Joint *A Contrario* Ellipse and Line Detection

Viorica Pătrăucean, Pierre Gurdjos, and Rafael Grompone von Gioi

Abstract—We propose a line segment and elliptical arc detector that produces a reduced number of false detections on various types of images without any parameter tuning. For a given region of pixels in a grey-scale image, the detector decides whether a line segment or an elliptical arc is present (*model validation*). If both interpretations are possible for the same region, the detector chooses the one that best explains the data (*model selection*). We describe a statistical criterion based on the *a contrario* theory, which serves for both validation and model selection. The experimental results highlight the performance of the proposed approach compared to state-of-the-art detectors, when applied on synthetic and real images.

Index Terms—Ellipse detection, Line segment detection, *A contrario* theory, Model selection.

1 Introduction

The detection of line segments, circles, or ellipses has found applicability in very different computer vision setups: from traffic sign detection [48], tree counting in aerial images [36], face detection in natural images [46], sporting events analysis [50], to camera calibration using geometric patterns [22], or autonomous driving in unknown environments [47], to cite only a few. Each of these applications has specific image data in terms of shape variability and noise, and specific expected output in terms of accuracy and computational complexity. Hence, designing a reliable fully automatic general-purpose detector for a certain geometric primitive is a challenging task. One of the crucial aspects under such varying conditions is the tuning of the detection thresholds, which has direct impact on the number of false detections. Too permissive parameter values will report *false positives* on noisy regions or on other types of primitives, whilst too restrictive values will produce *false negatives* on valid partially-occluded or noisy primitives.

Our aim is to propose an automatic detection method allowing to reliably detect line segments and elliptical arcs in images, regardless of their size, content, or source. Two aspects are critical in primitive detection regarding reliability: the control of false positives and the correct geometric interpretation of a given image region when multiple interpretations are possible, e.g. a polygonal line and an elliptical arc. As these aspects are usually addressed by adequate tuning of detection thresholds, our key concern is to propose an automatic computation of these thresholds.

Preliminary results of the proposed detection approach were published in [41], where we introduced ELSD (Ellipse and Line Segment Detector), as a three-stage detection procedure: *(i) candidate generation*, *(ii) candidate validation*, and *(iii) model selection*. Here, we describe ELSDc (Ellipse and Line Segment Detector, with Continuous formulation), which follows the same three-step scheme as ELSD. Both algorithms use *a contrario* criteria [11] for candidate validation and model selection. However, ELSDc uses a *continuous* formulation instead of the discrete one in ELSD. As we will show, this new criterion allows to handle different noise levels in a natural way, resulting in a detector more robust to noise. Moreover, in [41], line segments and elliptical arcs were handled and compared directly. Here, the line segments are grouped by proximity into convex and smooth polygonal lines. Then, the elliptical arcs are compared to the whole polygonal line instead of isolated line segments, giving a more fair comparison and improving significantly the model selection step. Eventually, polygonal lines are decomposed into their forming line segments for the output.

By means of extensive experiments on synthetic and natural images, we demonstrate the robustness against noise and the precision of the proposed detection approach on incomplete (occluded) contours. The source code and a demo where users can upload images and test the detector are available at github.com/viorik/ELSDc.

In the following, we start by discussing related works and the geometric primitives studied in this paper. Section 2 describes the heuristic used to generate polygonal and elliptical candidates. Sections 3 and 4 detail the candidate validation and model selection steps, introducing a continuous criterion for validation and model selection. Implementation details of the overall detection pipeline are given in section 5, together with a pseudo-code. The advantages of using the new criterion for validation and model selection tasks are empirically highlighted in section 6. Section 7 reflects experimentally the efficiency and the robustness of the overall approach in relation to state-of-the-art geometric primitive detectors. Section 8 concludes the paper.

1.1 Related Work

Most of the existing geometric primitive detectors address a single primitive type, e.g. line segments or ellipses, and use either different versions of the *Hough Transform* (HT) [24] or *edge chaining* methods. Regardless of the method, the detectors usually take as input an edge map produced by an edge detector [7] and not the original grey-scale image.

The Hough-based algorithms [1] start by sampling primitive candidates from the primitive parameter space in

- *V. Pătrăucean is with the Department of Engineering, University of Cambridge, UK. E-mail: vp344@cam.ac.uk*
- *P. Gurdjos is with IRIT-ENSEEIHT, France*
- *R. Grompone von Gioi is with CMLA-ENS Cachan, France*

Manuscript received ; revised

an exhaustive manner or using some selection schemes [30]. Subsequently, the edge points cast votes to candidates according to a predefined fitness criterion and the vote accumulations that exceed a detection threshold become detections. The critical parameters involved are the detection thresholds and the quantisation precision when sampling the parameter space. Their values require careful (manual) tuning each time the image size, type, or content changes, making them less suitable for automatic applications. Another drawback of this class of detectors resides in the resource requirements (execution time, memory consumption). Randomised approaches reduce the execution time by considering only a fraction of the edge points in the voting procedure [17], [52], whereas statistical formulations address the parameter tuning aspect. In [3], [26], the authors model the *uncertainty* associated to edge pixels induced by the image noise or by the edge detection procedure. Subsequently, the votes casted by the edge points are weighted accordingly using a Bayesian formulation. Within the same Bayesian framework, [9] uses kernel-based modelling to bypass the tuning of the quantisation precision. Similarly, Barinova et al. [2] propose a Bayes-based statistical framework to obtain a parameterless non-maxima suppression using Maximum-a-Posteriori inference. However, only line detection implementations were presented for these Bayesian approaches, and even in this case their computational complexity is very high, reducing their applicability.

The detection methods relying on edge chaining techniques use extensively the geometric properties of the sought primitives, such as straightness criteria for line segments or curvature properties for ellipses [8], [32], [37]. These algorithms chain connected edge points into lists and fit the sought primitives on each edge list using deterministic fitting techniques (e.g. based on least-squares [14]), or randomised RANSAC-like approaches [32], [37]. Although efficient in execution time (as opposed to Hough-based methods), these algorithms [8] have difficulties in handling noisy edge maps, reporting an important number of false detections caused by inappropriate detection thresholds. The ellipse detector proposed by Prasad et al. [37] uses smoothness and convexity constraints to find ellipse candidates and performs an additional merging step to group incomplete ellipses that can occur due to noise or occlusions. Eventually, saliency scores are used to validate the ellipse candidates. The grouping step proves to be non-trivial in practice and the saliency criteria lack self-tuning capabilities, generating false detections. These are illustrated in our experimental study (section 7).

A few works have tried to automate the tuning of the detection thresholds. The Progressive Probabilistic Hough Transform (PPHT) [15] stops the voting procedure for a particular feature when an excess is observed in the accumulator that could not have appeared by accident. They assess the accidentalness by using as detection threshold a cut-off value on the probability that the observed excess occurred by chance in a noise image. This reasoning allows the rejection of false positives. Nonetheless, PPHT lacks scalability: the cut-off values are set for a predefined image size, and the guard against false positives is not ensured when larger images are analysed. The ellipse detector proposed by Chia et al. [8] tackles this problem by learning detection threshold values on computer-generated images, but false detections are still present when testing on real images. The learnt thresholds are not able to reject small ellipses reported on parasite contours, while falsely rejecting some valid incomplete ellipses. The formal control of the number of false detections within the primitive detection problem was addressed by some authors. Stewart proposed MINPRAN for 3D alignment detection [49], but it still needs a parameter tuning.

For our work we adopt the approach introduced by Desolneux et al. [10], [11], known as the *a contrario* approach. It provides an efficient technique for the automatic computation of the detection thresholds based on the so-called *non-accidentalness principle*, or Helmholtz's perception principle, which informally states that there is no perception in random images [10]. This reasoning originates partially in Lowe's seminal work on parameter-free significance assessment, where the significance of an event is quantified based on the probability that the event occurred by chance: *"we need to determine the probability that each relation in the image could have arisen by accident. Naturally, the smaller that this value is, the more likely the relation is to have a causal interpretation."* (D. Lowe [31, p.39]). Albeit parameter-free, Lowe's criterion does not assess the *overall* non-accidentalness, which causes an incomplete guard against false detections. This is exactly the main contribution of the *a contrario* statistical framework: by focusing on the *expected* number of events occurred by chance, instead of on the probability as done by Lowe, a control on the overall number of false detections is achieved. Specifically, the *a contrario* framework associates to each candidate a Number of False Alarms (NFA), representing the expected number of candidates, at least as structured as the analysed one, that could have appeared by chance in a noise image of the same size. As in Lowe's criterion, the smaller the NFA, the more structured the primitive is. This quantity is used as validation criterion: when a primitive's NFA is below a given value, the primitive is declared valid.

This statistical formulation of non-accidentalness comes to the Multiple Hypothesis Testing (MHT) framework [29], widely used in statistics to control the number of false positives. Numerous applications of the MHT framework guard against false positives using p-values. More recently, for problems where the number of tests is large, E-values proved to be more adapted[1] [16]. The NFA score described above corresponds precisely to the E-value. Indeed, in geometric feature detection problems, the number of tests is large and using p-values does not ensure an effective control of false detections. We refer the reader to [40, section 4.4] for a more detailed discussion on the analogy between the *a contrario* methodology and MHT. The *a contrario* reasoning was used by several authors to address various computer vision tasks: automatic line segment detection [19], [21], feature point matching [42], image registration [34], segmentation of satellite imagery [44], symmetry detection [39].

Most of the works mentioned above address a single primitive type, e.g. lines, or circles, or ellipses, and try to limit the number of false detections produced by noise and

1. In statistics, the p-value and E-value are defined as the probability and expectation, respectively, of observing under the null hypothesis a result at least as extreme as the actual observation, in the sense of the test to be performed [29].

occlusions. Another important source of false detections comes from the misidentification of the primitive type, e.g. an ellipse reported on a convex polygon. To avoid this, a detector should be enhanced with model selection capabilities[2]. An important number of model selection criteria are available in the statistical literature [12]. We refer the reader to [45] for a general classification. In this respect, our work relates to existing techniques for curve segmentation like Etemadi's method for curve segmentation into line segments and circular arcs [13] or West et al.'s work, which segments curves into line segments and elliptical arcs [51]. These methods assume a noiseless edge map as input and report numerous false positives on noisy edge maps, as it will be shown in section 7. These methods use the Maximum Likelihood (ML) criterion to perform model selection: the model which approximates the data most faithfully is kept. However, we suggest that this is in contradiction with the model selection theory, which states that one must penalise the complexity of the model in order to prevent overfitting. The same idea is shared by Kanatani [28], who uses an adapted version of Akaike's Information Criterion (AIC). Isack and Boykov [25] proposed an elegant solution by formulating the detection problem as an optimal labelling problem, which alternates between a discrete optimisation step that assigns points to geometric models, and a continuous optimisation step that tries to find the models that best fit the data in a least-squares sense. The energy optimisation problem incorporates a Minimum Description Length (MDL) term, acting as a model selection criterion.

1.2 Geometric Primitives

The geometric primitives considered throughout this paper are defined as (see Fig. 1 middle and right):

- *d-side polygonal line* g: a sequence of $d+1$ rectangle end-points and a common width;
- *elliptical arc* a: an elliptical ring given by centre, axes, orientation, delimiting angles, and width;
- *circular arc* $\overline{a}$: an elliptical ring with equal axes and orientation equal to 0.

We consider grey-scale images of size $m \times n$, defined on a Cartesian grid $\Gamma = [1, m] \times [1, n] \subset \mathbb{N}^2$, with values in $\mathbb{R}$. The image pixels falling within the boundaries of a primitive constitute its *support region* (Fig. 2 far right), and will be used to validate the primitive. The primitive width modulates the thickness of the support region, allowing accurate detections on both blurred or sharp image regions.

2 Candidate Generation

The candidate generation step proposes candidates for polygonal and elliptical primitives. Its target is to obtain a high recall, i.e. avoid false negatives, while ignoring the precision rate, i.e., dealing with false positives. Improving the precision is the target of the *a contrario* validation. Any of the existing detectors [8], [13], [37] could be used to generate candidates. We choose to use a greedy heuristic

2. A similar, more general idea is present in [11, p. 231]: *"The outcome of a partial gestalt detector is valid only when all other partial gestalts have been tested and the eventual conflicts dealt with."*

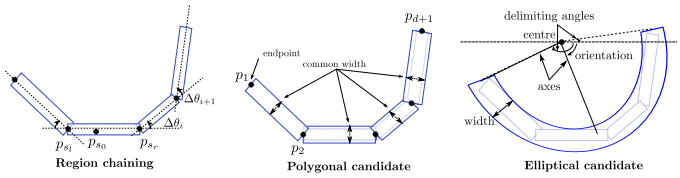


Fig. 1. Left: Region chaining. Middle and right: Parameters defining the polygonal and the elliptical candidates.

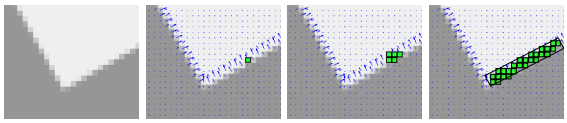


Fig. 2. From left to right: original image, gradient orientations and seed pixel, region growing starting from the seed pixel. Far right: the final support region and the corresponding candidate rectangle.

extending the one proposed in Line Segment Detector (LSD) [19], which operates directly on the image intensity. Namely, the proposed algorithm alternates *region growing* and *region chaining* operations to obtain polygonal approximations of the image contours.

The region growing step [21] starts from a seed pixel p_{s_0} and then recursively groups connected pixels that share the same gradient orientation up to a given precision (see Fig. 2). When no more pixels can be added, a rectangle is fitted on the gathered pixels. The end-points of the obtained rectangle are used as new seed pixels, p_{s_r} and p_{s_l}, for subsequent region growing procedures in each direction (Fig. 1 left). The region chaining step merges the rectangles found that have common end-points, provided they obey some loose, elementary constraints that characterise the elliptical primitives (Fig. 1 left):

- *convexity*: consecutive rectangles turn in the same direction, thus $\Delta\theta_i$ and $\Delta\theta_{i+1}$ have the same sign;
- *smoothness*: the orientations of consecutive rectangles differ by less than $\pi/2$.

When no more rectangular regions can be added in the sequence, the polygonal and the elliptical candidates are estimated (see Fig. 1, middle and right). The polygonal candidate g is derived directly from the rectangle sequence, by imposing the maximum rectangle width as common for the whole polygonal line. For the elliptical candidate a, we compute the first five parameters of the ellipse that fits the gathered pixels using the conic fitting technique introduced in [41]. For the remaining three parameters, the delimiting angles are given by the end-points of the last added rectangles, whilst the width is computed by summing the distances[3] of the points that are the farthest from the ellipse towards the exterior and the interior respectively. Being a particular case of ellipse but with less parameters, a circular candidate $\overline{a}$ is fitted as well, to counteract the poor accuracy of ellipse fitting techniques when input data are sampled only along small arcs. In these cases, the curve,

3. Computing the exact Euclidean distance between a point and an ellipse requires solving a quartic equation. For efficiency reasons, we use Rosin's approximation [43], detailed in the supplemental material.

even if part of an ellipse, can be fairly approximated by a circular arc [43].

Starting from the seed pixel, the candidate generation step outputs three primitive candidates: g (polygonal), a (elliptical), and $\overline{a}$ (circular). If one of them is validated (as described in sections 3 and 4), the pixels in the corresponding support region are marked and not used again. Then, a new seed pixel is selected and the process is repeated until all pixels are marked. More implementation details are given in section 5.

3 *A Contrario* Validation

The main idea in the *a contrario* validation is to automatically compute the detection thresholds in a way that rejects candidates whose presence might be accidental [11]. Two main ingredients are required within the *a contrario* framework: *(i)* a measure function giving a score to each candidate, reflecting its *degree of structuredness* and *(ii)* a stochastical model of unstructured image data. A candidate is validated when it is too structured, in the sense of the measure of *(i)*, to be expected in the model of *(ii)*. This section defines precisely this *a contrario* setup for our problem. We refer the reader to [11] for more details.

We describe in the following the unstructured model used in our work, proposed by Desolneux et al. for the line detection problem [11]. Then, after a brief reminder of the discrete measure function for primitive structuredness used in [41], we introduce a new continuous measure function and we give the full statistical setup for validation. The theoretical reasoning is completed by the proof of the control of false positives within this setup.

Desolneux et al. use the gradient orientation to assess the degree of structuredness of a line segment [10]. With this choice, a convenient unstructured model, that we denote as the *a contrario* $\mathcal{H}_0$ model, is:

1) $\forall p \in \Gamma, \text{angle}(\nabla x(p))$ is uniformly distributed over $[0, 2\pi)$,
2) $\{\text{angle}(\nabla x(p))\}_{p\in\Gamma}$ are independent random variables,

where $\nabla x(p)$ is the gradient vector of the image x at point p and $\text{angle}(\nabla x(p))$ defines its orientation. It has been proven by Desolneux et al. that these assertions hold under certain subsampling conditions if x is a Gaussian white noise image of any power [11, p. 67]. Henceforth, x will refer to the analysed image and X to a random image of the same size as x, drawn from the $\mathcal{H}_0$ model[4].

The above *a contrario* model can be seamlessly applied for polygonal and elliptical primitives detection as well. Note that $\mathcal{H}_0$ is not meant to model the real noise present in the analysed image. Instead, since it is a simple model for equiprobable orientations, it can represent well unstructured, isotropic regions of real images, where no aligned structures are perceived. This model was thoroughly evaluated in [11] and a convincing agreement was obtained between results of line segment detectors grounded on this model and human perception. Our experimental results (section 7) also support this statement.

4. Note that, strictly speaking, $\mathcal{H}_0$ is a model for random gradient orientation fields, not for random images. We use this slight abuse of notation to simplify the writing, with no risk of confusion since it is always the gradient orientation field of X that is used throughout the entire paper.

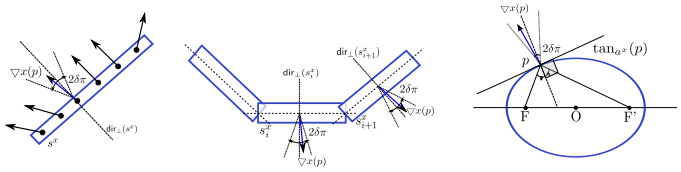


Fig. 3. Left: Three pixels δ-aligned with an oriented line segment. Centre: Pixels aligned with a polygon. Right: Aligned pixel in ellipse case.

We use the notation c^x to refer to a candidate primitive c observed in image x. We use different letters to indicate the primitive type: s for line segment, g for polygonal line, a for elliptical arc, and $\overline{a}$ for circular arc.

To assess the degree of structuredness of a primitive, Desolneux et al. use the concept of *aligned pixel*. Intuitively, it refers to pixels contained in the support region of the primitive and where the image gradient is orthogonal to the primitive (see Fig. 3, left). For an oriented line segment s^x, a pixel $p \in \Gamma$ is said to be *aligned* up to a precision $\delta \in [0, 1]$ if

$$\text{angle\_err}(\nabla x(p), \text{dir}_\perp(s^x)) \leq \delta\pi, \tag{1}$$

where $\text{dir}_\perp(s^x)$ is the direction orthogonal to the line segment s^x and $\text{angle\_err}(a, b)$ is the non-oriented angle between vectors a and b, giving a value in $[0, \pi]$.

For a polygonal line $g^x = \cup_i s_i^x$ (see Fig. 3, centre), a pixel p in its support region is said to be aligned when it is aligned with the line segment s_i^x containing the pixel. For an elliptical arc a^x (see Fig. 3, right), a pixel p in its support region is said to be aligned up to a precision δ if

$$\text{angle\_err}(\nabla x(p), \text{dir}_\perp(\tan_{a^x}(p)) \leq \delta\pi, \tag{2}$$

where $\text{dir}_\perp(\tan_{a^x}(p))$ is the direction orthogonal to the tangent $\tan_{a^x}(p)$ to the ellipse at its closest point to p. Circles are particular cases of ellipses and need no dedicated definition.

Let $l(c^x)$ be the number of pixels in the support region of the candidate c^x. We define binary variables $y_1, y_2, \ldots, y_{l(c^x)}$ associated with each pixel in the support region and taking the value 0 when the pixel is aligned up to a given precision δ, and 1 otherwise[5]. Finally, let $k(c^x)$ be the measure function used to assess the degree of structuredness of a candidate c observed in an image x.

For the line segment detection problem, the measure function $k(s^x)$ used in [11], [19], [41] is $k(s^x) = \sum_{j=1}^{l(s^x)} y_j$. We use the same additive measure function for polygonal lines and elliptical arcs, with the corresponding notions of aligned pixels. This measure has a *discrete* nature, taking integer values between 0 and $l(c^x)$: 0 for a perfectly structured primitive and $l(c^x)$ for an unstructured one.

To bypass the precision parameter δ involved in this discrete formulation, Grompone von Gioi et al. proposed

5. It is common in the *a contrario* literature to use these values interchanged, i.e. 1 corresponds to an aligned pixel and 0 to a non-aligned pixel. We switched the values here to have a homogeneous writing with the continuous formulation, described later in this article. Note, however, that this is only a change of notation, but there is no change at all in the formulation.

a continuous measure function for the line segment detection problem [18], [20], following the idea of Igual et al. [44]. With this formulation, a primitive candidate c containing $l(c^x)$ pixels is not any more evaluated through binary variables y_j. Instead, the angular errors introduced in equations (1) and (2) are normalised and kept as is: $\tilde{y_j} = \frac{\text{angle\_err}(\nabla x(p_j), \text{dir}_\perp(c^x))}{\pi} \in [0,1]$.

To evaluate the degree of structuredness of a candidate c^x, a new measure function $\tilde{k}(c^x)$ needs to be defined. Analogically to the discrete case, $\tilde{k}(c^x)$ is defined as the sum of the normalised per-pixel errors:

$$\tilde{k}(c^x) = \sum_{j=1}^{l(c^x)} \tilde{y_j} = \sum_{j=1}^{l(c^x)} \frac{\text{angle\_err}(\nabla x(p_j), \text{dir}_\perp(c^x))}{\pi}. \quad (3)$$

The smaller the $\tilde{k}(c^x)$, the more meaningful the candidate is. Since the normalised angular errors are real numbers in $[0,1]$, $\tilde{k}(c^x)$ can take any real value in $[0, l(c^x)]$ and not only integers as in the discrete formulation. As before, $\tilde{k}(c^x) = 0$ corresponds to a perfectly structured primitive, whereas $\tilde{k}(c^x) = l(c^x)$ corresponds to the worst case.

To validate *a contrario* a given candidate c^x in the analysed image x, we need to compute the probability of observing in the random image X candidates c^X at least as structured as c^x. This comes down to computing $\mathbb{P}[k(c^X) \leq k(c^x)]$ and $\mathbb{P}[\tilde{k}(c^X) \leq \tilde{k}(c^x)]$.

In the discrete case, the probability for a pixel to be aligned up to a given precision δ under the *a contrario* model $\mathcal{H}_0$ is $\frac{2\pi\delta}{2\pi} = \delta$. Since the gradient orientations are independent under $\mathcal{H}_0$, the score $k(c^X)$ corresponds to a sum of Bernoulli variables and thus follows a binomial law with parameters $l(c^x)$, $k(c^x)$, and δ. Therefore, we can write:

$$\mathbb{P}[k(c^X) \leq k(c^x)] = \sum_{j=0}^{k(c^x)} \binom{l(c^x)}{j} \delta^{l(c^x)-j}(1-\delta)^j. \quad (4)$$

More details about the discrete formulation are given in [11].

For the continuous case, under the *a contrario* model $\mathcal{H}_0$, the normalised per-pixel errors $\tilde{y_j}$ are real random variables, uniformly distributed in $[0,1]$. Consequently, $\tilde{k}(c^X)$ is the sum of $l(c^X)$ i.i.d. uniform random variables. But the sum of i.i.d. random variables uniformly distributed in $U(0,1)$ follows the Irwin-Hall distribution [27], which has a closed-form expression. Let Z be an Irwin-Hall random variable of parameter r, i.e. the sum of r i.i.d. $U(0,1)$. Then its cumulative distribution function is:

$$\mathbb{P}[Z \leq z] = \frac{1}{r!} \sum_{j=0}^{\lfloor z \rfloor} (-1)^j \binom{r}{j} (z-j)^r, \quad (5)$$

where $\lfloor z \rfloor$ is the largest integer not bigger than z. (An alternative derivation of this expression is detailed in the supplemental material.) In our case, $\tilde{k}(c^X)$ corresponds to Z, the observed value $\tilde{k}(c^x)$ corresponds to z, and the number of variables r corresponds to $l(c^x)$. Then,

$$\mathbb{P}[\tilde{k}(c^X) \leq \tilde{k}(c^x)] = \frac{1}{l(c^x)!} \sum_{j=0}^{\lfloor \tilde{k}(c^x) \rfloor} (-1)^j \binom{l(c^x)}{j} (\tilde{k}(c^x) - j)^{l(c^x)}. \quad (6)$$

The following definition and proposition complete the *a contrario* validation setup. We detail the reasoning for the continuous formulation. For the discrete case, the derivation is straightforward.

Definition 1. *Let $\mathcal{Z} = \{Z_1, \ldots, Z_{N_\mathcal{Z}}\}$ be a set of $N_\mathcal{Z}$ positive real random variables. Observing a value z_i for Z_i is an ε-meaningful event in $\mathcal{Z}$ if its associated* Number of False Alarms, *defined by*

$$\text{NFA}(z_i) = N_\mathcal{Z} \mathbb{P}[Z_i \leq z_i] \quad (7)$$

is less than or equal to ε.

Proposition 1. *The expected number of ε-meaningful events in $\mathcal{Z}$ is less than or equal to ε.*

Proof. *We define the set of thresholds $\tau_i = \max\{z \in \mathbb{R} \mid N_\mathcal{Z} \mathbb{P}[Z_i \leq z] \leq \varepsilon\}$. With this definition, an observed value z_i is ε-meaningful if and only if $z_i \leq \tau_i$. Then, the subset of elementary ε-meaningful events associated to Z_i is $\{Z_i \leq \tau_i\}$, and the expected overall number of ε-meaningful events in $\mathcal{Z}$ is given by*

$$\mathbb{E}\left[\sum_{i=1}^{N_\mathcal{Z}} \mathbb{1}_{\{Z_i \leq \tau_i\}}\right] = \sum_{i=1}^{N_\mathcal{Z}} \mathbb{P}[Z_i \leq \tau_i] \leq \sum_{i=1}^{N_\mathcal{Z}} \frac{\varepsilon}{N_\mathcal{Z}} = \varepsilon. \ \blacksquare$$

In our case, the set $\mathcal{Z}$ corresponds to the set of scores $\left\{\tilde{k}(c_1^X), \ldots, \tilde{k}(c_{N_c}^X)\right\}$ associated to the family of possible candidates. We denote by N_c the cardinal of this family. In agreement with Definition 1, the Number of False Alarms associated to the observation $\tilde{k}(c_i^x)$ is $\text{NFA}\left(\tilde{k}(c_i^x)\right) = N_c \mathbb{P}\left[\tilde{k}(c_i^X) \leq \tilde{k}(c_i^x)\right]$. With no risk of confusion, we simplify the notation to $\text{NFAc}(c_i^x)$ for the continuous formulation and to $\text{NFAd}(c_i^x)$ for the discrete case. In agreement with Definition 1, a candidate c_i^x occurring in image x will be considered valid and said to be ε-meaningful when

$$\text{NFAc}(c_i^x) \leq \varepsilon\ . \quad (8)$$

The Number of False Alarms (continuous or discrete) reflects the number of candidates at least as structured as the analysed one that could be expected in a random image X of the same size as x. The smaller the NFA value, the more unlikely is c_i^x to appear in an image X drawn from $\mathcal{H}_0$; thus, the more meaningful. Thanks to Proposition 1, if a candidate is accepted as valid detection when the equation (8) holds, then the number of accidental (false positive) detections is guaranteed to be less than the chosen ε for a random image drawn from $\mathcal{H}_0$.

In practice, ε can be set as small as desired. As proven in [11, p. 77], the detection results have a (weak) logarithmic dependence on ε. The value $\varepsilon = 1$ yielded satisfactory results [19] and we fix it once and for all for the proposed detector as well.

Finally, we need to specify the number of candidates N_c. In practice, only the candidates found by the first step of the algorithm are tested. However, this does not imply that one can set N_c equal to the number of candidates effectively tested. The support regions of the actual candidates have complex statistics, very different from $\mathcal{H}_0$. Indeed, each one of them is the result of a heuristic procedure that groups pixels with similar gradient orientations; even if the input image follows $\mathcal{H}_0$, each support region resulting from the grouping process would not. In the absence of a simple formula for this complex statistics that would allow a control

$$\text{For a polygonal line } g^x, \qquad \mathrm{NFAd}(g^x) = 2^{d+1}(mn)^{(d+1)+\frac{1}{2}} \sum_{j=0}^{k(g^x)} \binom{l(g^x)}{j} \delta^{l(g^x)-j}(1-\delta)^j \tag{7a}$$

$$\text{For an elliptical arc } a^x, \qquad \mathrm{NFAd}(a^x) = 18(mn)^4 \sum_{j=0}^{k(a^x)} \binom{l(a^x)}{j} \delta^{l(a^x)-j}(1-\delta)^j \tag{7b}$$

$$\text{For a polygonal line } g^x, \qquad \mathrm{NFAc}(g^x) = 2^{d+1}(mn)^{(d+1)+\frac{1}{2}} \frac{1}{l(g^x)!} \sum_{j=0}^{\lfloor \tilde{k}(g^x) \rfloor} (-1)^j \binom{l(g^x)}{j} (\tilde{k}(g^x) - j)^{l(g^x)} \tag{7c}$$

$$\text{For an elliptical arc } a^x, \qquad \mathrm{NFAc}(a^x) = 18(mn)^4 \frac{1}{l(a^x)!} \sum_{j=0}^{\lfloor \tilde{k}(a^x) \rfloor} (-1)^j \binom{l(a^x)}{j} (\tilde{k}(a^x) - j)^{l(a^x)} \tag{7d}$$

of the number of false detections, we must be contented with counting the number of polygonal and elliptic arcs in a family rich enough to include all the candidates evaluated in practice.

The candidates are validated by measuring the gradient orientations within the corresponding support regions. What counts then are the support regions, which are sets of pixels defined up to pixel precision. Since the circular candidates are particular elliptical candidates, we include both primitives in the same elliptical family. Let us consider each candidate type. A polygonal candidate with d sides is defined by $d+1$ points and a width (see Fig. 1 middle). In an $m \times n$ image, there are roughly $(mn)^{d+1}$ possible values for the point coordinates and $(mn)^{1/2}$ possible values for the width. Hence, the number of potential d-side polygonal candidates can be approximated by $N_g = (mn)^{(d+1)+\frac{1}{2}}$. An alternative and simpler derivation is possible by counting the degrees of freedom: a d-side polygonal candidate has $2(d+1)+1$ degrees of freedom and we count $(mn)^{1/2}$ possible values for each degree of freedom. With this method, elliptical arc candidates have eight degrees of freedom: centre (2), axes (2), orientation (1), width (1), and delimiting angles (2) (see Fig. 1 right). However, since we are interested in elliptical arcs, we need to consider that the centre of the ellipse can be located outside the image as well. Experiments show that it is sufficient to consider the arcs belonging to ellipses whose centres are located within the image and within the set of 8-connected rectangles of size $m \times n$, sharing a vertex or and edge with the rectangle surrounding to the image. So the contribution of the centre needs to be multiplied by 9. Thus, the family of elliptical arcs has an approximate size of $N_a = 9(mn)^4$.

To account for the multiple families of primitives while still controlling the overall number of false positives at the ε level, we divide the accepted quantity of false positives among families. Since we have no prior on the frequency of occurrence of the different types of primitives, we choose to divide this quantity equally among families; hence the elliptical family can contribute with $\frac{\varepsilon}{2}$ false positives, and each d-polygonal family contributes with $\frac{\varepsilon}{2^{d+1}}$. The latter term guarantees a total contribution from the polygonal family of $\frac{\varepsilon}{2}$, while considering any possible value for d. Summing these contributions, the accepted number of false positives equals ε. To simplify the writing, the factor $\frac{1}{2}$ for ellipses and $\frac{1}{2^{d+1}}$ for polygons is included as a multiplicative factor in the NFA expression.

Putting together the multiplicative factors mentioned above, the number of candidates for each case, N_g or N_a, and the probability term for the discrete, eq. (4) or continuous case, eq. (6) into eq. (7), we obtain the final NFA expression in eq. (7a) to (7d).

4 Model Selection

When more than one candidate is declared meaningful for the same support region during the validation step, a subsequent model selection step is needed. We choose to use the (continuous or discrete) NFA of the primitives as model selection criterion: the model possessing the smallest NFA is considered as most meaningful, and kept as valid detection. This was suggested by Desolneux et al. [11, p. 245] and introduced in our previous work for the discrete case [41].

Note that ELSDc entails a twofold usage of NFA as model selection criterion. First, it decides between primitives belonging to the same family: the circular and the elliptical models, and the priority is given to the one having the smallest probability of being generated by noise. This usage is recurrent in the *a contrario* theory. For example in [11], NFA is used to choose the best line segment explaining some image data among multiple line segment interpretations, or in [33], [34], NFA serves to decide between multiple possible models of fundamental matrices connecting two images.

The second usage is less straightforward. We need to decide between models belonging to different families: polygonal and elliptical models. In this case, a trade-off is required between the goodness of fit and the complexity of the model.

To illustrate the pertinence of using NFA as model selection criterion, we briefly compare NFA with AIC, which is one of the most representative model selection criteria, and an adapted version of AIC was proposed in [28] within the geometric model selection problem. Specifically, AIC offers a *relative* measure of the information loss when a given model is used to describe some real data. AIC is very simple, using a penalty term which increases with the number of parameters. For an input sample containing l points and an estimated model $\hat{f}_j$ having t_j parameters, AIC can be written as $\mathrm{AIC}_j = -\frac{2}{l}\log \mathsf{L}_j + 2\frac{t_j}{l}$, where L_j

is the maximised value of the likelihood function for the estimated model $\hat{f}_j$ [45].

The two terms multiplied in the NFA expression (7) are similar to the AIC terms: the probability term reflects the goodness of fit, whereas the number of candidates, dependent on the number of parameters of the model, penalises the complexity of the model. However, note that for a given dataset and some candidate models, AIC allows to rank the models according to their suitability to explain the data. AIC is, par excellence, a tool for model selection, but it can not measure how well a model fits the data in an *absolute* sense: if all the candidate models fit poorly, the AIC values will not reflect that. In contrast, NFA gives an absolute measure of structuredness, making it a powerful tool for validation and model selection purposes.

5 Implementation Details

In this section we detail some implementation choices, and outline the main detection steps in Algorithm 1. The region growing procedure (Algorithm 1, line 8) described in section 2, groups pixels sharing the same gradient orientation up to a chosen precision. We use the precision value $\delta = \frac{1}{8}$, corresponding to a $22.5°$ angle, which was used also in [5], [11], [19], giving good results in practice. The support regions reported by the region growing step are then merged by the region chaining step (Algorithm 1, line 9), detailed in section 2. More details about the region growing heuristic can be found in [21] and [40].

The noise inherently present in images can affect the candidate generation by breaking the continuity or/and the convexity of the polygonal line. To limit this effect, we smooth the original image using a Gaussian filtering with variance $\sigma = 1$ (Algorithm 1, line 2). The validation, however, is carried out on the original image (Algorithm 1, line 17), in order to respect the independence assumption of the *a contrario* model described in section 3.

Another way of limiting the noise influence during both candidate generation and validation, is to consider only the pixels whose gradient is strong enough, as in this case the noise is not dominant and the pixel orientation can be considered reliable. We use the same value of $\rho = 5$ as in [11], [19]. Limiting the contribution of pixels with low gradient magnitude is also used in statistical Hough methods [9], because the measurements associated to these pixels, mainly the gradient orientations, have high uncertainty and are unsuitable for statistical inference.

To improve the accuracy of the detector, the primitive width estimated during candidate generation is refined during validation. Namely, for each candidate, the primitive width is reduced successively by 0.5 pixels, and the width reporting the best validation score is kept as primitive width (Algorithm 1, lines 13–15). However, this refinement has no influence over the validation step; instead it is a simple way to generate extra candidates. Note that there is no need to modify the estimates of the number of possible candidates introduced in the previous section, which are already overestimations. Indeed, we counted $\sqrt{mn}$ possible values for the primitive width. But the primitives' widths are far less than $\sqrt{mn}/2$, so the half pixel refining steps are largely counted.

From a numerical point of view, computing $\mathbb{P}[Z \leq z]$ in equation (5), where $z \in \mathbb{R}_+$ and Z is the sum of r i.i.d. random variables, may lead to numerical issues as it is often the case when computing sums of values with large ranges [38]. A similar problem appears when computing the NFA in the discrete case (equation 4), which is often bounded using Hoeffding's inequality [11, chap. 4]. In our case, it can be shown that the first term of the sum involved in equation (5) gives an upper-bound of this probability: $\mathbb{P}[Z \leq z] \leq \frac{z^r}{r!}$. The latter computation is trivial using Stirling's formula [6]. Using this bound instead of the exact probability value guarantees that a validated candidate is indeed an ε-meaningful event. But it is possible that some ε-meaningful events are not validated due to an overvalued NFA. In practice, however, true detections have strong structuredness scores and using this upper-bound does not influence the final result in a relevant way[6]. This justifies the following approximation of the probability:

$$\mathbb{P}[\tilde{k}(c^X) \leq \tilde{k}(c^x)] \approx \frac{[\tilde{k}(c^x)]^{l(c)}}{l(c)!}. \tag{10}$$

Note that the values of the parameters present in the algorithm (ε, δ, ρ, and σ) have been used in several related works [5], [11], [19], [41] and have consistently shown good results in practice on numerous and various types of images. Hence, they are considered as constants in our work, making our detector fully automatic.

Algorithm 1 Main steps of the ELSDc detection pipeline.

Input: x – image to be analysed
Output: $\mathcal{E}$ – set of detected elliptical arcs, $\mathcal{S}$ – set of detected line segments

1: $\delta \leftarrow \frac{1}{8}$, $\rho \leftarrow 5$, $\sigma \leftarrow 1$
2: $x_\sigma \leftarrow$ Gaussian_filtering (x, σ)
3: $Seeds \leftarrow \{p \in x_\sigma | \nabla x_\sigma(p) > \rho\}$
4: **while** $Seeds \neq \varnothing$ **do**
5: $\quad p_s \leftarrow \underset{p \in Seeds}{\operatorname{argmax}}(\nabla x_\sigma)$
6: $\quad g^x \leftarrow \varnothing$
7: $\quad$ **repeat**
8: $\quad\quad [s^x, p_{s_{new}}] \leftarrow$ region_growing$(p_s, \delta, \rho, \nabla x_\sigma)$
9: $\quad\quad g^x \leftarrow$ region_chaining(g^x, s^x)
10: $\quad\quad p_s \leftarrow p_{s_{new}}$
11: $\quad$ **until** g^x not convex or not smooth
12: $\quad a^x, \overline{a}^x \leftarrow$ fit_conic$(g^x, \nabla x_\sigma)$
13: $\quad g^x \leftarrow$refine(g^x)
14: $\quad a^x \leftarrow$refine(a^x)
15: $\quad \overline{a}^x \leftarrow$refine$(\overline{a}^x)$
16: $\quad c^x \leftarrow \underset{c \in \{g^x, a^x, \overline{a}^x\}}{\operatorname{argmin}} \mathrm{NFAc}(c)$
17: $\quad$ **if** $\mathrm{NFAc}(c^x) < 1$ **then**
18: $\quad\quad$ Add c^x to the corresponding set $\mathcal{E}$ or $\mathcal{S}$
19: $\quad\quad$ Remove supporting pixels of c^x from $Seeds$
20: $\quad$ **end if**
21: **end while**

6. We implemented a version of ELSDc that computes equation (5) exactly using arbitrary-precision arithmetic (GMP library, gmplib.org). This version is of course much slower. The results obtained are essentially the same as when using the upper-bound, with the occasional appearance of a new detection.

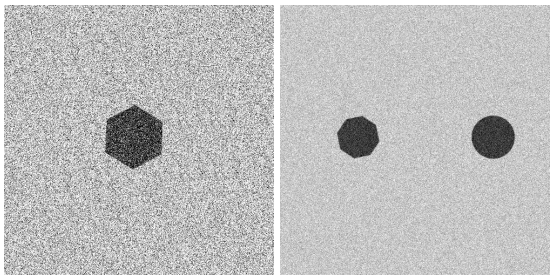


Fig. 4. Examples of test images used to compare NFAc and NFAd. Left: A polygon with 6 sides with a circumscribed circle radius of 60 pixels; the noise variance is 0.1. Right: A polygon with 8 sides and a circle with radius of 40 pixels; the noise variance is 0.075.

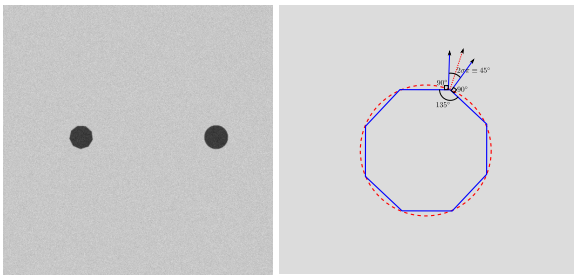


Fig. 5. Left: A 9-sides polygon with circumscribed circle of 25 pixels radius, and a circle with 25 pixels radius. This is an example where NFAc is not able to choose the correct polygonal interpretation. Right: An octagon and its circumscribed circle; all the pixels aligned with the polygon are also aligned with its circumscribed circle.

The complexity of the overall detection procedure is dictated by the candidate generation step, which considers every pixel of the image as seed at most once. In the worst case scenario, all the other pixels of the image are added to the current support region, but no primitive candidate is validated by the validation procedure, so no pixel is removed from the set of seeds. The next seed would again gather all the pixels into a support region, and so on. Hence, the complexity of the algorithm is quadratic in the number of pixels $O((mn)^2)$. However, this case does not occur, either because not all pixels can be seeds (because of low gradient magnitude), or because some candidates do get validated, hence their supporting pixels are removed from the seeds set. In a typical case, the complexity is $O(mnh)$, where h is the mean support region size, usually negligible relative to mn. In practice, the complexity is roughly linear in the number of pixels of the image.

6 Comparison of NFAc and NFAd

The discrete NFA formulation (NFAd) depends on the precision parameter δ. In this section we will highlight its influence along with the advantages of using the parameter-less continuous formulation (NFAc) in both validation and model selection tasks. Thus, two sets of experiments will be presented, the first comparing the validation capabilities and the second concentrating on model selection.

The experiments described here were carried out on 500×500 pixels synthetic images of circles and polygons, altered by different levels of Gaussian noise. On a grey scale where 0 corresponds to black and 1 to white, the grey level of the objects is 0.2, and the background is 0.8 (see Fig. 4).

For the validation task, we are interested in the robustness to noise of the two criteria, NFAc and NFAd. To this end, we generated images of hexagonal primitives, altered with different levels of Gaussian noise, one hundred images per noise level (see Fig. 4 left). Since the goal here is to test the theoretical capabilities of the validation part independently of the quality of the candidates proposed by the heuristic candidate generation step, we use the ground truth as candidate. For each criterion we counted the number of images in which the hexagon is validated. Table 1 reports the success rates for the two criteria. It can be observed that NFAc is able to validate the candidates at much higher noise levels, hence being more robust to noise, owing to the fact that it captures at a finer level the structuredness of a primitive, whereas NFAd considers the structuredness as a binary (rough) characteristic.

TABLE 1
Success rate in primitive validation using the two *a contrario* criteria NFAc and NFAd (see text).

Criterion/Noise level	0.01	0.025	0.05	0.075	0.1
NFAc	1.0	0.89	0.41	0.21	0.06
NFAd	1.0	0.88	0.05	0.0	0.0

For the model selection task, to set the limits of the two NFA formulations in distinguishing geometric models, we consider the case when a polygon and a circle are present in an image (see Fig. 4 right), and the candidate models for each primitive are a polygon and a circle. We run tests on synthetic images containing polygons with different number of sides, and circles with different radii.

We generated images containing regular cyclic polygons (all vertices belong to the same circle) with 6 to 12 sides. The radii of the circumscribed circles lay between 10 and 50 pixels, with a 5-pixels step between two consecutive radius values. The images were contaminated with Gaussian noise with variance 0.01. For each couple (number of sides, radius), we generated 100 images. The chosen noise level is the maximum level at which both criteria validate the primitives in all cases (see table 1), which allows to evaluate the model selection capabilities independently of the validation part.

We give quantitative results only for the case when the ground truth primitive is a polygon. The case when the ground truth primitive is a circle does not raise difficulties; a simple explanation is that the number of points aligned with the polygonal candidate will be at most equal to the number of points aligned with the circle candidate, whilst the number of tests for the polygon will be considerably larger, penalising its increased complexity. Thus, the polygonal candidate will be less meaningful than the circle candidate. This is a clear example of the general trade-off between model complexity and goodness of fit, underlying the model selection theory. It reflects well the pertinence of using NFA as model selection criterion, since it chooses the model that fits well the data, while avoiding overfitting by penalising the complexity of the model.

For the polygonal primitive, for the same reason as before, the polygonal candidate was obtained using the

TABLE 2
Success rate in model selection when using NFAc and NFAd. The true primitive is a polygon with different number of sides and different circumscribed circle radius. The competing models are the ground truth polygon and a fitted circle.

Radius	10 pixels		20 pixels		25 pixels		30 pixels		35 pixels		40 pixels		50 pixels	
Sides	NFAc	NFAd	NFAc	NFAd	NFAc	NFAd	NFAc	NFAd	NFAc	NFAd	NFAc	NFAd	NFAc	NFAd
6 sides	0.02	0.00	1.00	0.01	1.00	0.29	1.00	0.56	1.00	0.77	1.00	0.94	1.00	0.99
7 sides	0.00	0.00	0.59	0.00	0.97	0.00	0.99	0.02	1.00	0.13	1.00	0.37	1.00	0.86
8 sides	0.00	0.00	0.02	0.00	0.70	0.00	0.92	0.00	0.98	0.00	1.00	0.01	1.00	0.17
9 sides	0.00	0.00	0.00	0.00	0.04	0.00	0.51	0.00	0.76	0.00	0.98	0.00	1.00	0.00
10 sides	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.29	0.00	0.75	0.00	0.99	0.00
11 sides	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.29	0.00	0.91	0.00
12 sides	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.37	0.00

ground truth vertices, so that the results obtained are independent with respect to the quality of the candidates proposed by the candidate generation step. The width of the polygon was computed as to cover all the points that are δ-aligned with the sides of the polygon. Afterwards, we computed the circular ring that fits and covers these points. The discrete and continuous NFA were computed for each candidate. The interpretation possessing the smallest NFA was deemed the correct one. Table 2 reports the correct guess rate for the two criteria.

For both NFA formulations, we can observe that the results are in accordance with our expectation: at small radius, it is difficult to distinguish between a polygon and a circle as the number of sides increases. For greater values of the radius, the selection precision improves.

Clearly, NFAc shows a better performance in model selection tasks than NFAd, as it is able to capture more precisely the quality of a point to be aligned with a certain primitive. As shown in Table 2, NFAc's success rate in model selection is superior to the one of NFAd, and the cases where NFAc has difficulties are expectable. For example, in the 9-sides case and the given level of noise, NFAc is not able to choose the correct interpretation below 25 pixels radius (see Fig. 5, left), returning the circle as the best interpretation. NFAd's model selection capabilities are inherently limited by the precision parameter δ. A simple way to illustrate this is by considering the toy example from Fig. 5 right, which contains an octagon. Assuming a noise-free image, since the precision parameter is $\delta = \frac{1}{8}$ (corresponding to a 22.5° angle), it can be observed that any point that lies on the octagon contour is also δ-aligned with the circumscribed circle. This is actually the case for any polygon with at least 8 sides. Hence in this case, it is not the contour pixels' quality of being δ-aligned with the primitive that will point to the correct geometric interpretation. Instead, the decision will depend on the number of tests (which will falsely favour the circular candidate), and on the circular candidate's ability to cover the polygon contour with a ring as thin as possible. The continuous formulation circumvents elegantly this problem.

7 Experimental Evaluation of ELSDc

In this section, we present quantitative and qualitative results to highlight ELSDc's main strengths and weaknesses and position it with respect to existing works.

7.1 Baseline Algorithms and Setup

Since ELSDc covers several types of primitives, our comparative analysis uses several publicly available geometric primitive detectors as baselines: Prasad ellipse detector[7] [37], Hough (circle and ellipse) detectors [4], [52], and Etemadi line segment and circular arc detector[8] [13].

The edge maps needed for Prasad, Hough, and Etemadi algorithms are obtained using Canny edge detector [7] with $(0.1, 0.2, 1.0)$ for the low threshold, high threshold, and variance, respectively. To limit the number of false positives possibly reported on noisy images, the edge maps are *cleaned* before detection by discarding all edge chains with less than 5 edge points. This is the setting indicated by the authors of Prasad algorithm [37]. For ELSDc, no prior edge detection is needed.

Based on the number of true positives (TP), false positives (FP), and false negatives (FN)[9], we compare the detectors in terms of precision, recall, and F_β-measure, for $\beta \in \{1, 2\}$, with precision $= \frac{\text{TP}}{\text{TP+FP}}$, recall $= \frac{\text{TP}}{\text{TP+FN}}$, and $F_\beta\text{-measure} = (1+\beta^2)\frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$. A detection is considered a true positive if the area resulted from the intersection of a true primitive with a detected primitive, over the area resulted from their union, is superior to a predefined threshold; we use the value 0.9 as in [8], [37].

7.2 Test Datasets

We first report quantitative results on two datasets of synthetic and natural images, respectively. *Dataset1* consists of 20 synthetic images, 500×500 pixels, containing possibly overlapping circles to simulate occlusions; any degree of occlusion is allowed. By corrupting the images with five different levels of Gaussian noise, with five different noise realisations generated for each noise level and for each image, we obtain 500 image instances in total. We add ten images of pure Gaussian noise, where all detections are necessarily false positives[10]. We use images of circles (and not ellipses) to be able to include Etemadi detector

7. The implementation available online for Prasad detector is not complete sites.google.com/site/dilipprasad/source-codes. We implemented the missing validation part using the published paper [37].

8. Code available online for Etemadi detector github.com/encuadro/encuadro/tree/master/c/ort/ORT-2.3.

9. TP – a primitive correctly detected; FP – no primitive exists, but the algorithm reports a detection; FN – a primitive exists, but the algorithm does not detect it.

10. Due to space limitations, we discuss this sanity check in more detail in the supplemental material.

as well. For the Hough detector, we consider the circle variant in this experiment: having a reduced number of parameters compared to the elliptical detector, it performs better. Sample images are given in Fig. 6, first row.

Justified by the recurrent application of geometric primitive detectors for calibration patterns detection, *Dataset2* consists of 40 natural images of patterns included in Higuchi et al.'s package for camera calibration [23]. The patterns contain coplanar disjoint and concentric circles; see samples in Fig. 7, first row. We manually labelled the contours belonging to circles and rings projections to obtain the ground truth. In this experiment we compared ELSDc against Prasad and Hough ellipse detector, since the circles and ring boundaries project to ellipses. Etemadi was not included in the experiment, as it is not able to handle ellipses.

We eventually ran a third set of experiments to evaluate the performance of the detectors on natural images that exhibit various common challenges: reduced resolution, complex geometric primitives, background clutter. This evaluation is given only qualitatively, owing to the fact that establishing a reliable ground truth necessary for quantitative evaluation on arbitrary natural images is far from trivial. We illustrate the typical behaviour of the detectors by means of examples. As we will see, these examples confirm the quantitative results reported on the previous two datasets. Results of ELSDc, Prasad, Etemadi, and Hough circle detector are shown in Fig. 9 to illustrate their typical behaviour; most often than not, Hough ellipse detector reports no detection on arbitrary natural images, showing its limitations in dealing with complex image content. Additional qualitative results on natural images can be found at https://github.com/viorik/ELSDc, together with the source code and an online demo of our detector; all the test data used here and the associated ground truth labels for the first two datasets are equally included.

7.3 Results

a) The experiment on *Dataset1* compares the robustness against noise of ELSDc, Prasad, Etemadi, and Hough circle detector by reporting their mean precision, and the accuracy on incomplete contours (occlusions), reflected in their mean recall. The results are summarised in Fig. 8 and the typical behaviour of each detector can be observed in the examples in Fig. 6. For increasing noise level, Etemadi exhibits the most significant drop in accuracy, lacking tools to guard against noisy input. This is clearly observed in the last two columns of Fig. 6, where Etemadi reports many false detections. Hough performs poorly and is also affected by noise, as can be observed in Fig. 6. Prasad is not strongly affected by noise due to its restrictive detection thresholds, which prune most of the detections occurred in noise. However, the same restrictiveness is responsible for false negatives on incomplete (occluded) contours (see first three columns of Fig. 6), which explains the lower recall of Hough and Prasad. Due to its principled self-tuning thresholds, ELSDc does not encounter issues with noise, reporting a precision close to 1. ELSDc's recall is equally close to 1, due to the fact that ELSDc models explicitly elliptical arcs, being able to handle any degree of occlusion. The false positives issue

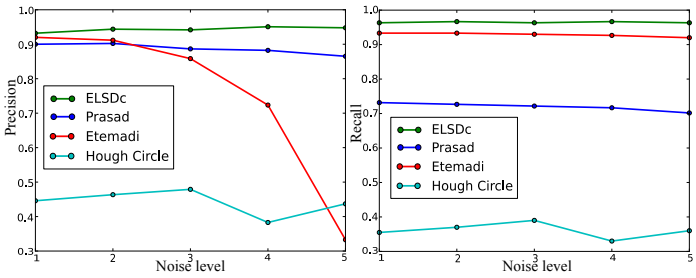


Fig. 8. Precision and Recall obtained by ELSDc, Prasad, Etemadi, and Hough circle detector, on synthetic images.

TABLE 3
Evaluation of ELSDc, Prasad, and Hough detectors on calibration pattern images (see text).

	Mean precision	Mean recall	Mean F_1-measure	Mean F_2-measure
ELSDc	0.8831	**0.9023**	**0.8926**	**0.8984**
Prasad	**0.9796**	0.6914	0.8106	0.7346
Hough	0.5795	0.1984	0.2956	0.2284

is visible at the extreme for Prasad, Hough, and Etemadi, when running the detectors on pure noise images (see Fig. 6, last column), where all detections are false positives. The cleaning step in this case is not reliable, and the edge maps are highly cluttered. On average, Prasad reports 67.4 detections per noise image, Hough 3616.3 detections, and Etemadi 459.2 detections; ELSDc reports no detection, reflecting the ability of the underlying statistical setup to guard against false positives.

b) Even if the images in *Dataset2* are relatively simple, the detection task is far from trivial (see Fig. 7). Specifically, the high number of primitives has negative influence on the Hough detector, reducing drastically its recall; this is a well-known issue for Hough detectors, and is explained by the excessive clutter generated in the Hough space when many primitives are present in the image. Prasad detector has difficulties especially on ring images; performing a pooling step to group arcs belonging to the same ellipse, this task is error-prone for concentric circles, where arcs belonging to different circles can be erroneously grouped, and eventually discarded, resulting in low recall. ELSDc reports satisfactory results regardless of the distribution of the primitives, or their number. However, the high distortions do affect ELSDc, which tends to report inaccurate detections on ring or circle contours located towards the borders of the image where the distortions are more severe; ELSDc reports polygons instead of ellipses, reducing the recall rate; the precision is also affected by reporting inaccurate elliptical arcs. Table 3 summarises the results of the three detectors. Prasad reports a better precision score, undermined however by the poor recall rate, observed also in Fig. 7, 3rd row. Overall, ELSDc reports better F-measure values.

c) The results of the different detectors on arbitrary natural images are illustrated in Fig. 9. The first example (1st column) shows the accuracy in detecting calibration patterns at reduced resolution. The results of Prasad, Etemadi, and Hough are decisively influenced by the edge detection step, which partially discards valid contours. Moreover, covering

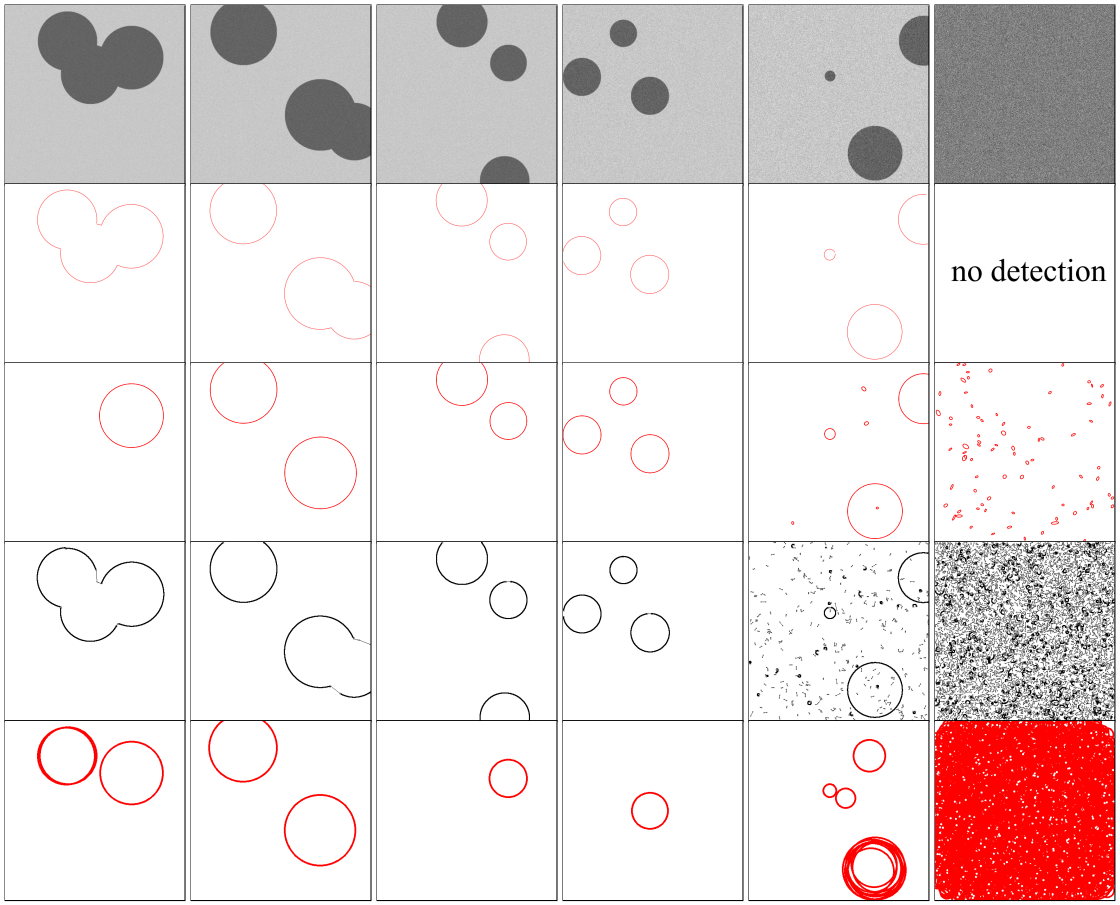


Fig. 6. 1st row: samples of synthetic images, altered by Gaussian noise, included in Dataset 1. The noise level increases from left to right. The rightmost image contains pure Gaussian noise; 2nd row: ELSDc result (red: detected elliptical arcs); 3rd row: Prasad result (red: detected ellipses); 4th row: Etemadi result (bolded curve: detected circles, regular line: detected line segments); 5th row: Hough circle result (red: detected circles). Even if the images contain circles, ELSDc and Prasad report ellipses, which in most of the cases are very close to circles, as expected.

only circular arcs and line segments, Etemadi reports as expected an oversegmentation of the ellipse contours into circular arcs; Hough detector gets confused in the presence of multiple primitives and reports a single circle. Operating directly on the grey-scale image, ELSDc is able to report accurate detections.

The second column illustrates the behaviour of the detectors on common natural images containing highly cluttered (textured) background. The results of Prasad and Hough have very poor accuracy, being drowned in false positives. Etemadi reports some correct detections (good recall), along with numerous useless detections (poor precision); this behaviour was observed equally in Fig. 6. ELSDc reports correct detections and a reduced number of detections on textured regions. This is due to the *a contrario* model: although the background does not follow exactly the $\mathcal{H}_0$ model, ELSDc is still able to discard most of the detections on it, owing to the fact that $\mathcal{H}_0$ is a good model for isotropic regions, which is the case for many textures [35].

The third example (3rd column) illustrates the role of model selection capabilities in limiting the number of false detections. Prasad reports ellipses on most of the polygonal contours in the background, but it misses the pupil contour, showing its difficulties in handling incomplete ellipses, as observed in Fig. 6 as well. Covering different families of primitives, Etemadi and ELSDc are able to report interpretations in agreement with the geometric image content. The improved accuracy of ELSDc is due mainly to the new continuous NFA criterion.

Finally, the geometric content of the last example (4th column) puts in difficulty our detector, and more precisely its candidate generation part, which does not succeed in proposing an optimal segmentation of the spiral contour into circular arcs; hence the polygonal interpretations are preferred. Prasad and Hough report unsatisfactory interpretations, confirming their difficulties in handling concentric circles. Etemadi is capable of reporting an interpretation closer to our expectation, due to its segmentation capabilities. However, Etemadi cannot be straightforwardly used as hypothesis generator for our model since it does not generate ellipse primitives.

8 Conclusion

We proposed an automatic procedure for line segment and elliptical arc detection, based on the *a contrario* approach. The main concern is to avoid false detections induced by noise or by erroneous interpretation of the image content in terms of geometric primitives. To this end, the proposed procedure follows three essential steps: *candidate generation*

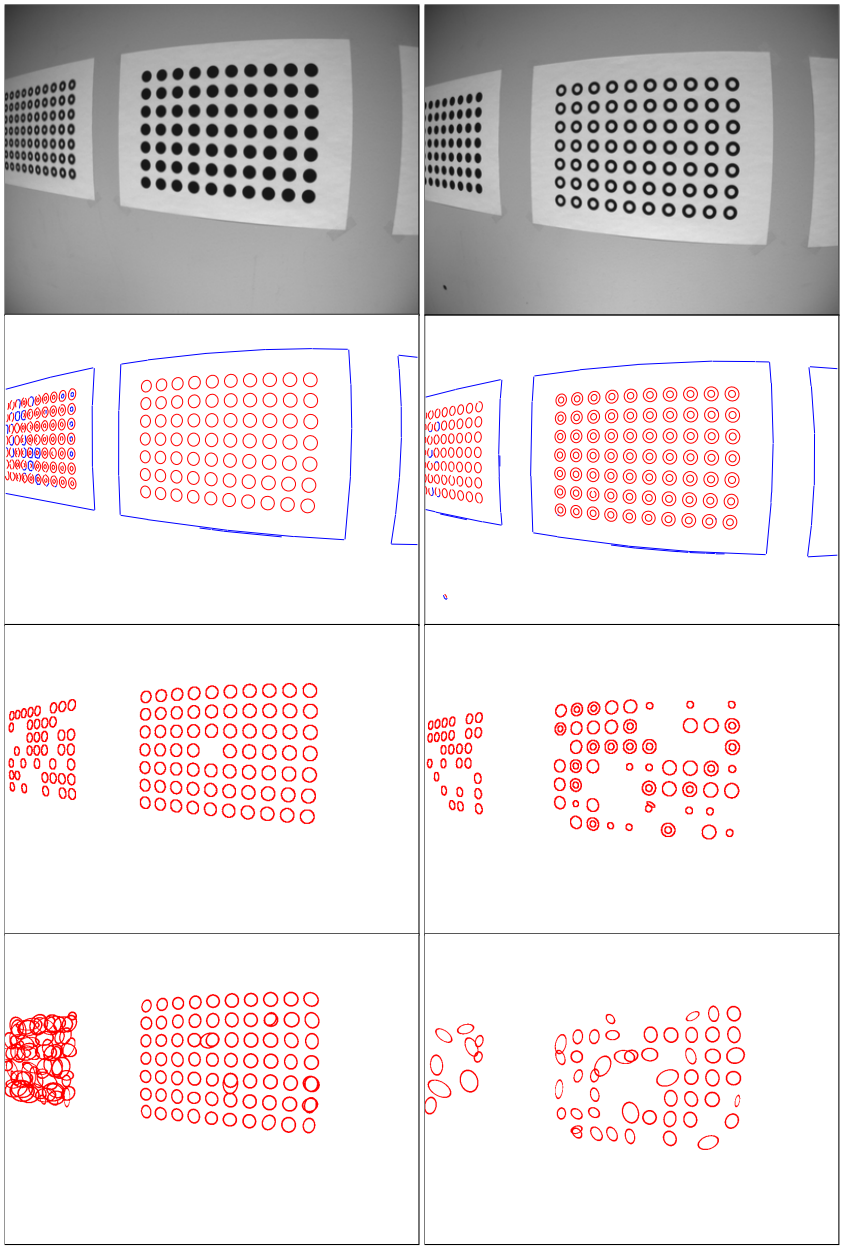


Fig. 7. 1st row: sample pattern images [23], 2nd row: ELSDc result (red: detected ellipse arcs, blue: detected polygons), 3rd row: Prasad result (red: detected ellipses), 4th row: Hough ellipse result (red: detected ellipses).

(to find potential candidates for the sought primitives), *candidate validation* (to eliminate false detections produced by noise), and *model selection* (to avoid misinterpretations of geometric models). Our study includes polygonal lines and elliptical arcs as elementary candidate primitives. The essential quantity of the *a contrario* approach, the Number of False Alarms (NFA), is used for both validation and model selection purposes. We introduced a continuous NFA formulation which exhibits better performance in validation and model selection tasks, compared to the discrete NFA. Note that the proposed validation and model selection scheme could be included as an add-on to any existing primitive detector, in order to reduce its false positive rate, without significant computational cost, as the proposed structuredness scores are very simple to compute.

Although satisfactory compared to existing works, the performance of the proposed detector is limited in handling complex primitives, due mainly to the candidate generation step. A pre-segmentation step could be considered as future improvement. Alternatively, it could be useful to look for more meaningful sub-polygons (and corresponding sub-arcs) during the refinement stage. Related to the model selection step, an aspect that needs further consideration is the weight associated to each sub-family of polygons when distributing the total quantity of allowed false positives among the different families.

ACKNOWLEDGEMENTS

This work was partially funded by the Qualcomm postdoctoral program at École Polytechnique Palaiseau, a Google

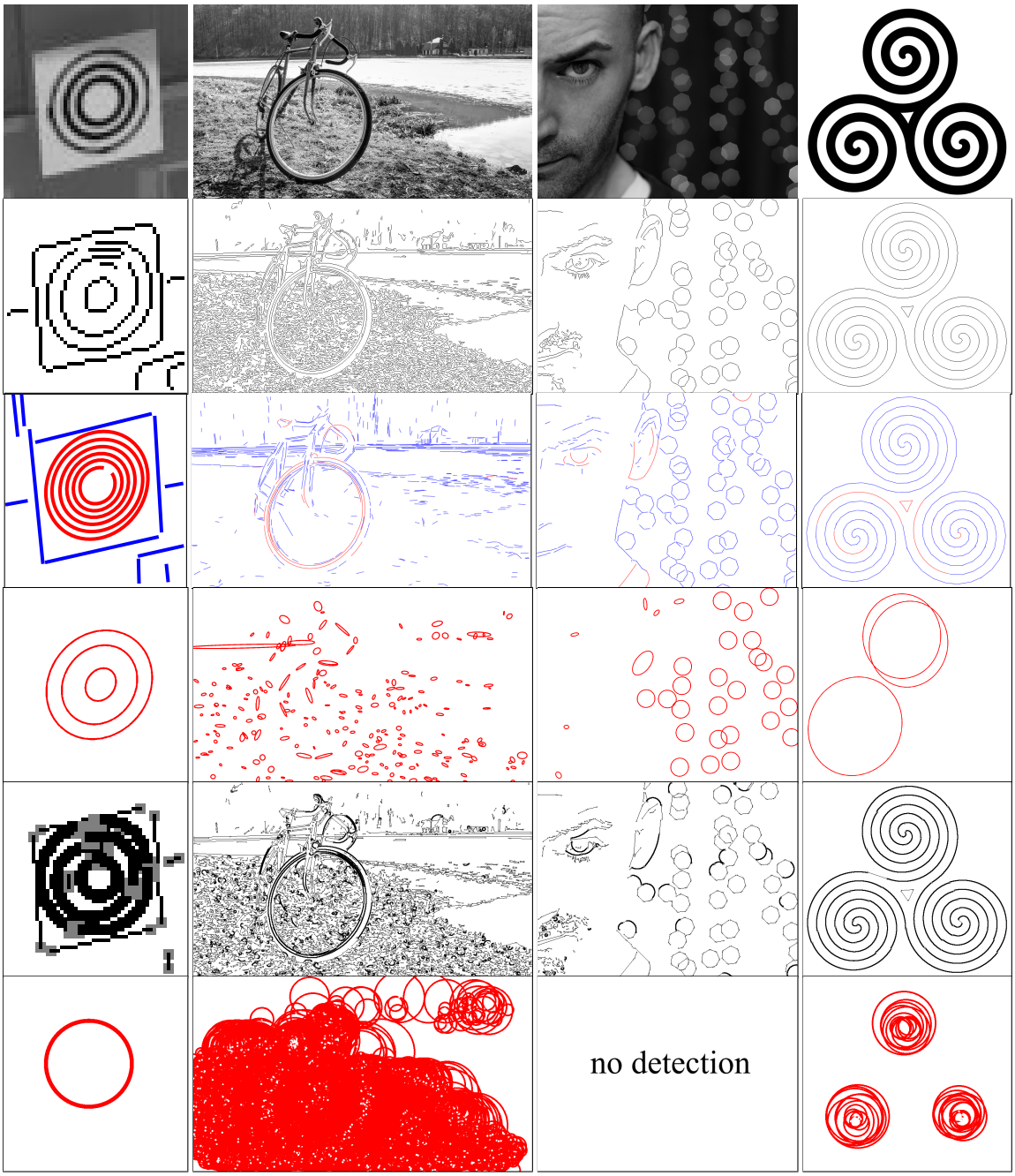


Fig. 9. Results on natural images. 1st row: original image; 2nd row: Canny edges – see text for parameters used; 3rd row ELSDc result (red: detected ellipses, blue: detected polygons); 4th row: Prasad (red: detected ellipses); 5th row: Etemadi (bold curves: detected circle arcs, regular lines: detected line segments); 6th row: Hough circle detector (red: detected circles).

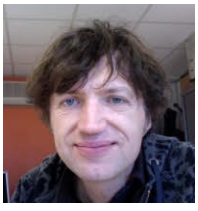
Faculty Research Award, the Marie Curie grant CIG-334283-HRGP, a CNRS chaire d'excellence and chaire Jean Marjoulet, and EPSRC grant EP/L010917/1.

REFERENCES

[1] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," 1981, vol. 13, pp. 111–122.
[2] O. Barinova, V. Lempitsky, and P. Kohli, "On detection of multiple object instances using Hough transforms," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2233–2240.
[3] A. Bonci, T. Leo, and S. Longhi, "A bayesian approach to the Hough transform for line detection," *IEEE Trans. Syst., Man, Cybern.*, vol. 35, no. 6, pp. 945–955, 2005.
[4] G. Bradski, "Opencv," *Dr. Dobb's Journal of Software Tools*, 2000.
[5] J. Burns, A. Hanson, and E. Riseman, "Extracting straight lines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 4, pp. 425–455, 1986.
[6] L. L. Cam, "The central limit theorem around 1935," *Statistical Science*, vol. 1, no. 1, pp. 78–91, 1986.
[7] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.
[8] A. Chia, S. Rahardja, D. Rajan, and K. Leung, "A split and merge based ellipse detector with self-correcting capability," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1991–2006, 2011.
[9] R. Dahyot, "Statistical Hough transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1502–1509, 2009.
[10] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *Int. J. Comput. Vision*, vol. 40, no. 1, pp. 7–23, 2000.

[11] ——, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, 1st ed. Springer Publishing Company, Incorporated, 2007.
[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
[13] A. Etemadi, "Robust segmentation of edge data," in *Int. Conf. Image Processing and Applications*, 1992, pp. 311–314.
[14] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, 1999.
[15] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic Hough transform for line detection," in *Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 1999.
[16] A. Gordon, G. Glazko, X. Qiu, and A. Yakovlev, "Control of the mean number of false discoveries, Bonferroni and stability of multiple testing," *Ann. Appl. Stat.*, vol. 1, no. 1, pp. 179–190, 2007.
[17] J. Y. Goulermas and P. Liatsis, "Incorporating gradient estimations in a circle-finding probabilistic Hough transform," *Pattern Analysis and Applications*, vol. 2, 1999.
[18] R. Grompone von Gioi, *A contrario line segment detection*. Springer, 2014.
[19] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.
[20] R. Grompone von Gioi and J. Jakubowicz, "On computational gestalt detection thresholds," *J. Physiology – Paris*, vol. 103, no. 1-2, pp. 4–17, 2009.
[21] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.
[22] J. Heikkila, "Geometric camera calibration using circular control points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1066–1077, 2000.
[23] M. Higuchi, A. Datta, and T. Kanade, "Method and Means for Recognizing Complex Patterns," http://www.ri.cmu.edu/research_project_detail.html?project_id=617&menu_id=261, 1962.
[24] P. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3.069.654, 1962.
[25] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *Int. J. Comput. Vision*, vol. 97, no. 2, pp. 123–147, 2012.
[26] Q. Ji and R. M. Haralick, "Error propagation for the Hough transform," *Pattern Recogn. Lett.*, no. 22, pp. 813–823, 2001.
[27] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Vol. 1 (Wiley Series in Probability and Statistics)*, volume 1 ed. Wiley-Interscience, 1995.
[28] K. Kanatani, "Comments on "nonparametric segmentation of curves into various representations"," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1391–1392, 1997.
[29] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*, 3rd ed., ser. Springer Texts in Statistics. Springer, 2005.
[30] H. Li, M. A. Lavin, and R. J. Le Master, "Fast Hough transform: A hierarchical approach," *Comput. Vision Graph. Image Process.*, vol. 36, no. 2-3, pp. 139–161, 1986.
[31] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Norwell, MA, USA: Kluwer Academic Publishers, 1985.
[32] F. Mai, Y. Hung, H. Zhong, and W. Sze, "A hierarchical approach for fast and robust ellipse extraction," in *Int. Conf. Image Processing (ICIP)*, vol. 5, 2007.
[33] L. Moisan, P. Moulon, and P. Monasse, "Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers," *Image Processing On Line*, vol. 2, pp. 56–73, 2012.
[34] L. Moisan and B. Stival, "A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix," *Int. J. Comput. Vision*, vol. 57, no. 3, pp. 201–218, 2004.
[35] A. Myaskouvskey, Y. Gousseau, and M. Lindenbaum, "Beyond independence: An extension of the a contrario decision procedure," *Int. J. Comput. Vision*, vol. 101, no. 1, pp. 22–44, 2013.
[36] G. Perrin, X. Descombes, and J. Zerubia, "A marked point process model for tree crown extraction in plantations," in *Int. Conf. Image Processing (ICIP)*, vol. 1, 2005.
[37] D. K. Prasad, M. K. H. Leung, and S.-Y. Cho, "Edge curvature and convexity based ellipse detection method," *Pattern Recogn.*, vol. 45, no. 9, pp. 3204–3221, 2012.
[38] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. New York, NY, USA: Cambridge University Press, 2007.
[39] V. Pătrăucean, R. von Gioi, and M. Ovsjanikov, "Detection of mirror-symmetric image patches," in *Int. Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 211–216.
[40] V. Pătrăucean, "Detection and identification of elliptical structure arrangements in images: theory and algorithms," 2012. [Online]. Available: http://oatao.univ-toulouse.fr/6985/
[41] V. Pătrăucean, P. Gurdjos, and R. G. von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *12th European Conf. on Computer Vision (ECCV)*, 2012, pp. 572–585.
[42] J. Rabin, J. Delon, and Y. Gousseau, "A statistical approach to the matching of local features," *SIAM J. Img. Sci.*, vol. 2, no. 3, pp. 931–958, 2009.
[43] P. L. Rosin, "Ellipse fitting using orthogonal hyperbolae and Stirling's oval," *Graphical Models and Image Processing*, vol. 60, no. 3, pp. 209–213, 1998.
[44] B. Rougé, V. Caselles, A. Almansa, L. Garrido, J. Preciozzi, and L. Igual, "Automatic low baseline stereo in urban areas," *Inverse Problems and Imaging*, vol. 1, no. 2, pp. 319–348, 2007.
[45] R. T. Rust, D. Simester, R. J. Brodie, and V. Nilikant, "Model selection criteria: An investigation of relative accuracy, posterior probabilities, and combinations of criteria," *Management Science*, vol. 41, no. 2, pp. 322–333, 1995.
[46] A. Senior, R.-L. Hsu, M. A. Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 696–706, 2002.
[47] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *Proc. British Machine Vision Conf. (BMVC)*, 2006, pp. 17–26.
[48] A. Soetedjo and K. Yamada, "Fast and robust traffic sign detection," in *Int. Conf. Syst., Man, Cybern.*, vol. 2, 2005, pp. 1341–1346.
[49] C. Stewart, "MINPRAN: a new robust estimator for computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 10, pp. 925–938, 1995.
[50] G. Thomas, "Real-time camera tracking using sports pitch markings," *J. Real-Time Imag. Proc.*, vol. 2, no. 2-3, pp. 117–132, 2007.
[51] G. W. West and P. L. Rosin, "Multistage combined ellipse and line detection," in *Proc. British Machine Vision Conf. (BMVC)*, 1992, pp. 197–206.
[52] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized hough transform (rht)," *Pattern Recogn. Lett.*, vol. 11, no. 5, pp. 331–338, 1990.

Viorica Pătrăucean received the B.Sc. degree in Computer Engineering from the Military Technical Academy of Bucharest (2008). She obtained the M.Sc. (2008) and Ph.D. (2012) in Computer Science from the Institut National Polytechnique de Toulouse, France. After a postdoc at École Polytechnique-INRIA, Paris (2014), she is currently a research associate with the Engineering Department at the University of Cambridge, UK.



Pierre Gurdjos is a CNRS research engineer. He received his Eng. Degree in Computer Science and Applied Mathematics from ENSEEIHT, Toulouse France and his M.Sc. in Computer Science from the University of Toulouse III. He is currently a member of the VORTEX team at the IRIT laboratory, working on many different topics in Computer Vision, with particular attention to structure-from-motion and camera tracking.

Rafael Grompone von Gioi received the B.Sc. degree in Electrical Engineering from the Universidad de la República, Uruguay (2004). He then obtained the M.Sc. (2006) and Ph.D. (2010) in Applied Mathematics from the Ecole Normale Supérieure de Cachan, France. He is currently a post-doc at the Ecole Normale Supérieure de Cachan.