It decomposes a tree into the root and several unconnected subtrees. Each subtree contains a single child of the root. The probability of detecting k parts is calculated for each subtree and then merged to get the result for the whole tree. The following terms are used:

- T: MST with K nodes/parts.
- i: $(1 \le i \le K)$, an index, corresponding to one model part and one node in the MST.
- X_i: A binary variable describing whether the part is detected.
- $ch(i)$: The set of children of node i, as specified by the tree structure, and the choice of the root.
- T_i a subtree containing the node i as a root, and all its descendants.
- $P_k(T_i|x_i)$: The probability to detect exactly k parts in the subtree T_i conditioned on its root value $x_i \in \{0, 1\}$.
- $P(T_i|x_i)$: The $(K+1)$ dimensional vector of probabilities $P(T_i|x_i) = \{P_k(T_i|x_i)\}_{k=0}^K$.
- $P_k(T)$: The probability to detect exactly k parts in model tree T.
- $P(T)$: The $(K+1)$ dimensional vector of probabilities $P(T) = \{P_k(T)\}_{k=0}^K$.
- δ_{x_i} is a $(K+1)$ dimensional vector where all entries are zero except one entry which is 1. For $x_i = 0$, it is the first, while for $x_i = 1$ it is the second.

From the definitions above, it follows that

$$PFA(C_n, K_{min}, \mathcal{M}) = \sum_{k=K_{min}}^{K} P_k(T)$$

$$\begin{aligned} P(T) = {} & P(T_{root}|X_{root} = 0) \cdot P(X_{root} = 0) \\ & + P(T_{root}|X_{root} = 1) \cdot P(X_{root} = 1). \end{aligned}$$

$P(T_{root}|x_{root})$ is defined recursively by calculating $P(T_i|x_i)$ for all children of the *root*. Each $P(T_i|x_i)$ is calculated in the same way. Recall that the distribution of a sum of two random variables is the convolution of the two original distributions. The number of detected parts in a tree is the sum of the number of detected parts in the subtrees and the tree's root. Therefore, given the distributions of the number

of detected parts in each of the subtrees as well as the distribution of the root, they are convolved to get the distribution of the number of detected parts in the tree. Explicitly,

$$P(T_i|x_i) = \delta_{x_i} * \big(*_{n\in ch(i)}\big(P(T_n|X_n=0) \times P_\delta(X_n=0|X_i=x_i) + P(T_n|X_n=1)P_\delta(X_n=1|X_i=x_i)\big)\big), \quad (14)$$

where $*$ is the convolution operator. The inner convolution is between all children of the node i, and the outer one corresponds to the convolution with the root distribution. Note that as we calculate conditional distributions, the value x_i is known and therefore the corresponding distribution δ_{x_i} is known (and specified above). The final algorithm is again a kind of dynamic programming algorithm where the probabilities are calculated by backward induction starting from the leaves. For every node we need to calculate only two probability vectors $P(T_i|x_i)$; one for $x_i = 0$, and one for $x_i = 1$. Therefore the calculation is extremely efficient. In our experiments we found that it takes no longer than 4 ms (Python implementation, with 2.8 GHz CPU) for a model of up to 30 parts.

5 Experiments

This section describes the empirical testing of the proposed detection procedures.

5.1 Line Segment Detection Using Second Order Statistics

The proposed approach for line segment detection differs from the original method (Desolneux et al. 2000) in that it uses second order dependencies. These are inferred from the image at hand. As demonstrated below, taking second order statistics into account yields results that are significantly different from the original method. We run every test using the two versions described in Sect. 3:

1. The original *a contrario* method (Desolneux et al. 2000), based on the independence assumption (and a uniform distribution of orientations). It is denoted here *original*.
2. The proposed method, which uses a dependence model learned from the image. It is denoted as *dependent*.

For both methods, the precision at which a pixel is considered aligned with a segment is chosen to be $p = 1/16$. Unless otherwise specified (as in Experiment 4, Sect. 5.1.2) the detection is performed by considering evidence from pixels that are 2 pixels apart (as in Desolneux et al. 2000).

For the *dependent* method, edge orientation statistics are calculated from the test image. Recall that for a given line and a pixel with index i, we write X_i for the variable equal to one if the pixel is aligned with the line (up to precision p) and 0 otherwise. From now on, we will call X_i the *consistency* of pixel i (the pixel i is said to be *consistent* with the considered line if $X_i = 1$). All lines that start and end on the image borders were considered for the computation of these statistics. In most experiments, where a sampling step of 2 was used, all pixel pairs whose locations along the line differ by 2 were used to estimate the second order statistics. That is, we used the pixel pairs for which there is exactly one pixel on the line between the two pixels in the pair. Then, $P(X_i = 1, X_{i-1} = 1)$, for example, is estimated by the fraction of pairs taking values $(1, 1)$. The statistics estimates were averaged over all lines. We first give some numerical values for these estimated statistics in Sect. 5.1.1. Then, in Sect. 5.1.2, we show detection experiments on several images. Finally, in Sect. 5.1.3, we investigate the validity of the first order Markov assumption on orientations as well as the potential of higher order modeling.

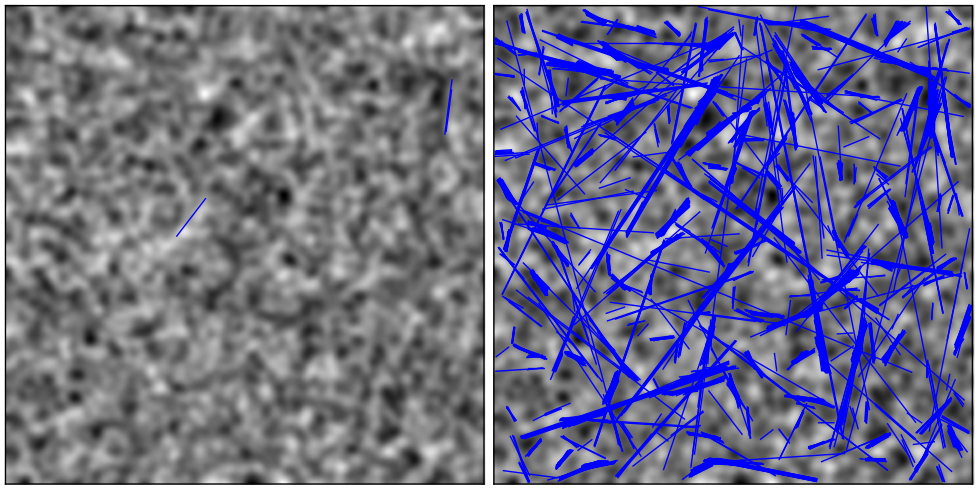
Table 1 Estimated conditional probability of orientation consistency at adjacent pixels, for various images. The "Lena" image is the classic 512×512 version. The "land" and "wall" images are visible in Fig. 6, bottom, and 7, bottom, respectively. The colored noise is obtained by filtering a white noise image with a Gaussian kernel with $std = 5$

	$P(0\|0)$	$P(0\|1)$
Lena	0.9465946	0.780059
Land	0.943820	0.815936
Wall	0.943294	0.823705
Colored noise	0.971864	0.417353

5.1.1 Orientation Dependency

As a first result, we display the conditional probability estimated on several images by the procedure described above. We write $P(0|0)$ for $P(X_i = 0|X_{i-1} = 0)$, etc. In all cases, the estimated value $P(X_i = 1)$ was, not surprisingly, very close to the theoretical $p = 1/16$ value. As may be seen from Table 1, the second order statistics differ for images with different content. $P(0|0)$ was usually around 0.95, which is a little higher than $P(0)$. $P(1|1)$ varied significantly, roughly from 0.1 to 0.6. The estimated probability for two adjacent pixels to both be consistent is much greater in natural images than under the independence assumption. Since we consider pixels to be adjacent when they are at a distance two apart, this observation is not due to the 2×2 neighborhood used to compute the gradient.

From these values, we can also perform a χ^2 test on the consistency at adjacent pixels (at a distance of 2 apart). For the land image in Fig. 7, bottom, using 5.4M samples and the values from Table 1, we conclude the non-independence of samples with probability one, up to machine precision. Such extreme confidence in the test is due to the high number of samples. This, in addition to detection experiments



(a) Dependent (b) Original approach

Fig. 5 Line segment detection in colored noise ($\sigma = 5$) with the two algorithms described above, using $\epsilon = 1.0$

Table 2 Number of detected segments in colored noise obtained from Gaussian kernels with varying standard deviation σ, using both the original and the dependent methods (average over 20 runs)

σ	1	3	5	10	20	50
Number of segments—original	0.0	174.9	2923.0	20918.9	37011.0	32443.0
Number of segments—dependent	0.0	0.0	0.5	2.3	47.9	1895.0

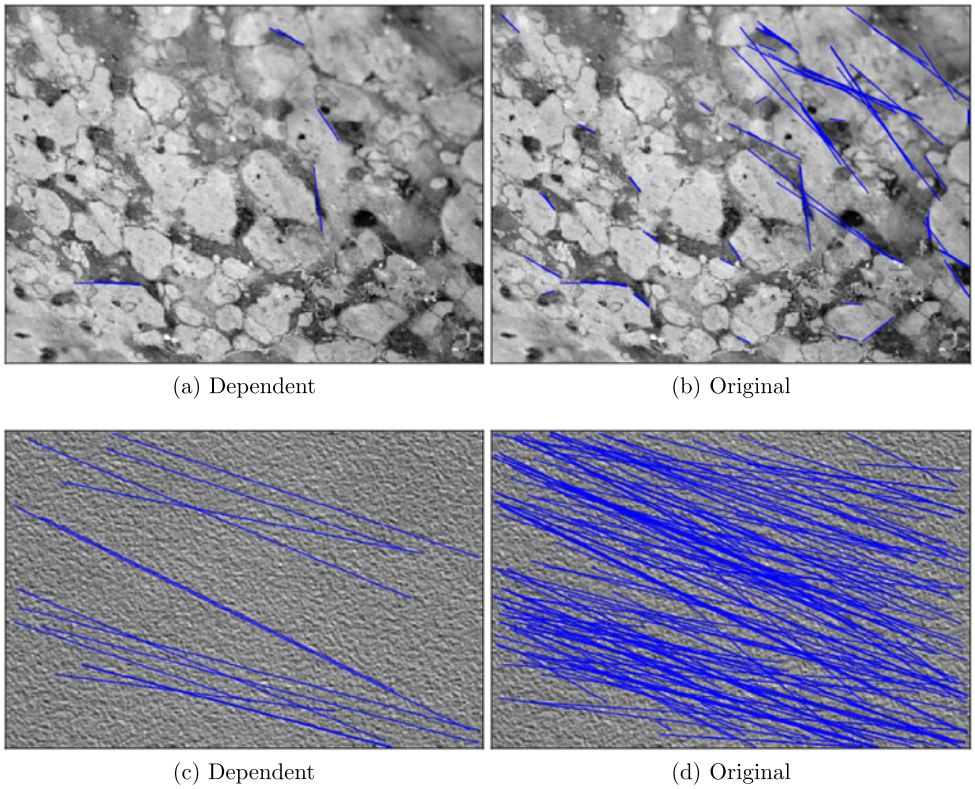
in colored noise displayed in the next section, justifies the modeling of the dependency between pixels to detect segments.

5.1.2 Detection Experiments

In this section, we compare the two detection methods (dependent and independent) on several images. In all experiments and for both methods, after segment detection, the maximally meaningful segments were retained, following the same procedure as in Desolneux et al. (2000). The same value $\epsilon = 1.0$ was used for all experiments. Unless otherwise specified (that is, in experiment 1), pixels where the gradient magnitude was very small (smaller than 2) were not considered as supporting the segments (i.e., $X_i = 0$). Neither were they taken into account for calculating the statistics for the *dependent* method.

Experiment 1: Line Segment Detection in a Colored Noise Image In this experiment random colored noise images are generated by convolving white noise images with a Gaussian kernel with standard deviation σ. In Fig. 5 a detection experiment with $\sigma = 5$ is shown. The original method detects many line segments (about 3000) while the proposed dependency based method detects only 3 for this noise realization. In this case, detecting no or very little segments seems reasonable, although we are not aware of any psychological study about the visual perception of such structures in colored noise. In Table 2, we display the number of segments that are detected in realizations of colored noise for Gaussian kernels having standard deviation ranging from 1 to 50, both for the original and the dependent methods. Results are averaged over 20 runs. The original method falsely detects many segments as soon as $\sigma = 3$, whereas false detections using the Markov assumption only become numerous at $\sigma = 20$. In order not to depend on the dynamic of the colored noises, all detection experiments were performed without any threshold on the gradient magnitude.

Experiment 2: Line Segment Detection in Real Texture Images Interestingly, colored noise is a realistic model for a certain class of textures, the so-called *micro-textures* or *random phase textures*: see Galerne et al. (2010). Such textures are completely characterized by the modulus of their Fourier transform. An example of a wallpaper image falling into this category is shown in Fig. 6. In this case, the proposed method detects far fewer segments, which seems justified. We repeated the line detection experiment with more structured texture images, taken from the UIUC texture database. We found that the dependency based method functions better than the original in the sense that it detects fewer line segments; see an example in Fig. 6. While the absence of linear structure is perhaps questionable in this case, it is clear that many of the lines detected by the original method are unjustified.



(a) Dependent (b) Original

(c) Dependent (d) Original

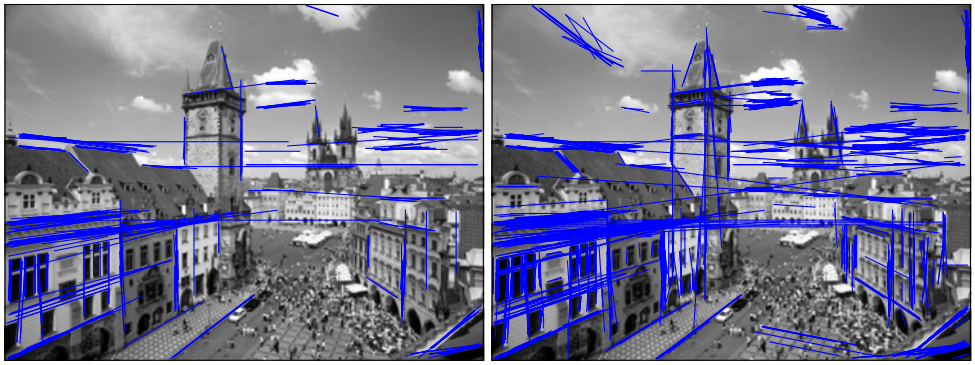
Fig. 6 Line segment detection in textures with the two algorithms described above, using $\epsilon = 1.0$ in all experiments. The *first row* is a texture from the UIUC database. The *second row* is a random-phase texture

Experiment 3: Line Segment Detection in Natural Images We also demonstrate the difference between the detection patterns in a natural image. It is sometimes difficult to tell a false detection from a true one. See, for example, Fig. 7. Both methods yield different results on this image, the dependent one being more conservative in the detection. Some spurious segments, as in the clouds, are no longer detected when modeling dependencies, but some segments related to rectilinear structures, as on the ground, disappear. Using the learned conditional probability, one may also compute the minimal length a segment should have to be detected, as explained in Sect. 3. For this image, one obtains minimal lengths equal (after rounding) to 10 and 15 pixels when using the independent and dependent methods, respectively. Here too, the dependent approach is more conservative than the independent one.

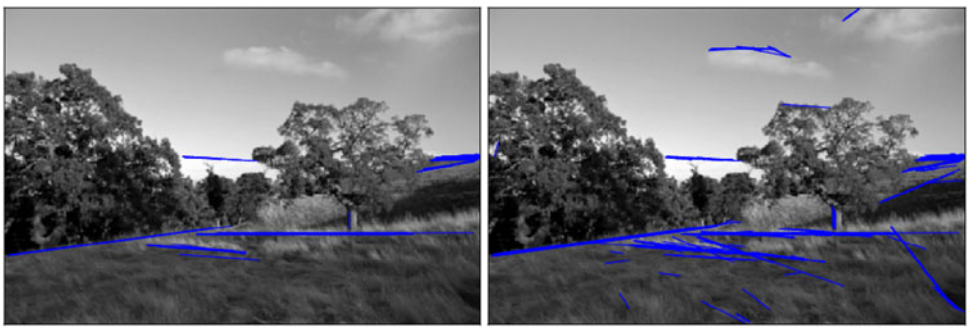
Experiment 4: Line Segment Detection Without Subsampling We also tested the behavior of the proposed method in a case where the orientations to be grouped are structurally not independent. To do so, we run segment detection by using all pixels on each line and not only pixels that are 2 pixels apart. In this case, orientations of neighbor pixels are dependent also because gradients are computed using a 2×2 mask. The first order Markovian assumption provides an approximation for this dependency. In order to run the detection using the dependent method, we estimated the second order statistics without subsampling as well. For the image of Fig. 7, the estimate of $P(1|1)$ is equal to 0.2830 when considering all pixels and to 0.1819 when considering pixels at a distance of 2 apart. We can see in Fig. 8 that the independence assumption yields an over-detection, while the dependency-based method suffers less from the structural dependency. In fact, the proposed *dependent* method seems to offset the dependency similarly as the *original* method when using subsampling, as may be observed in the clouds for instance. Experimenting with other images of natural scenes and textures, we found that the improvement is not systematic, but is significant in most cases. Observe that a similar experiment is performed in Grompone von Gioi et al. (2008), where it is shown that considering all points with the *original* method yields more details than using sub-sampling. However, it is also shown on the examples considered in this paper that the over-detection rate does not increase by much.

Experiment 5: Line Segment Detection and Blur Structural dependency also occurs in the presence of blur. Here we test how the method behaves when applied to an image containing an out-of-focus background. In Fig. 9, we see that the dependent method resists this type of degradation better than the original one, and accepts fewer spurious segments. This is expected because, visually, the out-of-focus background could be described as colored noise.

To summarize, in Sect. 5.1.2 we compared two approaches to line detection: the original independence based approach and the proposed approach, which uses 2nd order

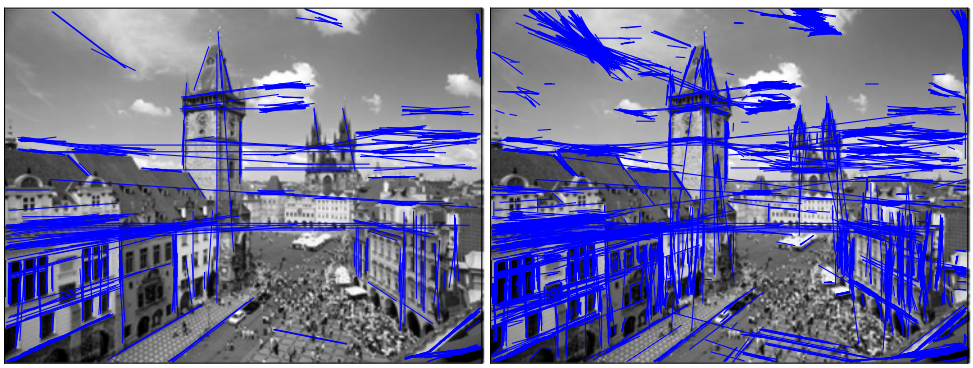


(a) Dependent (b) Original



(c) Dependent (d) Original

Fig. 7 Line segment detection in an image of the Old Town Square in Prague (*top*) and in a natural outdoor image (*bottom*), with the two algorithms described above, using $\epsilon = 1.0$



(a) Dependent (b) Original approach

Fig. 8 Line segment detection without subsampling with the two algorithms described above, using $\epsilon = 1.0$ in both experiments

(a) Dependent (b) Original approach

Fig. 9 Line segment detection in blurred images. On this image with an out-of-focus background, the *dependent* method detects fewer spurious segments than the *original* one

statistical modeling. We demonstrated that the proposed line detection algorithm detects fewer spurious lines in colored noise, real textures and blurred image regions. A natural question is whether using a higher order model is more adequate. We consider this question in the next subsection.

5.1.3 Testing the Markov Assumption

Here we describe two experiments that further investigate the dependency between orientation consistency at nearby pixels. We shall be interested in the orientation consistency in sets of pixels sampled along the line. In the first experiment we shall use it to test whether the observed discrete line is possibly a discretization of a continuous Markov process. In the second experiment we come back to the discrete domain and test whether the observed process is indeed a first order Markov process or possibly a Markov process of higher order.

We shall use the following notation. Let $i_1, \ldots, i_n$ be a sequence of indexes, corresponding to pixels on a line sampled at the default interval (i.e., 2). We shall be interested in some low dimensional particular cases of the joint probabilities $Pr(X_{i_1} = \epsilon_1, \ldots, X_{i_n} = \epsilon_n)$ associated with these sampled pixels and with values $\epsilon_i \in \{0, 1\}$. For these cases we estimates the probabilities as empirical frequencies over all sequences along some line and average the results over all lines (in all directions), in the same way as it is done for estimating first order conditional probability of orientation consistency.

Continuous Markov Chains We would like to test here whether the observed discrete orientation consistency process is a discretization of a continuous Markov process. Our motivation is to get a more complete understanding of the statistical properties of the observed process, and in particular to test how these properties depend on the sampling step.

A (stationary) continuous stochastic process $X(t)$ is a continuous Markov chain if the following condition is satisfied:

$$Pr\big(X(t_n) = \epsilon_n | X(t_{n-1}) = \epsilon_{n-1}, \ldots, X(t_1) = \epsilon_1\big) = Pr\big(X(t_n) = \epsilon_n | X(t_{n-1}) = \epsilon_{n-1}\big),$$

whenever $t_1 < \cdots < t_n$. When the state space is binary, that is when $\epsilon_i \in \{0, 1\}$, this hypothesis implies that

$$Pr\big(X(t) = 0 | X(0) = 0\big) = \frac{1}{a+b}\big(af(t) + b\big),$$

and

$$Pr\big(X(t) = 1 | X(0) = 1\big) = \frac{1}{a+b}\big(a + bf(t)\big),$$

where $f(t) = \exp(-(a+b)t)$, and a and b are infinitesimal transition probabilities; see Grimmett and Stirzaker (2001, p. 260).

Neglecting the necessary low-pass filtering in the sampling operation, such a continuous underlying model for the consistency implies that

$$Pr(X_n = 0 | X_0 = 0) + Pr(X_n = 1 | X_0 = 1) - 1 = \exp\big(-c(a+b) \cdot n\big), \tag{15}$$

where c is a constant related to the sampling step. In order to check this hypothesis, we plot, in Fig. 10, log-linear plots of this sum of conditional probability as a function of n for a natural image (the land image was chosen because it does not contain prominent alignments) and for a colored noise image (again with $std = 5$). These plots shows that while the assumption that the process may be regarded as a discretization of a continuous Markov process is reasonably satisfied for the colored noise image (despite some erratic oscillations), it is not a very accurate model for the natural image. It seems that the statistics of the natural image, however, exhibit linear behavior in log-log plots (Fig. 10), which implies a power law distribution for the combination of consistency given by Formula (15). Such statistics, observed also for other natural images, are possibly explained by the scaling invariance of images, which is at the origin of many power law behaviors in the statistics of natural images (Ruderman 1994).

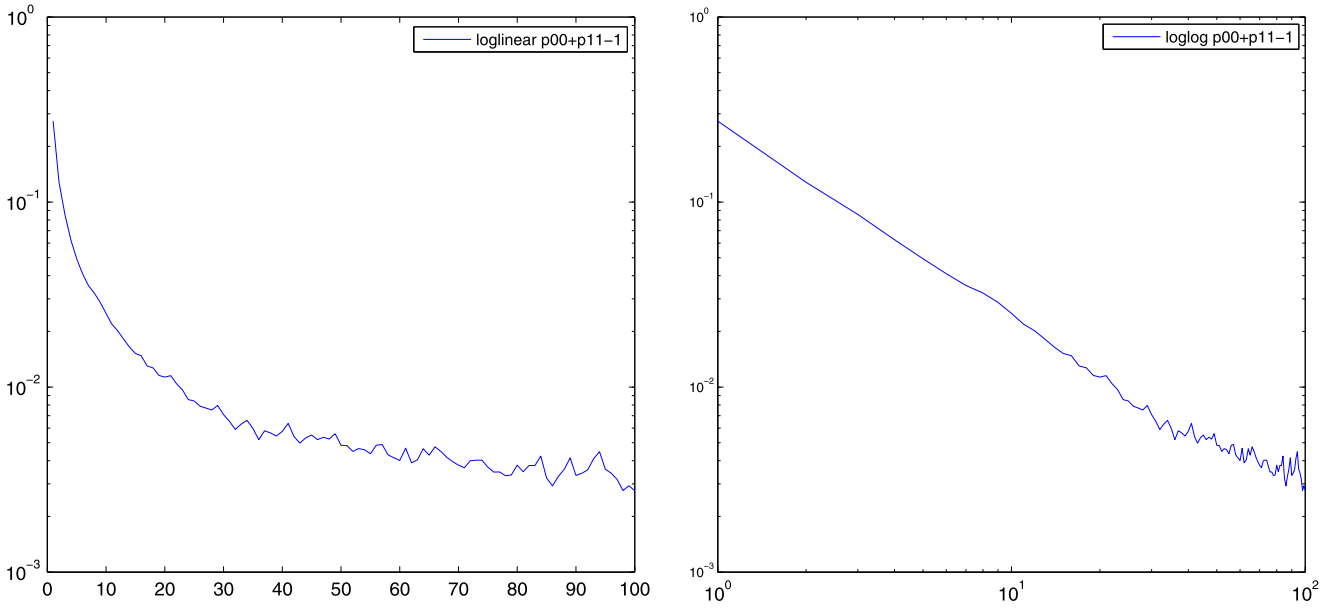
First Order Discrete Markov Model Additional investigation of the Markov property may be carried out simply by estimating the probabilities

$$Pr(X_2 = \epsilon_2 | X_1 = \epsilon_1, X_0 = \epsilon_0).$$

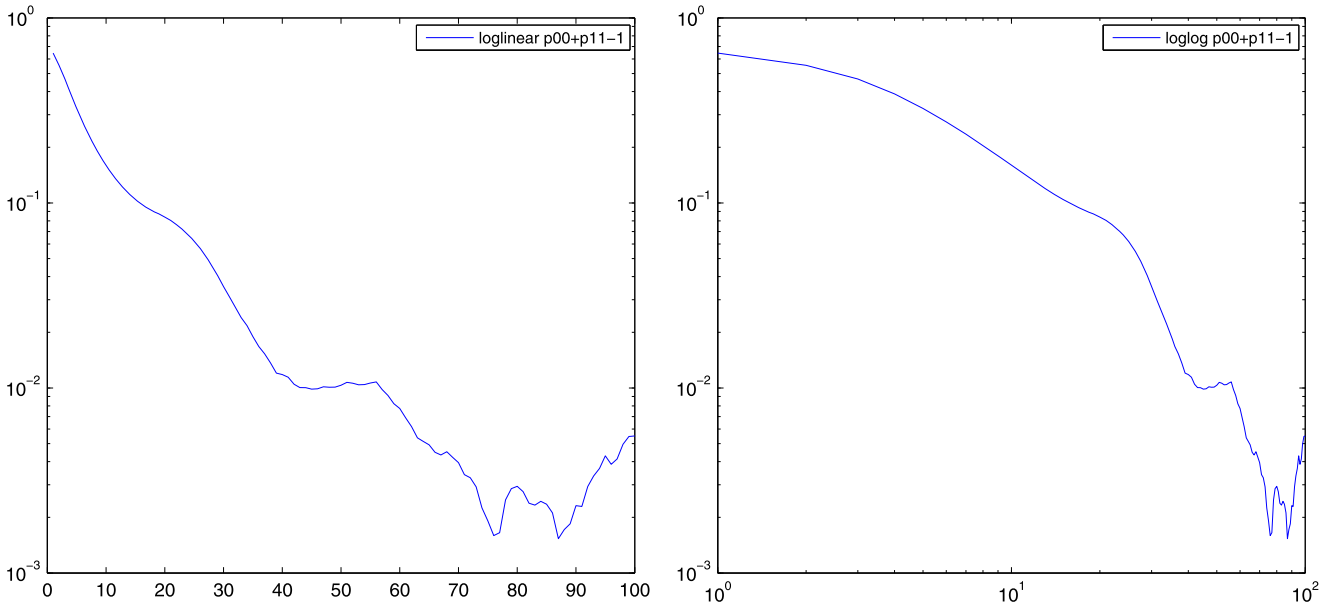
The estimates, given in Table 3, demonstrate that the probability of getting a certain value does not depend only on the previous sample but also on the one before it. Again these estimates imply that modeling the data using a first order Markov model is not precise. This observation was confirmed on other images. We do not display the third or-

Table 3 Second order conditional probability estimated from two images. $P(0|0,0)$ stands for $P(X_i = 0 | X_{i-1} = 0$ and $X_{i-2} = 0)$, etc.

	$P(0\|0,0)$	$P(0\|1,0)$	$P(0\|0,1)$	$P(0\|1,1)$
Land	0.940560	0.912470	0.929588	0.885698
Colored noise	0.958440	0.701535	0.937086	0.526415



(a) Land Image



(b) Colored Noise

Fig. 10 Conditional probability as a function of the distance for the land image (*top*) and a colored noise with standard deviation 10 (*bottom*). On the left: log-linear plot of the sum of conditional probability given by Formula (15), on the right: log-log plots of the same quantity. See the text for an interpretation

der conditional probability here, but it turned out that the consistencies are also not fully captured by a second order Markov model. Therefore, and even though the first order model is beneficial as a background model, it would be useful to investigate segment detection using higher order Markov models.

In conclusion, the first order modeling of dependency between consistencies is a reasonable working hypothesis, leading to a background model that enables robust segment detection. Although the background model is not supposed to fully model the image, the first order Markov model may nonetheless be a bit too crude. Finer, higher order Markov models or scale invariant geometric image models such as Gousseau and Roueff (2007), could also be considered for *a contrario* structure detection.

5.2 Predicting False Detection Rate in Object Detection Using the *A Contrario* Method

In this section we test the false detection rate obtained by a part-based detection method that relies on the *a contrario* approach. This example is important in the context of the generalization of *a contrario* methods proposed in this paper: it demonstrates that applying *a contrario* decisions in the classic, independence based way leads to an excessive number of false detections. Working with the generalized algorithm provides results in agreement with the predictions.

The detection algorithm itself, described in Sect. 4, is relatively simple and is not competitive with state-of-the-art methods. It is, however, characteristic of many computer vision algorithms that use overlapping, and therefore dependent, image data. Because this is, to the best of our knowledge, the first use of *a contrario* background models for object detection, we added, in Sect. 5.3, an empirical study demonstrating that decisions relying on a background model and NFA thresholding are more stable and adaptive that the usual decisions based on similarities.

We performed a number of experiments to test how well the *a contrario* based detection algorithms approximate the empirical false detection rate. All the experiments below use a face category model from the Caltech4 dataset[1] and were conducted on 400 different images of non-face categories (e.g., motorbikes) from this dataset.

FA Prediction Based on the Independence Assumption First we tested the standard *a contrario* model, which is based on the assumption that model part detections are independent (Sect. 4.4). The test was carried out for various part numbers and various ϵ values. A false detection is an event where a candidate is falsely detected as a model instance. The empirical number of false alarms is the total number of false detections in all test images, divided by the number of images, and is denoted $eNFA_T$ (e-empirical, T-total).

Ideally, we expect that the empirical number of false detections, $eNFA_T$, will be close to or lower than the specified rate ϵ. We found, however, that $eNFA_T$ was much higher than ϵ. See Fig. 11. This indicates that the independence assumption is not valid, and the independence based background model does not adequately account for real image dependencies. This is indeed expected, due to the proximity of the parts. Note also that the prediction accuracy decreases with the number of parts. This implies that working with a more complex model (more parts), in an effort to gain better discriminative power, reduces predictability.

The experiments were carried out on a set of 400 images. This makes the empirical validation of the predictions for small ϵ (smaller than 0.005) values highly inaccurate, which is apparent from the plots.

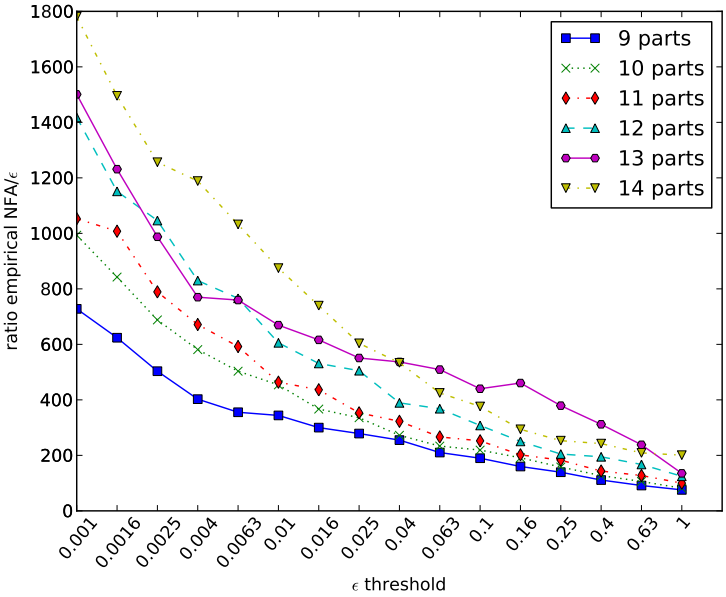
Usually several nearby candidates are detected together and, as they correspond to the presence of a single visual event (either an object instance or not), their number is often not of practical interest. Therefore, we also refer here to distinct false detection, where one detection corresponds to one or more adjacent candidates that are falsely detected. By adjacent we refer to candidates corresponding to neighbors on the location grid. The empirical distinct false detection rate is the number of such events in a set of images, divided by the number of images, and is denoted $eNFA_D$ (D for distinct).

For practical reasons, we may want to change the decision so that it controls the number of distinct detections (and not number of simple detections). To that end, we observed that the number of detections in a detection group is roughly constant and depends on the candidate density (i.e., on N) but is largely insensitive to other parameters such as the image type, or the number of parts K_{min}. In the experiments described here, where the candidate locations differ by 4 pixels, the empirical average number of detections in a detection group was $GS = 2.7$. Therefore, to get ϵ distinct detections, we should aim for about $GS\epsilon$ simple detections. Consequently, we divide the NFA by GS before we test it against the threshold ϵ, which now becomes an estimate of the number of distinct detections. Figure 12 compares the empirical distinct false detection rate $eNFA_D$ with the specified number ϵ. Clearly, the prediction is very inaccurate in this case as well, showing that the independence assumption is indeed not satisfied.

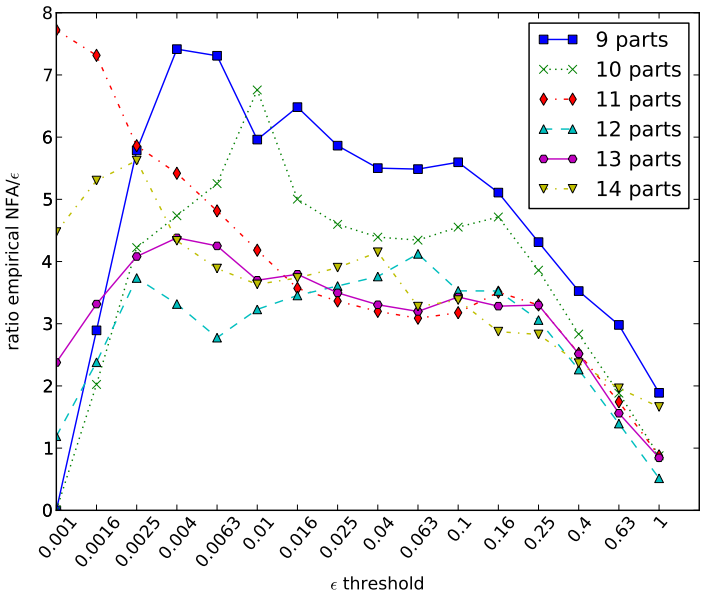
FA Prediction Based on the Tree Model The next experiment considers the same detection task, but this time with the modified/generalized *a contrario* method, whose background model is specified by a tree-based graphical model. (See Sect. 4.5 for a detailed description of the model.) The results are described in Fig. 11 (simple detections) and in Fig. 12 (distinct detections). Clearly, the results are better in two respects: First, the empirical false detection rates are much closer to the specified rates (which are always ϵ). We believe the false alarm detections to be very accurate, especially if we recall that they were made with no training at all. A second observation is that the predictability does not change much with the model size (the number of parts) and within a large range of the specified number of false detections (ϵ). This provides evidence that the number of parts and the specified NFA are both taken "correctly" into account by the background model and the implied NFA calculation.

This section provides additional experimental evidence that using a dependence based background model may lead to more predictable *a contrario* decisions. In contrast to the

[1] http://www.robots.ox.ac.uk/~vgg/data/data-cats.html.



(a) Prediction based on the independence assumption.



(b) Prediction based on the second order model.

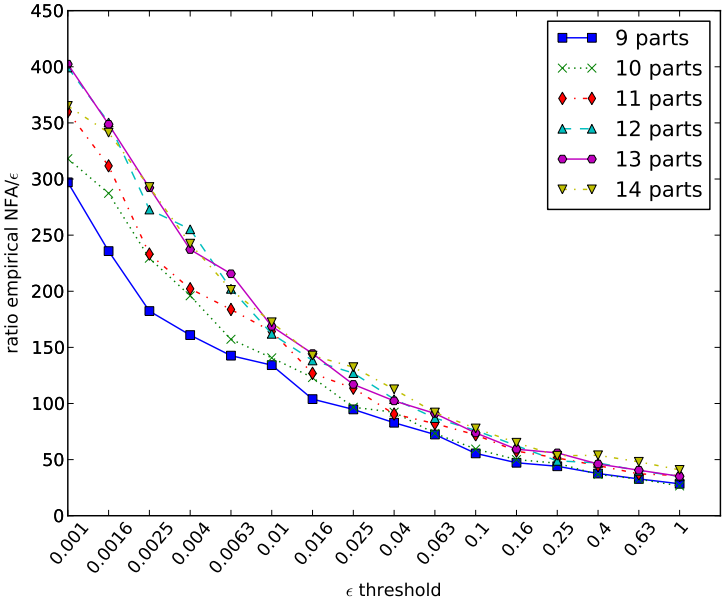
Fig. 11 Prediction of the average number of false face detections using the independence assumption and the second order model: the average number of false face detections in the non-target (motorbike) images vs. the predicted number. The *x-axis* shows the threshold on the number of false alarms per image (ϵ). The *y-axis* shows the ratio between the empirical NFA, $eNFA_T$, and the predicted NFA, which is ϵ. As can be seen, the empirical results are close to the predicted ones in the tree-based model and above two orders of magnitude for the independence assumption

visual evidence provided in the line detection experiments, we here draw quantitative conclusions. We also found that when using the other (non-face) categories for models, using 2nd order statistics in the *a contrario* decision always improved false detection predictability.

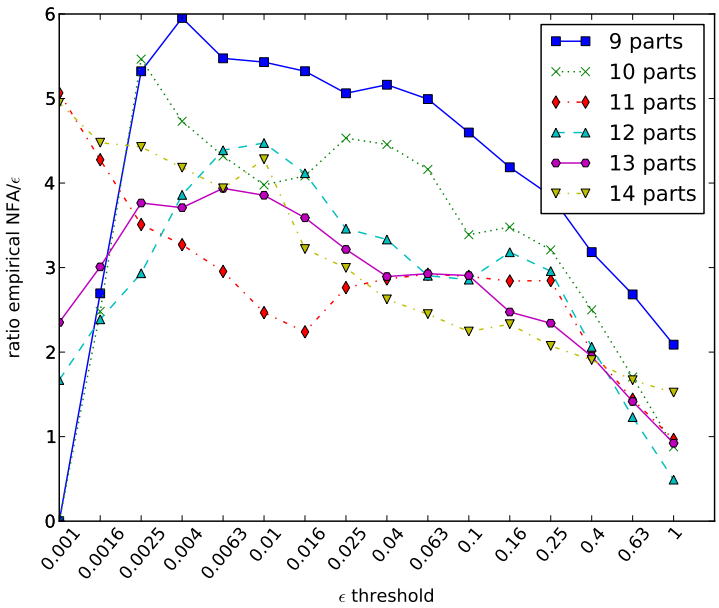
5.3 Thresholding NFA is More Predictable than Thresholding Appearance Distance

The *a contrario* based detection algorithm works by calculating the expected number of false detections and thresholding it. This contrasts with more traditional approaches that work by thresholding, say, model-to-image distances. In this section we show that using the NFA as a decision function is advantageous not only in the *a contrario* detection context but also in training based detection.

To make the comparison, we consider two alternative decision functions that are related to the proposed algorithm but that also use a training phase. Like the proposed algorithm (Sect. 4), they both use the appearance similarity between the HOGs of the model parts and the HOGs of the candidate. The first is an NFA based algorithm. It uses the appearance similarity to independently calculate the NFA for every image in the training set and then de-



(a) Predictions based on the independence assumption.



(b) Prediction based on the second order model.

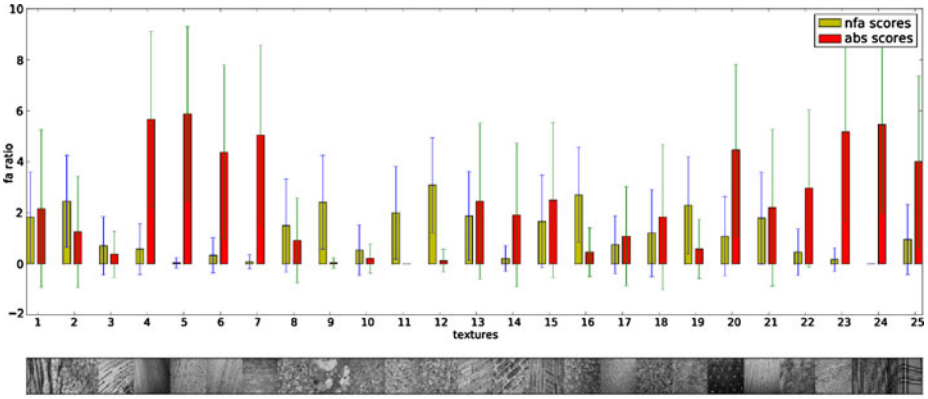
Fig. 12 Prediction of the average number of *distinct* false detections using the independence assumption and the second order model: the average number of false face detections in the non-target (motorbike) images vs. the predicted number. The *x-axis* shows the threshold on the number of false alarms per image (ϵ). The *y-axis* shows the ratio between the empirical number of distinct detections (in non-target, motorbike images), $eNFA_D$, and the predicted NFA, which is ϵ

termines the optimal threshold that, when applied on the NFA values yields the best performance on the training set. The second algorithm does not calculate the NFA at all but rather determines the optimal threshold that, when applied directly on the appearance similarities, yields the best performance.

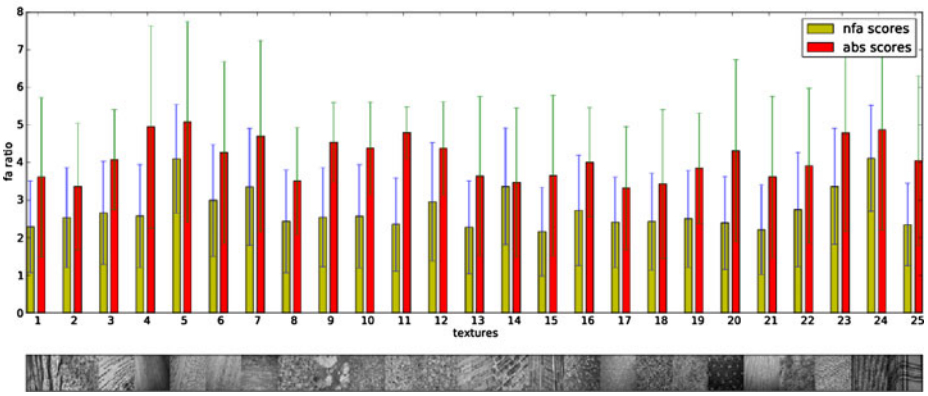
When trained on a data set similar to the test set, both thresholding methods are comparable and provide similar tradeoffs between misses and false detections; they give a similar equal error rate (EER). However, when the training set is not similar to the test set, decisions based on thresholding the NFA are more consistent and less sensitive to the difference between the training and test conditions.

To show this lower sensitivity, we applied the two detection algorithms to highly variable texture images. The texture collection from the UIUC dataset,[2] which contains 25 texture classes with 40 images in each class, was used. See Fig. 14 for one example from every class. One texture class was selected as a training set, and a distance from each image to the face model was calculated, A threshold leading to a 10 % false detection rate was specified. With this

[2] http://www-cvr.ai.uiuc.edu/ponce_grp/data/.



(a) Arithmetic mean of ratios. The x-axis shows the texture index. The y-axis shows the arithmetic mean values.



(b) Normalized arithmetic mean of ratios. The x-axis shows the texture index. The y-axis shows the normalized arithmetic mean values.

Fig. 13 Error bar graphs of ratio averages. Each bar represents the mean and the standard deviation of false detection ratios corresponding to the 24 test texture classes. The closer the bar is to 1, the more consistent with training data the decision criterion is. The *red bar* shows the ratio for the similarity-based decision (L_1 distance) and the *yellow bar* shows ratios for the NFA-based distance (Color figure online)

threshold, a face would be erroneously found in 4 out of the 40 images in this class. Then, using the same threshold, the false detection rate was measured for the remaining (24) classes. Ideally, the false detection rate should be 10 % in the other classes as well, and the ratio between the false alarm rate of some test class and the false alarm rate of the training class should be 1. To check the actual values, we measured the ratio statistics for the L_1 appearance distance based decision and for the NFA (MST version) based decision. This procedure was repeated for all classes as training sets.

Figure 13a shows the arithmetic mean of the calculated ratios. Note that for many textures serving as training data, the decisions based on appearance similarity (L_1 distances) provide high ratios—corresponding to high sensitivity to the choice of training data.

These raw ratios are not the best consistency indicators because ratios below and above 1 (ideal ratio) are averaged. Therefore we also provide a normalized view where, before averaging, each ratio r is modified to $\max(r, 1/r)$. This way ratios above and below 1 are treated identically. Figure 13b shows the arithmetic mean of the modified ratios for the two choices of decision function. It is clear, both from the means but also from the associated standard deviations, that the NFA based decisions are much more predictable and consistent.

This last experiment demonstrates the adaptivity of the NFA measure, which results from using the normalized distance (as described in Sect. 4.4.2). In many algorithms the decision is made by thresholding some distance and the threshold is determined by a training phase. The disadvantage of this approach is that when testing on non-target images that are significantly different from those in the training set, this threshold may not be optimal. An adaptive threshold, which changes depending on the image at hand, may provide better performance and predictability. Here we demonstrated that such an adaptive threshold, based on normalization, is feasible.

6 Conclusions

In this paper we proposed a generalization to the *a contrario* decision method. The generalization replaces the independence based background model with a more general model that is based on second order graphical models and is more suitable for correlated events, common in natural images.

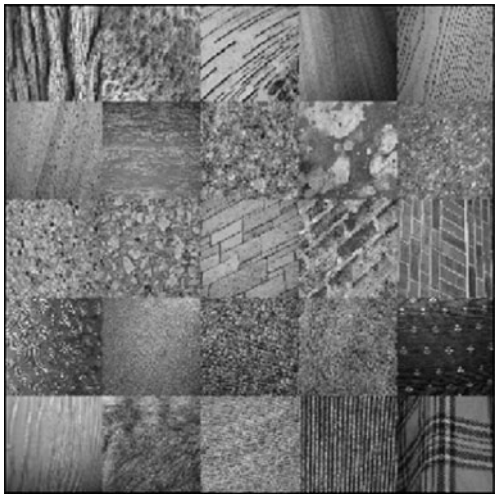


Fig. 14 Texture image example from the UIUC database

A computationally efficient decision procedure is proposed as well. We show the advantage of the modified model in the context of two applications: line segment detection and part-based object detection.

We believe that the proposed generalization may be applied to many other decision tasks. In this paper we used a relatively simple part based model and a simple category database. We believe that using of 2nd order (and higher order) statistics would also improve predictability in other detection algorithms and in more complex databases. Another possible application is SIFT matching (Rabin et al. 2009), where the matching of nearby sub-patches are likely to be dependent events, which could explain the observed gap between ϵ and the empirical false detection rate. In many applications of *a contrario*, a trade-off between independence and completeness of the description has to be found; see, e.g., the task of matching geometric structures (Muse et al. 2006). We expect that applying the proposed modeling to such matching tasks will alleviate this trade-off and possibly increase the discriminative power of the descriptor.

One question that remains open is the choice of probabilistic background model. Here we used a simple model, based only on second order properties, and which achieved better results than the independence based approach for two precise tasks. Other, more complex models of statistical dependence could be used. A model that perfectly characterizes the image data, is, however, undesirable as it would yield no detections in the modeled images. Characterizing the degree of best dependency is probably a difficult task.

A related question is whether and to what extent the proposed generalization is consistent with human judgement. For instance, do we see structure in colored noise? It seems that we do see it when the noise bandwidth is low, but not when the noise is close to white; see, e.g., Rajashekar et al. (2006). Finding the conditions for seeing structure may help to specify the best background model but may also help probe human perceptual organization.

References

Chow, C. K., & Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, *14*, 462–467.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms*. Cambridge: MIT Press.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. IEEE conf. comp. vision patt. recog.* (pp. 886–893).

Desolneux, A., Moisan, L., & Morel, J. M. (2000). Meaningful alignments. *International Journal of Computer Vision*, *40*(1), 7–23.

Desolneux, A., Moisan, L., & Morel, J. M. (2001). Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision*, *14*(3), 271–284.

Desolneux, A., Moisan, L., & Morel, J. M. (2008). *From gestalt theory to image analysis*. Berlin: Springer.

Fergus, R., Perona, P., & Zisserman, A. (2007). Weakly supervised scale-invariant learning of models for visual recognition. *International Journal of Computer Vision*, *71*(3), 273–303.

Flenner, A., & Hewer, G. A. (2011). Helmholtz principle approach to parameter free change detection and coherent motion using exchangeable random variables. *SIAM Journal on Imaging Sciences*, *4*(1), 243–276.

Galerne, B., Gousseau, Y., & Morel, J.-M. (2010). Random phase textures: theory and synthesis. *IEEE Transactions on Image Processing*, *20*(1), 257–267.

Gousseau, Y., & Roueff, F. (2007). Modeling occlusion and scaling in natural images. *Multiscale Modeling & Simulation. SIAM Interdisciplinary Journal*, *6*(1), 105–134.

Grimmett, G., & Stirzaker, D. (2001). *Probability and random processes* (3rd ed.). Cambridge: Oxford University Press.

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., & Randall, G. (2008). On straight line segment detection. *Journal of Mathematical Imaging and Vision*, *32*, 313–347.

Grompone von Gioi, R., Jakubowicz, J., Morel, J.-M., & Randall, G. (2010). Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 722–732.

Grosjean, B., & Moisan, L. (2009). A-contrario detectability of spots in textured backgrounds. *Journal of Mathematical Imaging and Vision*, *33*(3), 313–337.

Konishi, S., Yuille, A. L., Coughlan, J. M., & Zhu, S. C. (2003). Statistical edge detection: learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(1), 57–74.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Moisan, L., & Stival, B. (2004). A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, *57*(3), 201–218.

Muse, P., Sur, F., Cao, F., Gousseau, Y., & Morel, J. M. (2006). An a contrario decision method for shape element recognition. *International Journal of Computer Vision*, *69*(3), 295–315.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Mateo: Morgan Kaufman.

Rabin, J., Delon, J., & Gousseau, Y. (2009). A statistical approach to the matching of local features. *SIAM Journal on Imaging Sciences*, *2*(3), 931–958.

Rajashekar, U., Bovik, A. C., & Cormack, L. K. (2006). Visual search in noise: revealing the influence of structural cues by gaze-contingent classification image analysis. *Journal of Vision*, *6*(4), 379–386.

Ruderman, D. L. (1994). The statistics of natural images. *Network: Computation in Neural Systems*, *5*, 517–548.

Sabater, N., Morel, J. M., & Almansa, A. (2012). *Reliable matches in stereovision*. doi:10.1109/TPAMI.2011.207

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. (2005). Discovering objects and their location in images. In *Proc. int. conf. comp. vision*.

Van Trees, H. L. (1965). *Detection, estimation, and modulation theory, part I*. New York: Wiley.

Vidal-Naquet, M., & Ullman, S. (2003). Object recognition with informative features and linear classification. In *ICCV03* (pp. 281–288).

Weber, M., Welling, M., & Perona, P. (2000). Unsupervised learning of models for recognition. In *ECCV00* (pp. 18–32).

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: a comprehensive study. *International Journal of Computer Vision*, *73*(2), 213–238.