J Math Imaging Vis
DOI 10.1007/s10851-016-0689-x

Detecting Curved Edges in Noisy Images in Sublinear Time

Yi-Qing Wang[1] · **Alain Trouvé**[2] · **Yali Amit**[3] · **Boaz Nadler**[1]

Received: 30 June 2016 / Accepted: 2 November 2016

Abstract Detecting edges in noisy images is a fundamental task in image processing. Motivated, in part, by various real-time applications that involve large and noisy images, in this paper we consider the problem of detecting long curved edges under extreme computational constraints, that allow processing of only a fraction of all image pixels. We present a sublinear algorithm for this task, which runs in two stages: (1) a multiscale scheme to detect curved edges inside a few image strips; and (2) a tracking procedure to estimate their extent beyond these strips. We theoretically analyze the runtime and detection performance of our algorithm and empirically illustrate its competitive results on both simulated and real images.

Keywords Curved edge detection · Sublinear algorithm · Noisy images

Alain Trouvé and Yali Amit were partially supported by NSF-DMS 0706816.

✉ Boaz Nadler
boaz.nadler@weizmann.ac.il

Yi-Qing Wang
yqwang9@gmail.com

Alain Trouvé
trouve@cmla.ens-cachan.fr

Yali Amit
amit@galton.uchicago.edu

[1] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

[2] CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France

[3] Departments of Statistics, Computer Science and the College, University of Chicago, Chicago, IL, USA

1 Introduction

Edge detection is a fundamental task in image analysis. Since the seminal works of Marr and Hildreth [21] and Canny [5], numerous edge detection algorithms have been proposed. Traditional methods mainly focused on step edges and relied on discretized differentiation operators for their detection [6,9,12]. In contrast, to trace semantically meaningful edges in natural images with significant texture content, several learning-based algorithms were developed over the past years [2,8,16].

In various applications, thousands of large and noisy images are acquired per hour and need to be quickly analyzed. Examples include high-throughput industrial imaging inspection systems, and cameras in either cars or autonomous drones, that detect lanes, power lines, to avoid crashes. The requirement to analyze many images in real time may limit the transfer from memory to CPU, or the analysis in the CPU, to only a small fraction of all image pixels. This raises an interesting statistical and computational question at the focus of our work: How well can edge detection be performed under such restrictive sublinear runtime constraints?

Even though the field of sublinear time algorithms is quite developed, see for example the review [26], relatively few sublinear methods have been proposed for image processing tasks. Notable exceptions include the early works on randomized and probabilistic Hough transforms [13,14,32]. More recent ones include [15,24,28] which applied sublinear property testing methods to binary images, and [17] which developed a sublinear time algorithm for approximate template matching in natural images. In a more general context of sparse signal recovery, in particular from large sensor networks, [10,31] developed adaptive sequential sampling strategies, where an initial portion of the sampling budget is used to crudely measure the signal, and only its statisti-

Published online: 11 November 2016

cally significant parts are kept for more refined measurements and analysis. These schemes can recover signals with significantly reduced communication and energy consumption. Closely related to our work is the paper by Horev et al. [11], who focused on detection of long straight edges in sublinear time. They proved that under a certain worst-case setting, sampling whole image columns is an optimal scheme. They developed a strip-based edge detection algorithm that consists of two main steps: (1) detection of straight edges inside few strips of the input image; (2) matching detections in adjacent strips and validating them.

In this work, we extend the approach of Horev et al. [11] to detect long *curved* edges in sublinear time. Our main contributions are the development of a fast multiscale algorithm to detect curved edges in image strips, a tracking algorithm to trace edges between strips and a theoretical analysis of the complexity and statistical accuracy of our method. A key insight of our work is that if the input image contains few long edges, there is no need to process all of its pixels, hence potentially leading to significant savings.

Following a precise formulation of our problem setup in Sect. 2, we first give a high-level description of our approach in Sect. 3. Motivated by [11], we also adopt their strip-based approach for the initial detection step. Their method, however, is not directly applicable to detecting curved edges, and there are two significant obstacles that need to be addressed. First, as our goal is to detect faint edges in possibly very noisy images, the strips considered need to be wide. In such wide strips, the curved edges cannot be well approximated by linear functions as done in [11]. Secondly, since edges are curved they cannot be easily matched in adjacent strips.

To handle the first problem, we approximate curved edges by low degree polynomials. As this significantly increases the set of candidate curves, a key question is whether processing this larger set is still compatible with the sublinear complexity constraint. We study this question in Sect. 4, under the assumption that image edges are smooth. We construct a finite set of polynomials so that any curved but sufficiently smooth edge can be accurately approximated inside the strip by one of these polynomials. We show that for a not-too-wide strip, this set is sufficiently small to allow a sublinear runtime. Specifically, in a strip of $O(n^\kappa)$ columns in an $n \times n$ image, the number of polynomials in this set is

$$\begin{cases} O(n^{1+\kappa}) & \kappa \in [0, \frac{1}{2}] \\ O(n^{3\kappa}) & \kappa \in \left(\frac{1}{2}, \frac{5}{8}\right) \end{cases}. \tag{1}$$

Furthermore, thanks to the efficient line integral algorithm [4], when these polynomials are substituted by their continuous piecewise linear interpolants (see Fig. 1), detecting curved edges in a strip of width $O(n^\kappa)$ columns can be carried out with time complexity

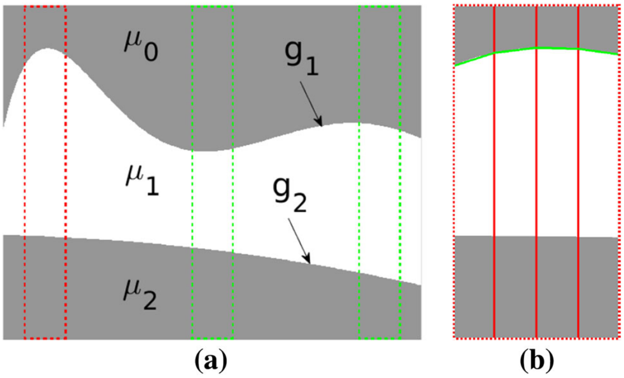


Fig. 1 **a** A piecewise constant image with two curved step edges and three equidistant strips. **b** A zoom-in of the leftmost strip. The *upper edge* is well approximated by both a quadratic polynomial (not shown) and its piecewise linear interpolant, composed of four equal-length segments (in *green*) (Color figure online)

$$\begin{cases} O(n^{1+\kappa}\log n) & \kappa \in \left[0, \frac{1}{2}\right] \\ O(n^{4\kappa-1/2}) & \kappa \in \left(\frac{1}{2}, \frac{5}{8}\right) \end{cases}. \tag{2}$$

Hence, as long as the image strip has width $O(n^\kappa)$ with $\kappa < 5/8$, asymptotically in n, curved edge detection is possible in time sublinear in the image size n^2.

It is interesting to compare this result to [11]. There, detecting straight edges in a strip of width $O(n^\kappa)$ was possible in sublinear time for any $\kappa \in (0, 1)$, requiring $O(n^{1+\kappa}\log n)$ operations. For curved edges, in contrast, not only is there an upper bound on the strip width, but also for $\kappa > 1/2$ processing it takes significantly more operations, compared to straight edge detection.

The strip-based detection approach described above relies heavily on an a-priori assumption of the edge smoothness. In particular, it may miss strong but wiggly edges. To make our algorithm robust against such smoothness mis-specifications, in Sect. 5.1 we develop a scheme to detect curved edges in several sub-strips of different scales. Furthermore, we prove that under suitable conditions this scheme produces well-estimated edges with high probability.

Next, we consider the problem of tracing the edges outside of the narrow strips in which they were detected. Horev et al. [11] assumed straight edges, which enabled them to easily match detections across adjacent strips, and estimate the end location of detected edges. Their approach is not applicable to curved edges. In Sect. 5.2, we present an iterative *tracking* procedure which extends the edges detected in the strips to the rest of the image by sequentially exploring an adaptively determined sequence of small windows. In Sect. 5.3, we show that with a window width $O(n^\kappa)$, our tracking procedure costs at most $O(n^{2\kappa\vee 1})$ operations for $\kappa \in [0, \frac{5}{8})$. Hence, tracking has negligible computational cost compared to the detection step. Combining our analysis of the detection and tracking procedures allows us to describe the trade-off between the overall computational complexity of our algorithm and its minimal detectable edge saliency (Lemma 5).

Finally, to successfully apply our algorithm to real images, in Sect. 5.4 we address some practical issues such as edge localization and nonmaximal suppression. We present in Sect. 6 several illustrative results of our algorithm and runtime comparison with the Canny edge detector [5] and the line segment detector (LSD) [29,30], which are both fast methods with linear time complexity. We conclude the paper with a discussion and future work in Sect. 7.

2 Problem Setup

In this work, we develop an algorithm and accompanying theory for detecting curved edges in noisy images, in sublinear time. While our algorithm can be applied to arbitrarily sized images, for clarity of the exposition and the theoretical analysis, we assume that the observed images are square and contain step edges. Specifically, consider the following idealized image formation model. Let u_c be a piecewise constant function defined on the unit square $[0, 1]^2$. The boundaries between adjacent constant-valued regions are delineated by long, smooth and potentially curved edges. The observed $n \times n$ noisy image U is the result of impulse sampling of u_c on a regular lattice, corrupted by additive zero mean independent Gaussian noise: for $1 \leq i, j \leq n$,

$$U(i, j) = u_c\left(\frac{i}{n}, \frac{j}{n}\right) + \xi(i, j), \quad \xi(i, j) \sim N(0, \sigma^2). \quad (3)$$

For simplicity, we assume the noise variance σ^2 is known. When unknown, it can be estimated from the observed image by various methods, see [19,20].

Given the noisy image U, our goal is to detect and localize the step edges in it. Similar to [11], we wish to do it in time sublinear in the total number of image pixels. However, unlike this prior work which focused exclusively on straight edges, we develop an algorithm to detect curved smooth edges. On the theoretical front, we study its performance and establish the trade-off between the overall time complexity of our algorithm and its minimal detectable edge saliency.

Before proceeding, we first introduce some notations and the signal-to-noise ratio (SNR) of a step edge.

Notations For any $z \in \mathbb{R}$, $(z)_+ = \max(z, 0)$, $\lfloor z \rfloor$ denotes the largest integer less than or equal to z, whereas $\lceil z \rceil$ is the smallest integer larger than or equal to z. For $z_1, z_2 \in \mathbb{R}$, we write $z_1 \wedge z_2 = \min(z_1, z_2)$ and $z_1 \vee z_2 = \max(z_1, z_2)$. λ denotes the Lebesgue measure on $\mathbb{R}$. The constant α in $O(n^\alpha)$ is referred to as the *growth exponent*. With some abuse of notation, I may denote either an interval on the real line, $[x_1, x_L]$, or the image strip with columns $x_1, \ldots, x_L$.

From continuous to discrete edges and their SNR Consider for simplicity an image whose corresponding function u_c has two regions with different intensities, separated by a single step edge defined by a function f,

$$u_c(x, y) = \mu_1 1_{y \in (f(x), 1]} + \mu_2 1_{y \in [0, f(x)]}, \ \forall (x, y) \in [0, 1]^2.$$

According to the model (3), the edge f is dilated to $g : x \in [0, n] \mapsto nf(x/n)$, and in the observed image discretized to $\Gamma(g) : i \in \{1, \ldots, n\} \mapsto \lfloor g(i) \rfloor$. We denote the *edge contrast* by $\mu_{\Gamma(g)} = |\mu_1 - \mu_2|$, and its *signal-to-noise ratio* (SNR) by $\mu_{\Gamma(g)}/\sigma$.

3 A Strip-Based Approach

The constraint of a sublinear runtime implies that the edge detection algorithm can process only a fraction of all image pixels. To fix the idea, assume that for some $\kappa \in [0, 1)$, it observes $O(n^{1+\kappa})$ out of the n^2 image pixels. The first question is thus which pixels to process?

In [11], a related straight *fiber* detection problem was studied. Under the assumption that an unknown straight fiber spans the entire image width, it was shown that under a worst-case scenario, sampling an equal number of pixels in each of the n image rows is optimal for its detection. The reason is as follows. Consider a sampling strategy which distributes its sampling budget, say mn pixels, unevenly across the image rows. Then there is at least one row with strictly less than m observed pixels. As a result, this sampling scheme has lower power for detecting a fiber that passes through precisely this row. The same analysis carries over to detecting curved fibers.

Admittedly, what is optimal for detecting fibers may not be the same for detecting edges. Nonetheless, motivated by the above argument, as in [11] we adopt a strip-based approach with $O(n^\kappa)$ whole columns. As outlined in Algorithm 1 and illustrated in Fig. 2, our proposed sublinear time curved edge detection method consists of two main steps: (1) detect edges in a few strips extracted from the input image; (2) track the detected edges outside the strips.

Pixel responses Most methods to detect step edges are based on some discrete differentiation operator. Here we use pixel responses, which for horizontal edges are obtained by convolving the image with a vertical *mask* of length 2ω consisting of ω ones and ω values of -1,

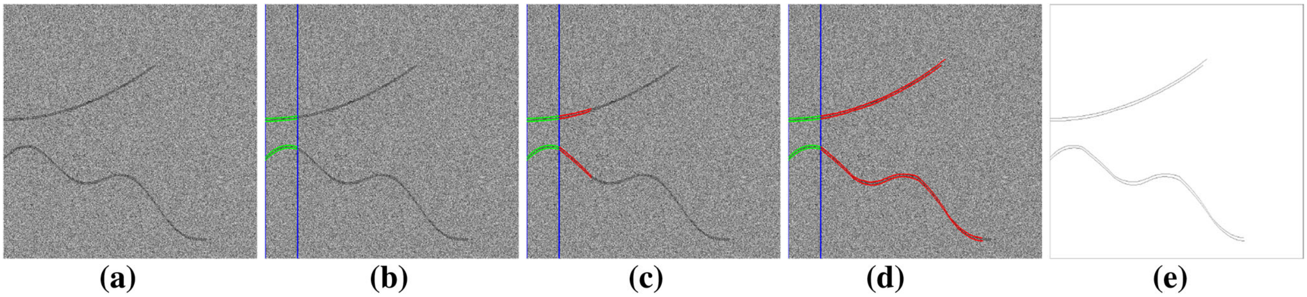


Fig. 2 Curved edge detection on a 1000 × 1000 synthetic noisy image. **a** Observed noisy image (SNR = 1/2). **b** Detected edges in the strip delimited by blue columns. **c** Output of the first tracking iteration. **d** Fully tracked edge estimates. **e** one-pixel-wide sketch of the final output

```
Algorithm 1 Sublinear Curved Edge Detection
  Input: an $n \times n$ noisy image
  Parameters: noise level, mask width, number of strips, detection strip width, detection false alarm rate, tracking extension width, tracking false alarm rate.
  Step 0: Extract several strips from the noisy image.
  Step 1: Detect edges in each of the strips (using detection false alarm rate).
  Step 2: Track the detected edges (using tracking false alarm rate).
```

$$R(i,j) = \frac{1}{\sqrt{2\omega}} \sum_{k=j+1}^{j+\omega} \Big(U(i,k) - U(i,k-\omega) \Big). \tag{4}$$

For other choices of masks and edge filters, see for example [23,27]. Clearly if a pixel (i,j) is at a vertical distance of at least ω pixels from any edge, then $\mathbb{E}R(i,j) = 0$. In contrast, if there is only one edge g close by, the *signal* of the pixel (i,j), defined as $|\mathbb{E}R(i,j)|$, is

$$|\mathbb{E}R(i,j)| = \frac{\mu_{\Gamma(g)}}{\sqrt{2\omega}} \big(\omega - |j - \lfloor g(i) \rfloor|\big)_+. \tag{5}$$

In particular, assuming that all image edges are separated by at least ω pixels, and ignoring boundary effects, the convolution (4) creates a signal tube of vertical width $2\omega - 1$ pixels around each edge g. Inside this tube the signal peaks on $\Gamma(g)$ and declines as the vertical distance from it increases. Clearly, a large value of ω is preferable for edge detection, as it leads to a stronger signal on the edge. However, since real images may contain edges separated by only a few pixels, too large values of ω may blur adjacent edges. In practice, ω is typically set to be an integer between 3 and 7.

Candidate curves and their edge responses To locate low-contrast edges in very noisy images, the individual pixel responses (4) may not be very informative. Rather, as in earlier works that handle high noise levels [11,22], we sum several pixel responses to form an *edge response*. Specifically, the edge response of a *candidate curve* $h : [x_1, x_L] \mapsto [\omega, n-\omega]$ is defined as

$$R(h) = \frac{1}{\sigma\sqrt{L}} \sum_{k=1}^{L} R(x_k, \lfloor h(x_k) \rfloor). \tag{6}$$

Due to the normalizing factor $\frac{1}{\sigma\sqrt{L}}$, the random variable $R(h)$ has unit variance. Its expectation depends on the distance between the candidate curve and the image edges. If h is at a vertical distance larger than ω from any edge, then $\mathbb{E}R(h) = 0$. In contrast, if it is uniformly close to an edge g, with $\sup_{x\in[x_1,x_L]} |g(x) - h(x)| \le \gamma$, for some integer $\gamma < \omega$, then

$$|\mathbb{E}R(h)| \ge \frac{\mu_{\Gamma(g)}}{\sigma} \cdot \frac{\sqrt{L}(\omega-\gamma)}{\sqrt{2\omega}}. \tag{7}$$

In particular, if $h = g$, then $|\mathbb{E}R(h)| = \frac{\mu_{\Gamma(g)}}{\sigma}\sqrt{\frac{L\omega}{2}}$.

Edge detection as hypothesis testing Similar in spirit to the *a-contrario* principle [7], we formulate edge detection in a strip as a multiple hypothesis testing problem. Given a finite set $\mathcal{S}$ of candidate curves, the null hypothesis is that the strip is edge-free and thus no candidate curve in $\mathcal{S}$ traces a real edge. The alternative is that at least one, but possibly more candidate curves $h \in \mathcal{S}$ trace actual edges. We compute the edge responses of all $h \in \mathcal{S}$ and retain for further analysis only those with a statistically significant response.

4 Edge Regularity and Search Spaces

To employ the above hypothesis testing approach we thus need to construct a suitable finite set $\mathcal{S}$ of candidate curves, referred to as a *search space*, devise a computationally efficient method to compute all its edge responses, and a suitable threshold to retain only the statistically significant ones. On the one hand, the search space $\mathcal{S}$ needs to be sufficiently large so that any image edge is well approximated by one of its candidate curves. On the other hand, to comply with the sublinear time constraint, its size must be significantly smaller than n^2. For these two conflicting requirements to hold, some regularity must be imposed on the edges. Similar to [3], we consider the following class of smooth edges:

Definition 1 A function $f : [0,1] \mapsto (0,1)$ is $\boldsymbol{b}$-regular with $\boldsymbol{b} = (b_1, b_2, \ldots, b_{r+1}) \in \mathbb{R}_+^{r+1}$ if it is at least $(r+1)$-times differentiable and satisfies

$$\frac{\|f^{(k)}\|_\infty}{k!} \le b_k, \quad k = 1, \ldots, r+1.$$

For future use, we denote the corresponding set of dilated edges on the square $[0,n]^2$ by

$$\mathcal{R}_{\boldsymbol{b},n} = \{g \mid f \text{ is } \boldsymbol{b}\text{-regular and } g : x \in [0,n] \mapsto nf\left(\tfrac{x}{n}\right)\}.$$

With detailed proofs in Appendix 3, the main result of this section (Theorem 1) is that for a sufficiently narrow strip, given a-priori knowledge of the vector $\boldsymbol{b}$, we can construct a search space $\mathcal{S}$ of quantized polynomials, such that: (1) the worst-case approximation error inside the strip, $\sup_{g \in \mathcal{R}_{\boldsymbol{b},n}} \min_{h \in \mathcal{S}} \|g - h\|_\infty$ is small; and (2) its size $|\mathcal{S}| \ll n^2$. Properties (1) and (2) allow the detection of curved edges in sublinear time.

Search space of quantized polynomials In our approach, we approximate curves by low-degree polynomials. Indeed, for any $g \in \mathcal{R}_{\boldsymbol{b},n}$, $|g^{(r+1)}(x)| \le b_{r+1}(r+1)!/n^r$. If this quantity is small, then by Taylor's theorem, locally g is well approximated by a degree r polynomial. As there are infinitely many polynomials of degree r, we quantize their coefficients to make them *searchable*.

Let $x_0, \ldots, x_{2L}$ be $2L+1$ consecutive integers. For any $g \in \mathcal{R}_{\boldsymbol{b},n}$, define its *symmetric approximator* over the interval $I = [x_0, x_{2L}]$ as

$$\mathcal{P}^s_{I,M_q}(g) = \sum_{k=0}^{r} \frac{a_k}{M_q} \left(\frac{\cdot - x_L}{L}\right)^k, \tag{8}$$

where

$$a_k = \left\lfloor \frac{g^{(k)}(x_L) L^k M_q}{k!} + \frac{1}{2} \right\rfloor, \quad k = 0, \ldots, r \tag{9}$$

and M_q is a positive integer which controls the level of quantization. The following lemma quantifies how well g is approximated by $\mathcal{P}^s_{I,M_q}(g)$ in the interval I.

Lemma 1 *Let $g \in \mathcal{R}_{\boldsymbol{b},n}$ and $p_q = \mathcal{P}^s_{I,M_q}(g)$ be its symmetric approximator. Their difference is bounded by*

$$\sup_{x \in I} \left|p_q(x) - g(x)\right| \le \frac{r+1}{2M_q} + \frac{b_{r+1}}{2(2n)^r} \lambda(I)^{r+1}. \tag{10}$$

In the following, we consider intervals whose length increases with image width n as $\lambda(I) = O(n^\kappa)$ for some $\kappa \in [0,1)$. The second term in the error bound (10) results from the polynomial approximation of g. If the interval I is sufficiently short, namely $\kappa \in [0, \frac{r}{r+1})$, this term tends to zero with n. The first term, due to coefficient quantization (8), decreases as M_q is increased.

For the image strip I with columns $x_0, \ldots, x_{2L}$, we thus define its *ideal search space* as

$$\mathcal{S}^*_p(\boldsymbol{b}, I, M_q, n) = \{\mathcal{P}^s_{I,M_q}(g) \mid g \in \mathcal{R}_{\boldsymbol{b},n}\}. \tag{11}$$

Analyzing the exact size of the set (11) and the precise vectors $(a_0, \ldots, a_r)$ that belong to it are difficult problems. Instead, we bound the individual coefficients.

Lemma 2 *Let $(a_0, \ldots, a_r)$ be the coefficients of the symmetric approximator (8) of a function $g \in \mathcal{R}_{\boldsymbol{b},n}$. Then, $a_0 \in [0, nM_q]$ and the higher-order coefficients satisfy*

$$|a_k| \le \left\lfloor \frac{b_k \lambda(I)^k M_q}{2(2n)^{k-1}} + \frac{1}{2} \right\rfloor, \quad k = 1, \ldots, r. \tag{12}$$

We denote by $\mathcal{S}_p(\boldsymbol{b}, I, M_q, n)$ the set of quantized polynomials whose coefficients satisfy the conditions of Lemma 2 and name it the *polynomial search space*. By definition, it contains the ideal search space (11). We first quantify its size, as $n \to \infty$, see also Fig. 3a.

Lemma 3 *The size of the polynomial search space of a strip I of width $O(n^\kappa)$ scales with n as follows:*
Case (i): linear edges, $b_1 > 0$, and $b_k = 0$ for all $k \ge 2$,

$$\left|\mathcal{S}_p(\boldsymbol{b}, I, M_q, n)\right| = O(n^{1+\kappa}). \tag{13}$$

Case (ii): general curved edges, $b_k > 0$ for all $k = 1, \ldots, r$,

$$\left|\mathcal{S}_p(\boldsymbol{b}, I, M_q, n)\right| = \begin{cases} O(n^{1+\kappa}) & \kappa \in \left[0, \frac{1}{2}\right] \\ O(n^{3\kappa}) & \kappa \in \left(\frac{1}{2}, \frac{2}{3}\right] \end{cases} \tag{14}$$

Equations (14) and (13) show that up to $\kappa \le 1/2$, the search spaces of straight or curved edges have comparable size. For $\kappa \in (1/2, 2/3]$, the curvature of edges is noticeable, leading to a significantly larger search space. We stop κ at $2/3$ since the size of the search space is then $O(n^2)$, implying that even with a single operation to process each candidate curve, this would violate the sublinear constraint. Furthermore, with $\kappa < 2/3$, Lemma 2 implies that for sufficiently large n, $a_k = 0$, for all $k > 2$. In other words, the sublinear constraint means that asymptotically in n, regardless of the smoothness vector $\boldsymbol{b}$, the search space contains quadratic polynomials.

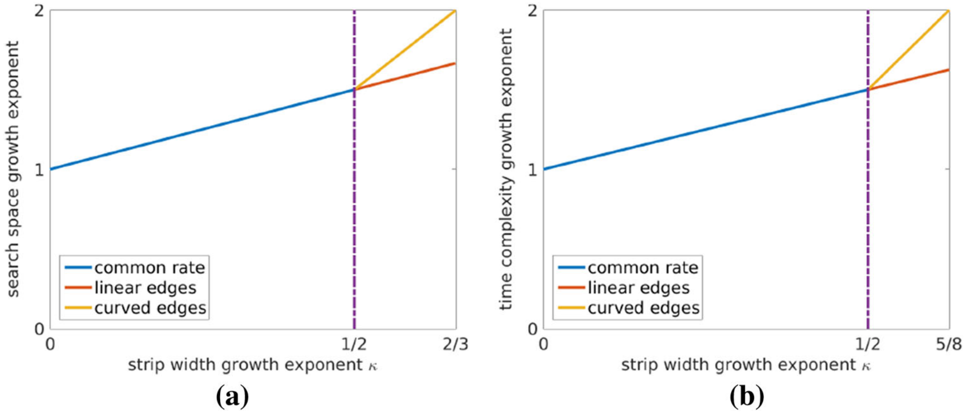


Fig. 3 Growth exponents of **a** the size of the polynomial search space. **b** The time complexity to compute all the edge responses associated to the polynomial search space

Piecewise linear interpolation From a computational viewpoint, a strip of width $L = O(\sqrt{n})$, namely $\kappa = 1/2$ is already problematic: by Lemma 3, it contains $O(n^{3/2})$ candidate curves, and by Eq. (6), each edge response, being a scaled sum of L individual pixel responses, requires $O(L)$ operations to compute. Hence, computing all edge responses yields $O(n^2)$ operations per strip, which contradicts the sublinear requirement.

To lower the computational cost, we further approximate each polynomial curve by its piecewise linear interpolant. This allows use of the line integral method [4], which calculates the edge responses of all linear candidate curves in a strip in an efficient recursive manner.

Specifically, consider an interval I of length $\lambda(I) = 2md$, for some integers m and d, divided into $2m$ segments each of length d. We approximate each quantized polynomial p_q by its piecewise linear function $\ell(x) = \mathcal{I}_m(p_q)(x)$, which interpolates p_q at the $2m+1$ equispaced points in the interval I. We denote the resulting set of piecewise linear functions as *the* search space,

$$\mathcal{S}(\boldsymbol{b}, I, M_q, m, n) = \{\mathcal{I}_m(p_q) \mid p_q \in \mathcal{S}_p(\boldsymbol{b}, I, M_q, n)\}.$$

To obtain the edge responses of all the candidate curves in this search space, we proceed in two steps. First, we compute all the linear edge responses in the $2m$ strips and store them in the memory. Then, for each candidate curve, we retrieve the responses of its $2m$ linear pieces and output their scaled sum as its edge response.

Our main result, stated in the theorem below, is that: (i) the error of this piecewise linear interpolation can be kept small; and (ii) computing all the edge responses can be done in sublinear time.

Theorem 1 *Let $\mathcal{S}$ be the search space of linearly interpolated symmetric approximators with spacing d, corresponding to a strip I of width $\lambda(I) = 2md$. Then, for any $g \in \mathcal{R}_{\boldsymbol{b},n}$, there is an $\ell \in \mathcal{S}$ such that inside I,*

$$\|\ell - g\|_\infty \leq E_1(\boldsymbol{b}, I, M_q, n) + E_2(\boldsymbol{b}, I, M_q, n, m) \qquad (15)$$

where the two error bounds E_1 and E_2 are given by

$$E_1 = \frac{r+1}{2M_q} + \frac{b_{r+1}}{n^r}\left(\frac{\lambda(I)}{2}\right)^{r+1} \qquad (16)$$

$$E_2 = \frac{1}{8M_q m^2}\sum_{k=2}^{r} k(k-1)\left\lfloor \frac{b_k \lambda(I)^k M_q}{2(2n)^{k-1}} + \frac{1}{2}\right\rfloor. \qquad (17)$$

For $d = O(n^{\kappa \wedge 1/2})$, the number of operations to compute the edge responses of all candidate curves in $\mathcal{S}$ is

$$\begin{cases} O(n^{1+\kappa}\log n), & \kappa \in \left[0, \frac{1}{2}\right] \\ O(n^{4\kappa - 1/2}), & \kappa \in \left(\frac{1}{2}, \frac{5}{8}\right) \end{cases}$$

For $\kappa \in [0, 5/8)$ and $d = O(n^{\kappa \wedge 1/2})$, the two error terms are bounded by a constant independent of n.

5 Sublinear Multiscale Curved Edge Detection

We now provide a more detailed discussion of our strip-based edge detection scheme. In Sect. 5.1, after a motivating example, we construct a multiscale search space for a strip and describe the accompanying detection algorithm, under the assumption that the strip contains a single step edge. In Theorem 2, we show that under certain conditions, with high probability, this algorithm accurately estimates the unknown edge.

Section 5.2 is devoted to the second step of the algorithm, namely tracking. We describe an iterative procedure which explores a small set of plausible extensions to the edges estimated so far, and based on the resulting edge responses, decides if and where to proceed. In Sect. 5.3, we show that the time complexity of the tracking procedure is asymptotically negligible compared to that of the detection step. Combining this with its minimal detectable edge saliency (Lemma

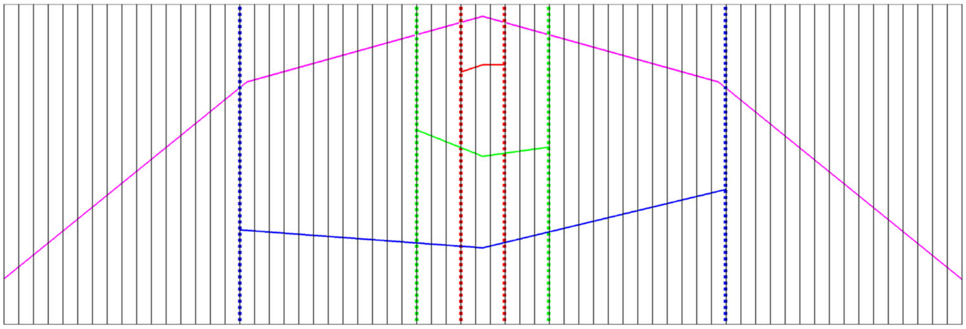


Fig. 4 An strip of $J = 6$ scales with $J_d = 4$. The columns confined by *red*, *green* and *blue dashed lines* correspond, respectively, to its sub-strips at scale 1, 3, and 5. In each of these scales, a candidate curve was drawn in the *same color*. A candidate curve at scale 6 has 4 linear pieces. Candidate curves at scale $s \le J_d + 1$ have only 2 linear pieces of length 2^{s-1} each (Color figure online)

5) yields the trade-off between computational cost and statistical accuracy of our algorithm. Finally, in Sect. 5.4, we consider several practical issues and required modifications, so that the resulting algorithm can successfully detect edges in real images.

5.1 Multiscale Strip-Based Edge Detection

The approach outlined in Sect. 4 was based on two components: (1) prior knowledge of edge smoothness; and (2) computing edge responses in a wide strip, to facilitate detection at low SNR settings.

In general, however, some of the image edges may be more wiggly than expected and violate the assumed smoothness condition. Then no curves in the search space can approximate these edges uniformly well across a wide strip. In this case, a large strip width is in fact *detrimental* to their detection.

For example, consider an L column strip containing a single wiggly edge g. With a mask width $\omega = 1$, the expected edge response of a candidate curve h is proportional to the number of pixels at which $\lfloor h \rfloor = \lfloor g \rfloor$,

$$|\mathbb{E}R(h)| = \frac{\mu_{\Gamma(g)}}{\sqrt{2}\sigma} \cdot \frac{|\{k \in \{1, \ldots, L\},\ \lfloor h(x_k) \rfloor = \lfloor g(x_k) \rfloor\}|}{\sqrt{L}}.$$

In contrast, the pixel responses in any column of the strip, precisely on the edge, have expected value

$$|\mathbb{E}R(x_k, \lfloor g(x_k) \rfloor)| = \frac{\mu_{\Gamma(g)}}{\sqrt{2}\sigma}, \quad k = 1, \ldots, L.$$

Thus, if for all $h \in \mathcal{S}$, the overlap with the wiggly edge g is smaller than $\sqrt{L}$ pixels, then the pixel responses are more discriminative than the longer edge responses.

The moral of this example is significant: A short candidate curve can be better suited for detecting a wiggly edge than a long one which approximates it poorly. Indeed if the strip is narrow, a misspecified b_2 has no effect on the resulting search space. Specifically, from Lemma 2 with $\lambda(I) < \sqrt{\frac{2n}{b_2 M_q}}$, it only contains linear candidate curves because $a_2 = 0$. Motivated by this observation, we develop a multiscale scheme to detect curved edges not only in the full strip, but also in its sub-strips of various widths, while retaining the sublinear computational efficiency.

Multiscale search space and detection procedure Consider a strip with $L = 1 + 2^J$ columns. We denote its middle column as the *scale* 0 *sub-strip*, while for $s \ge 1$, the *scale s sub-strip* is its $1 + 2^s$ middle columns. The strip *maximal scale* is J. Given a smoothness vector $\boldsymbol{b}$, we associate a search space to each of these $J+1$ nested sub-strips. The scale 0 search space is composed of the pixels of the middle column. The other search spaces are constructed as described in the previous section. Specifically, let $J_d < J$ be a positive integer. A candidate curve in the search space at scale $s \in \{1, \ldots, J_d\}$ has two linear pieces of length 2^{s-1} each, whereas for $s > J_d$, it has 2^{s-J_d} linear pieces of individual length 2^{J_d}. See Fig. 4 for an illustration. This design keeps the interpolation error under control by limiting the length of each linear segment to at most $d = 2^{J_d}$. As discussed previously, we choose J_d such that $2^{J_d} = O(n^{\kappa \wedge 1/2})$.

The union of the $J + 1$ search spaces is referred to as the *multiscale search space*. It is given by $\mathcal{T}_J := \cup_{s=0}^{J} \mathcal{S}(\boldsymbol{b}, I_s, M_q, m_s, n)$ where I_s is the interval of scale s, m_s is 1 for $s \le J_d$ and 2^{s-J_d-1} for $s > J_d$.

For analysis purposes, assume that the strip contains a single step edge. As outlined in Algorithm 2, the multiscale detection scheme works as follows: We first compute all the edge responses at scale 0. If a candidate curve *fires*, that is, the absolute value of its edge response exceeds a preset detection threshold, we stop the procedure and keep the candidate curve with the maximal absolute edge response as the estimated edge. Otherwise, we proceed to the next scale. The first scale at which some candidate curve fires is denoted s^*. If we reach

Algorithm 2 Multiscale Edge Detection

```
**Input:** a $(1 + 2^J)$-column strip with at most one edge.
**Parameter:** detection threshold $\tau_d$.
**for** scale $s = 0$ **to** $J$ **do**
    Calculate edge responses $R(\ell_s)$ of all candidate curves $\ell_s$ at scale $s$.
    **if** $\max_{\ell_s} |R(\ell_s)| > \tau_d$ **then**
        Output $\bar{\ell}_s = \operatorname{argmax}_{\ell_s} |R(\ell_s)|$ as the detected edge.
        Set $s^*$ to $s$.
        **return**
    **end if**
**end for**
Set $s^*$ to $+\infty$.
```

scale J and none of the scale J candidate curves fired, we conclude that there is no edge in the strip, or it is too weak to be detected, and set $s^* = +\infty$.

Detection threshold Since all the edge responses are Gaussian distributed with unit variance, by a standard union bound argument we have the following result.

Lemma 4 *Let* $\alpha \in (0, 1)$. *For any collection* $\mathcal{A}$ *of candidate curves, we have*

$$\mathbb{P}\left(\max_{h \in \mathcal{A}} |R(h) - \mathbb{E}R(h)| \leq \sqrt{2 \ln \frac{|\mathcal{A}|}{\alpha}}\right) \geq 1 - \alpha. \tag{18}$$

According to (18), in an edge-free strip, with probability at least $1 - \alpha$, all the edge responses are uniformly bounded in absolute value by $\sqrt{2 \ln(\alpha^{-1}|\mathcal{T}_J|)}$. However, if an edge is indeed present, we not only want to reliably decide that it exists, but also to accurately estimate it. Hence we use a slightly higher detection threshold

$$\tau_d = \Delta + \sqrt{2 \ln \frac{|\mathcal{T}_J|}{\alpha}} \tag{19}$$

with say $\Delta = 1$. By construction, $|\mathcal{T}_J|$ and the size of the search space at scale J are of the same order, which implies that τ_d grows at rate $O(\sqrt{\ln n})$.

Minimal detectable edge SNR Given the threshold (19), the following lemma characterizes the minimal edge SNR above which any dilated $\boldsymbol{b}$-regular edge can be detected with overwhelming probability.

Lemma 5 *In a strip of* $O(n^\kappa)$ *columns, with* $\kappa \in [0, 5/8)$, *the minimal detectable SNR of a dilated* $\boldsymbol{b}$*-regular edge decreases at rate* $O(n^{-\kappa/2}\sqrt{\ln n})$.

Asymptotic performance guarantee The selected threshold controls the number of false alarms in the background. But we also need to guarantee the candidate curves that do fire are well aligned with the true edge. We thus define the *covering ratio* as the proportion of a candidate curve covered by the signal tube of the edge.

Definition 2 Let $\{x_1, x_2, \ldots, x_L\}$ be L consecutive integers. Let h be a curve defined on the interval $[x_1, x_L]$. Its covering ratio with respect to a function $g \in \mathcal{R}_{\boldsymbol{b},n}$ is

$$\rho(g, h) = \frac{1}{L} \sum_{k=1}^{L} 1_{|h(x_k) - g(x_k)| \leq \omega - 1}.$$

To shorten the notation, we write the error terms (16) and (17) in Theorem 1 as $E_1(I)$ and $E_2(I, m)$, where $2m$ is the number of segments I is divided into. Clearly, $\max_{1 \leq s \leq J} E_1(I_s) + E_2(I_s, m_s)$ is small when n is large. Hence, we assume that for some integer $\gamma^* \leq \omega - 1$

$$\max_{1 \leq s \leq J} E_1(I_s) + E_2(I_s, m_s) \leq \gamma^* \tag{20}$$

Under this assumption, we have the following result.

Theorem 2 *Let* $(s^*, \bar{\ell}_{s^*})$ *be the output of Algorithm 2 with a fixed false alarm rate* α, *on a strip of width* $O(n^\kappa)$ *with* $\kappa \in [0, 5/8)$, *that contains a single edge* $g \in \mathcal{R}_{\boldsymbol{b},n}$. *Then, for any* $\epsilon > 0$, $\delta > \alpha$, *and* n *sufficiently large, with probability at least* $1 - \delta$, *either no candidate curve fired up to scale* J $(s^* = +\infty)$ *or the output curve* $\bar{\ell}_{s^*}$ *has a covering ratio at least* $\omega^{-1}(\omega - \gamma^*)(1 - \epsilon)$. *Formally, the following event has probability at least* $1 - \delta$

$$\left\{s^* = +\infty\right\} \cup \left\{s^* < +\infty,\ \rho(g, \bar{\ell}_{s^*}) > \frac{\omega - \gamma^*}{\omega}(1 - \epsilon)\right\}.$$

5.2 Edge Tracking

After detecting parts of edges in a few strips using Algorithm 2, we apply an iterative tracking procedure that extends them outside the strips. As detailed below, our tracking method relies on the assumed edge smoothness, to extend the detected edge only in a small window adjacent to its current estimated end location.

Tracking procedure For simplicity, we still assume that the image contains a single edge. Let $(s^*, \bar{\ell}_{s^*})$ be the output of Algorithm 2 on an image strip. If $s^* = \infty$, there is no edge to track. Otherwise, with some abuse of notation, let I_0 denote the interval corresponding to the scale s^*. To its right, define consecutive *tracking intervals* $I_i = [\bar{x}_i, \bar{x}_{i+1}]$ of length $\lambda(I_i) = L_{\text{track}},\ i \geq 1$.

As outlined in Algorithm 3, starting from I_1, we iteratively extend the edge estimate to the right. Denote by p_0 the polynomial curve on I_0 whose linear interpolant is $\bar{\ell}_{s^*}$. At the first iteration, given p_0 and the vector $\boldsymbol{b}$, we construct a set of polynomial candidate curves on I_1, referred to as p_0's *smooth extensions* $\mathcal{N}_1(p_0)$. From this set we select the curve p_1 with the largest edge response in absolute value. If $|R(p_1)|$ exceeds a preset tracking threshold, p_1 is accepted as a valid extension and tracking continues onto I_2 with p_1 taking up the role of p_0. Otherwise, we reject p_1 and stop the tracking.

The role of the tracking threshold is to avoid tracking non-existent edges. Specifically, let I_g denote the domain of definition of an edge g with $I_0 \subset I_g$. Then ideally, tracking should stop at $i^* = \min\{i \geq 1, I_g \cap I_i = \emptyset\}$.

Since the number of tracking intervals can potentially be $O(n)$, extreme noise variations of similar magnitude as the detection threshold τ_d can occur. Unlike in detection, in the tracking phase candidate curves are mostly concentrated around the true edge. As a single poorly selected extension could derail the whole tracking process, it is desirable to let good candidate curves have larger expected edge responses. By Eq. (7), we use longer intervals for tracking. Specifically, we set

$$L_{\text{track}} = \begin{cases} 2^t, & s^* = 0 \\ 2^t \lambda(I_0), & s^* > 0 \end{cases} \tag{21}$$

for some constant integer $t \geq 0$, called *scale offset*, In our experiments we used $t = 1$.

Algorithm 3 Rightward Edge Tracking

```
Input: a polynomial curve $p_0$ and $m$ tracking intervals.
Parameter: tracking false alarm rate.
for $i = 0$ to $m$ do
    Determine $p_i$'s set of smooth extensions $\mathcal{N}_{i+1}(p_i)$.
    Set the tracking threshold $\tau_{i+1}$.
    Calculate all the edge responses for $\mathcal{N}_{i+1}(p_i)$.
    Find $p_{i+1} = \mathrm{argmax}_{h \in \mathcal{N}_{i+1}(p_i)} |R(h)|$.
    if $|R(p_{i+1})| \leq \tau_{i+1}$ then
        Reject $p_{i+1}$.
        Abort.
    end if
end for
```

Edge responses and tracking threshold A good tracking threshold stops tracking neither too early nor too late. To help set such a threshold, in rightward tracking the edge response of a candidate curve on $[\bar{x}_i, \bar{x}_{i+1}]$ is based on its values at $\{\bar{x}_i + 1, \bar{x}_i + 2, \ldots, \bar{x}_{i+1}\}$, excluding the leftmost point. As noise in different columns is independent, this way, the calculated edge responses are independent of the previous ones leading up to it. Lemma 4 then implies the following result:

Lemma 6 *Let I_g denote the domain of an edge g and $i^* = \min\{i \geq 1, I_g \cap I_i = \emptyset\}$ the index of the first edge-free tracking interval. For $i \geq 1$, the tracking threshold*

$$\tau_i = \sqrt{2 \ln \frac{|\mathcal{N}_i(p_{i-1})|}{\alpha}} \tag{22}$$

ensures that the probability of the tracking procedure to continue beyond i^* intervals is at most α.

Smooth extensions For all $i \geq 0$, since we choose the extension p_{i+1} from the set $\mathcal{N}_{i+1}(p_i)$, ideally it should contain at least one candidate curve uniformly close to the edge on I_{i+1} with high probability. To construct such a set, we ask the following questions: (1) if p_i is a quantized polynomial approximator of the edge g on I_i, how can we use p_i to track g on I_{i+1}? (2) what if this assumption of p_i does not hold?

For simplicity, consider the problem of constructing $\mathcal{N}_1(p_0)$ when I_0 is degenerate, namely $\bar{x}_0 = \bar{x}_1$. This situation arises when some pixel fires in the middle column of the detection strip and p_0 is reduced to a quantized constant. To answer the first question, assume that we know the quantized location of the edge at $\bar{x}_1$

$$\frac{a_0'}{M_q}, \quad a_0' = \left\lfloor g(\bar{x}_1) M_q + \frac{1}{2} \right\rfloor \tag{23}$$

Our goal is to specify a set of quantized polynomials which contains the following approximator of g on I_1

$$\sum_{k=0}^{r} \frac{a_k}{M_q} \left(\frac{\cdot - \bar{x}_1}{\lambda(I_1)} \right)^k \text{ with } a_k = \left\lfloor \frac{g^{(k)}(\bar{x}_1) \lambda(I_1)^k M_q}{k!} + \frac{1}{2} \right\rfloor . \tag{24}$$

This is not the symmetric approximator defined in (8) because the corresponding Taylor series of g is expanded about the left end point of the interval I_1. We call this the *asymmetric approximator* of g on I_1 and denote it by $\mathcal{P}^a_{I_1, M_q}(g)$. This polynomial representation is more useful than its symmetric counterpart because for it to pass through $(\bar{x}_1, a_0'/M_q)$, we only need to set $a_0 = a_0'$. Furthermore, since g is $\boldsymbol{b}$-regular, by Lemma 2

$$|a_k| \leq \left\lfloor \frac{b_k \lambda(I_1)^k M_q}{n^{k-1}} + \frac{1}{2} \right\rfloor, \quad k = 1, \ldots r. \tag{25}$$

Hence, $\mathcal{P}^a_{I_1, M_q}(g)$ belongs to the set of asymmetric polynomials on I_1 whose coefficients satisfy Eqs. (25) and $a_0 = a_0'$, which we call the *oracle extensions* of g on I_1.

In practice, we do not know a_0'. However, Theorem 2 shows that the estimate p_0 approximates the edge well over its domain of definition. In particular, in the present degenerate case, with high probability the pixel $(\bar{x}_1, \lfloor p_0 \rfloor)$ belongs to the signal tube, and thus satisfies $|\lfloor g(\bar{x}_1) \rfloor - \lfloor p_0 \rfloor| \leq \omega - 1$. Hence, with high probability,

$$a_0 \in \bigcup_{|c - \lfloor p_0 \rfloor| \leq \omega - 1} [c M_q, (c+1) M_q] \tag{26}$$

since by Eq. (23), $\lfloor g(\bar{x}_1) \rfloor M_q \leq a_0' \leq \lfloor g(\bar{x}_1) \rfloor M_q + M_q$.

Hence, we choose $\mathcal{N}_1(p_0)$ as the set of asymmetric polynomials whose coefficients satisfy Eqs. (25) and (26). In simple words, $\mathcal{N}_1(p_0)$ is the union of several vertically shifted oracle extensions.

Next, consider the case where the detected candidate curve p_0 is non-degenerate, namely $\lambda(I_0) > 0$. Our goal is still to construct a set $\mathcal{N}_1(p_0)$ which contains $\mathcal{P}^a_{I_1,M_q}(g)$ with high probability. First assume analogously that we know the symmetric approximator

$$\mathcal{P}^s_{I_0,M_q}(g) = \sum_{k=0}^{r} \frac{a'_k}{M_q} \left(\frac{\cdot - \frac{\bar{x}_0+\bar{x}_1}{2}}{\lambda(I_0)/2} \right)^k$$

with

$$a'_k = \left\lfloor \frac{g^{(k)}(\bar{x}_1)\lambda(I_0)^k M_q}{2^k k!} + \frac{1}{2} \right\rfloor, \quad k = 0, \ldots, r.$$

Since $p_q = \mathcal{P}^s_{I_0,M_q}(g)$ does not uniquely characterize a curved edge g in general, we define the polynomials

$$\mathcal{E}_1\left(p_q\right) = \left\{ \mathcal{P}^a_{I_1,M_q}(g'), g' \in \mathcal{R}_{\boldsymbol{b},n} \text{ and } \mathcal{P}^s_{I_0,M_q}(g') = p_q \right\}$$

each of which may serve as a valid extension. To describe them, we observe that $\mathcal{P}^s_{I_0,M_q}(g)$, when restricted to $[\frac{\bar{x}_0+\bar{x}_1}{2}, \bar{x}_1]$, namely the right half of the interval I_0, is the asymmetric approximator of the edge g on that interval of length $\lambda(I_0)/2$. As a result, $\mathcal{P}^s_{I_0,M_q}(g)$ and $\mathcal{P}^a_{I_1,M_q}(g)$ can be related in terms of their coefficients thanks to the following lemma.

Lemma 7 *Let $p_q^{(0)}$ and $p_q^{(1)}$ be the asymmetric approximators of a function $g \in \mathcal{R}_{\boldsymbol{b},n}$ on the adjacent intervals $I'_0 = [x'_0, x'_1]$ and $I'_1 = [x'_1, x'_2]$*

$$p_q^{(i)}(x) = \sum_{k=0}^{r} \frac{a_k^{(i)}}{M_q} \left(\frac{x - x'_i}{\lambda(I'_i)} \right)^k, \quad i = 0, 1.$$

Then, the constant term of $p_q^{(1)}$ satisfies

$$\left| \frac{a_0^{(1)}}{M_q} - p_q^{(0)}(x'_1) \right| \leq \frac{r+2}{2M_q} + \frac{b_{r+1}}{n^r} \lambda(I'_0)^{r+1} \tag{27}$$

and its higher-order coefficients satisfy for $k = 1, \ldots, r$

$$\left| a_k^{(1)} - a_k^{(0)} \frac{\lambda(I'_1)^k}{\lambda(I'_0)^k} \right| \leq \frac{1}{2} \left(1 + \frac{\lambda(I'_1)^k}{\lambda(I'_0)^k} \right) + Q \tag{28}$$

where $Q = \frac{M_q \lambda(I'_1)^k}{n^{k-1}} \min(2b_k, (k+1)b_{k+1} \frac{\lambda(I'_0)}{n})$.

The scaling factor $\frac{\lambda(I'_1)}{\lambda(I'_0)}$ relates the two adjacent asymmetric approximators of different lengths. For illustrative purposes, consider the implications of this lemma when the edge g is assumed linear and the scaling factor is 1, namely $\lambda(I'_0) = \lambda(I'_1)$. Here, Eq. (28) simplifies to $|a_1^{(1)} - a_1^{(0)}| \leq 1$ since $b_2 = 0$. Hence, given $a_1^{(0)}$, the number of possible values for $a_1^{(1)}$ is 3, a substantial reduction from $1+2b_1 M_q \lambda(I'_1)$ as an exhaustive approach would imply. Similar to the degenerate case, additional computational saving is obtained thanks to the reduced uncertainty on $a_0^{(1)}$ as shown by Eq. (27).

Given $p_q^{(0)}$ on I'_0, the set of asymmetric polynomials $p_q^{(1)}$ whose coefficients satisfy Eq. (27) and (28) are referred to as its oracle extensions on I'_1. By this lemma, we then construct the oracle extensions of the symmetric approximator $p_q = \mathcal{P}^s_{I_0,M_q}(g)$ that contain $\mathcal{E}_1(p_q)$.

However, $\mathcal{P}^s_{I_0,M_q}(g)$ is not known in practice because the detected edge p_0 is only guaranteed to be high covering asymptotically (Theorem 2). Assume $\rho(g, p_0) \geq \rho$ for some $\rho > 0$. Then $\mathcal{P}^s_{I_0,M_q}(g)$ must belong to

$$\mathcal{B}_\rho(p_0) = \{\mathcal{P}^s_{I_0,M_q}(g'), \quad g' \in \mathcal{R}_{\boldsymbol{b},n} \text{ with } \ \rho(g', p_0) \geq \rho\}.$$

Its rationale echoes that of Eq. (26). Consequently, we obtain the *smooth extensions*

$$\mathcal{N}_1(p_0) = \bigcup_{p_q \in \mathcal{B}_\rho(p_0)} \mathcal{E}_1\left(p_q\right).$$

As it is difficult to derive a tight superset for $\mathcal{B}_\rho(p_0)$, we heuristically replace it by a small set of vertically shifted p_0, for instance, $\{p_0 - 1, p_0, p_0 + 1\}$.

5.3 Sublinear Complexity

Tracking has a lower complexity than detection. Specifically, when $b_k > 0$, $k = 1, \ldots, r$, by definition (21), the length of a tracking interval L_{track} is at most $O(n^\kappa)$, and all the intervals I_i have the same length except for $i = 0$. Then Lemma 7 implies

$$|\mathcal{N}_{i+1}(p_i)| \leq C(\boldsymbol{b}, M_q) \cdot n^{(2\kappa-1)\vee 0}, \quad i \geq 0.$$

With at most $n/\lambda(I_i) = O(n^{1-\kappa})$ tracking intervals, the algorithm goes over at most $O(n^{(1-\kappa)\vee\kappa})$ smooth extensions. As it takes $O(n^\kappa)$ operations to calculate the edge response of a smooth extension, tracking takes at most $O(n^{1\vee 2\kappa})$ operations, which is asymptotically small compared to that of detection (Theorem 1). In other words, the overall runtime of the algorithm is determined by its detection step.

The same conclusion holds for straight edges, where $b_1 > 0$ and $b_k = 0$ for $k \geq 2$. Here, the time complexity for tracking is $O(n)$, again smaller compared to $O(n^{1+\kappa}\log n)$, $\kappa \in [0, 1)$ needed for detection.

5.4 Multiple Edges and Other Practical Issues

Several modifications are needed before our algorithm can be applied to natural images.

Multiple edges and nonmaximal suppression. Real images typically contain multiple edges of different contrasts. The detection algorithm thus cannot stop at the first scale where a candidate curve fired. Instead it should go over all the nested sub-strips. Our analysis shows that if a dilated $\boldsymbol{b}$-regular edge is detected at scale s, it is likely to be detected at scale $s+1$, too. To minimize the number of times tracking is invoked, we should keep only one candidate curve for each true edge in the detection strip. To this end, we identify each detected curve in the strip with the pixel it passes through in the middle column and group the detected edges by their position and the sign of their edge responses using the connected component algorithm. For each resulting group, we then select the shortest candidate curve with the strongest edge response as its representative.

To detect edges that do not span the whole image width, we extract from the image several equidistant vertical strips, and then track the detected edges rightward till we reach an adjacent strip or the image border. Leftward tracking is done in a similar manner. To handle those nearly vertical edges, we transpose the image and repeat the same procedure. This amounts to extracting horizontal detection strips from the original image and then tracking upward/downward.

Consistency test. Edge response, though informative, cannot be used as the only criterion for judging the saliency of a candidate curve. In particular, it cannot rule out broken edges. As detailed in Appendix 1, we use a χ^2 consistency test to check if the pixel responses of a candidate curve are sufficiently uniform. This test not only reduces the number of spurious edges produced in the detection stage, but also enhances the robustness of the tracking procedure by eliminating candidate curves that deviate significantly from the true edge.

Post-processing. Ideally, tracking should stop at iteration $i^* = \min\{i \geq 1, I_g \cap I_i \neq \emptyset\}$. In practice, the last tracking interval may partially extend beyond the end of the edge, $I_{i^*} \backslash I_g \neq \emptyset$. Consequently, the last extension needs to be trimmed so the final output accurately reflects the true extent of the edges. Appendix 2 describes an endpoint location procedure for this purpose.

The actual detection and tracking procedure are summarized in Algorithms 4 and 5.

Algorithm 4 Multiscale Edge Detection

```
Input: a noisy $(1 + 2^J)$-column strip.
Output: a list of detected polynomial edges in the strip.
Parameter: detection false positive rate $\alpha_d$.
Calculate the edge responses at all the scales $s = 0, \dots, J$.
Set the detection threshold $\tau_d$ according to Eq. (18).
Remove the candidate curves whose edge responses are smaller than $\tau_d$ in absolute value.
Group the remaining candidates according to their position and the sign of their edge responses.
For each group, pick the shortest candidate curve with the largest response in absolute value.
Subject these candidate curves to the consistency test and output those passing the test.
```

Algorithm 5 Rightward Edge Tracking

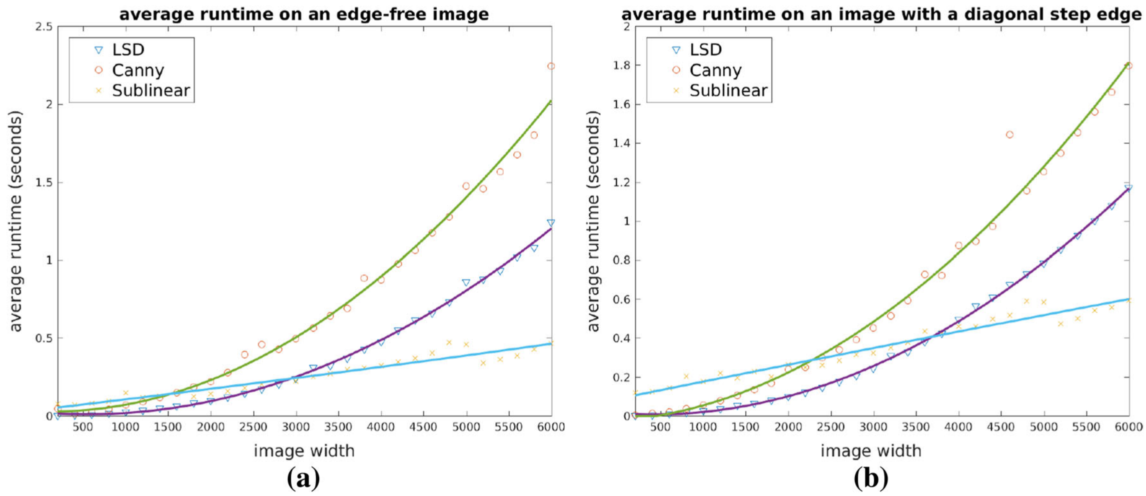
```
Input: a list of polynomial candidate curves and $m$ tracking intervals.

Output: a list of smooth extension sequences.
Parameter: tracking false positive rate $\alpha_t$.
for each detected polynomial candidate curve $p_0$ do
  for $i = 0$ to $m$ do
    Determine $p_i$'s smooth extensions $\mathcal{N}_{i+1}(p_i)$.
    Set the tracking threshold $\tau_{i+1}$ by Eq. (22).
    for each candidate curve in $\mathcal{N}_{i+1}(p_i)$ do
      Subject it to the consistency test.
       Calculate its edge response.
       Remove it from $\mathcal{N}_{i+1}(p_i)$ if (1) it fails the test or (2) its edge response has different sign from $p_0$.
    end for
    Find $p_{i+1} = \operatorname{argmax}_{h \in \mathcal{N}_{i+1}(p_i)} |R(h)|$.
    if $|R(p_{i+1})| \leq \tau_{i+1}$ then
      Reject $p_{i+1}$.
      Abort.
    else
      Trim $p_i$ using the endpoint location procedure.
    end if
  end for
end for
```

6 Experiments

We demonstrate the performance of our algorithm on both synthetic and natural images. In all experiments, the detection and tracking false alarm rates were set to 10^{-5} and 10^{-4}, respectively. Also fixed were the interpolation spacing $d = 8$, the quantization parameter $M_q = 1$, the scale offset $t = 1$, and the smoothness vector $\boldsymbol{b} = (3, 10, 10)$, which means that all candidate curves were quadratic polynomials.

First, we compared the runtime of the detection and tracking steps of our sublinear algorithm on two sequences of synthetic square images of increasing sizes, corrupted by additive Gaussian noise with $\sigma = 100$. The first sequence of images contained only noise. Hence, with high probability the detection algorithm found no edges and the tracking step was not invoked. In the second sequence, the images contained one step edge along their diagonal with edge contrast



(a) **(b)**

Fig. 5 Runtime comparison **a** on pure noise images ($\sigma = 100$); **b** on noisy images containing a diagonal step edge (SNR $= 2$). In agreement with our theoretical analysis that tracking is significantly faster than detection, the empirical runtime of the sublinear algorithm is only slightly higher on the images with an edge than on the noise-free images

200 and SNR $= 2$, separating two constant-valued triangles. In these images, both detection and tracking would run.

Our algorithm examined a single strip of scale $J = 5$ from the middle of each input image with mask width $\omega = 3$ and tracked detected edges to both the left and right sides of the strip. For comparison we also ran the line segment detector (LSD) [29,30] and the Canny edge detector. To make them yield good results on noisy images, their default parameters were modified to $[\text{low}, \text{high}] = [0.32, 0.62]$ (Canny) and $[S, \Sigma] = [0.17, 0.8]$ (LSD). All algorithms ran on a single CPU.

The runtime results in Fig. 5, averaged over 10 independent runs, show that our algorithm is indeed sublinear in the image size while both Canny and LSD are linear. We also see that, consistent with our theoretical analysis, tracking is faster than detection.

Next we tested our algorithm on two 1000×1000 synthetic images with edge SNR equal to 1 and 2 (Fig. 6). One horizontal and one vertical image strip were used for edge detection. We estimated the noise level from these two strips by the median absolute deviation estimator [25] and ran the algorithm with $\omega = 7$ and $J = 6$. The results show that under a weak SNR, our sublinear algorithm, thanks to its ability to account for curved edges, was less prone to false alarm than Canny and more robust compared with LSD.

To apply our algorithm to natural images, we need to estimate the noise level. However, when only contrasted edges are of interest, we may simply provide the algorithm with a high noise level without actually estimating it. This leads to edge outputs with starker contrast. We used this method in Figs. 7, 8 and 9. In these images, we extracted eight detection strips (four equidistant horizontal strips and four vertical ones) with $J = 5$. Input noise level σ was selected by the following rule of thumb: in smooth and poorly contrasted images, such as Figs. 8a and 9a, we set $\sigma = 3$. Otherwise, we set $\sigma = 20$ to focus on contrasted edges. The mask width ω was adjusted in a small range, between 3 and 7.

7 Discussion and Future Work

We have shown how multiple smooth curved edges can be tracked in noisy images in sublinear time, assuming the edges intersect one of the detection strips. An initial multiscale search in a detection strip triggers a detection. The search involves tests on a finite family of polynomial curves of increasing length, where the larger the length the lower the contrast at which a test reliably *fires*. A detection initializes a tracking step that extends the initial detection along the edge. Due to prior assumptions on the smoothness of the edges, tracking is very efficient since each edge segment can only be extended in a limited number of ways. For strong edges, a test fires early in the multiscale search and immediately triggers the tracking algorithm. Fainter edges are detected later, in wider sub-strips.

Local edge detection algorithms that do not make prior assumptions on the edges necessarily require at least linear complexity in the number of pixels. Depending on their parameter settings, they may either yield multiple false alarms or miss many of the true edges. In contrast, our method makes use of the assumption of smoothness to detect and track even low SNR edges in sublinear time, with virtually no false alarms.

We provided a theoretical analysis of the computational load of the detection and tracking parts of the algorithm. For the detection part we also carried out a statistical analysis

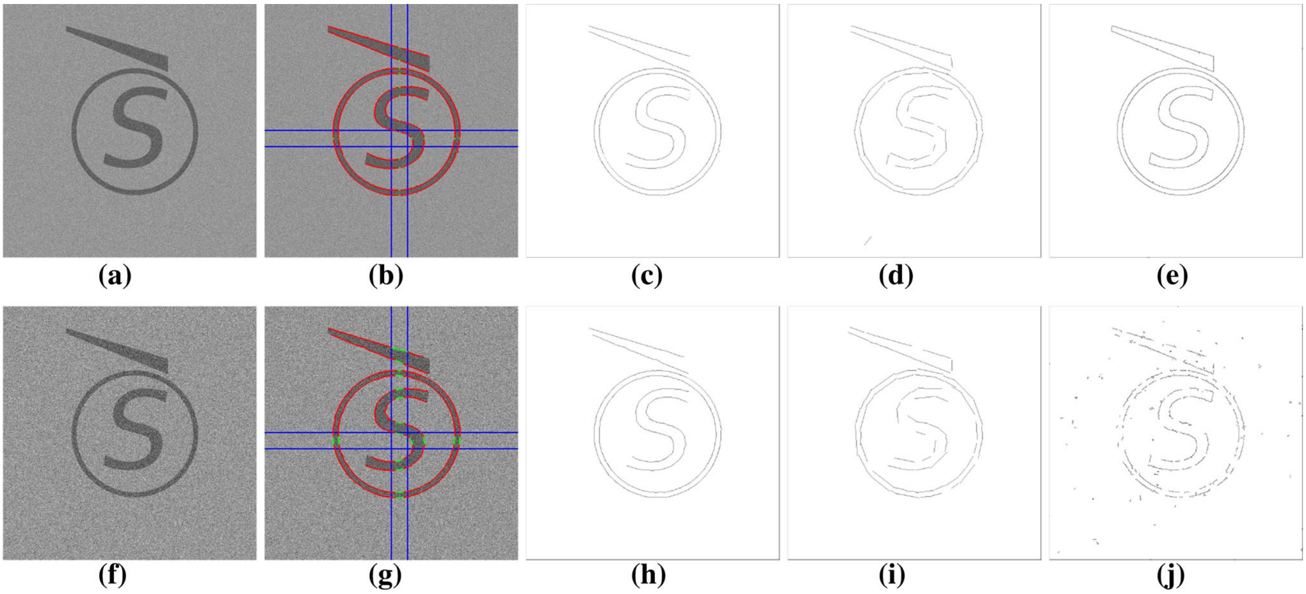
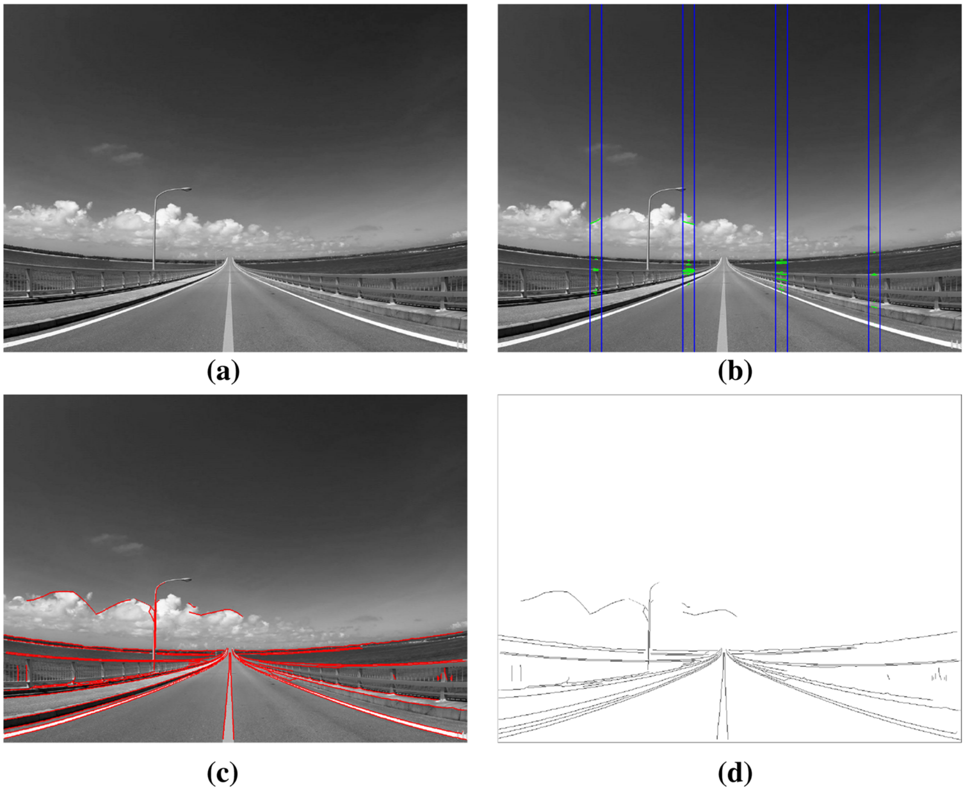


Fig. 6 Curved edge detection on 1000 × 1000 synthetic noisy images. *Top* and *bottom row* shows respectively detected edges by our sublinear algorithm from two strips, LSD and Canny in **a** and **f**. **a** Observed (SNR $= 2$). **b** Sublinear. **c** Sublinear sketch. **d** LSD sketch. **e** Canny sketch. **f** Observed (SNR $= 1$). **g** sublinear. **h** Sublinear sketch. **i** LSD sketch. **j** Canny sketch

Fig. 7 Curved edge detection in a natural image ($\omega = 3, \sigma = 20$). **a** Original image (1280 × 960). **b** Detected edges in four equidistant strips of equal width ($J = 5$). Depending on their contrast, the edges were detected at different scales. **c** Edges tracked from the detected segments in **d** and in four equidistant horizontal detection strips (not shown). **e** final sketch



showing that with high probability the only curves that fire have a significant overlap with the true edge. We also provided the minimal contrast at which an edge will be detected.

In terms of future research, we believe it is possible to extend the probabilistic analysis to the tracking part of the algorithm to control the probability of tracking the entire edge reliably. Furthermore, with more refined analysis, we believe it will be possible to provide analytic upper and lower bounds on the coefficients of the smooth extensions that are *guaranteed* to contain highly covering curves in the next tracking

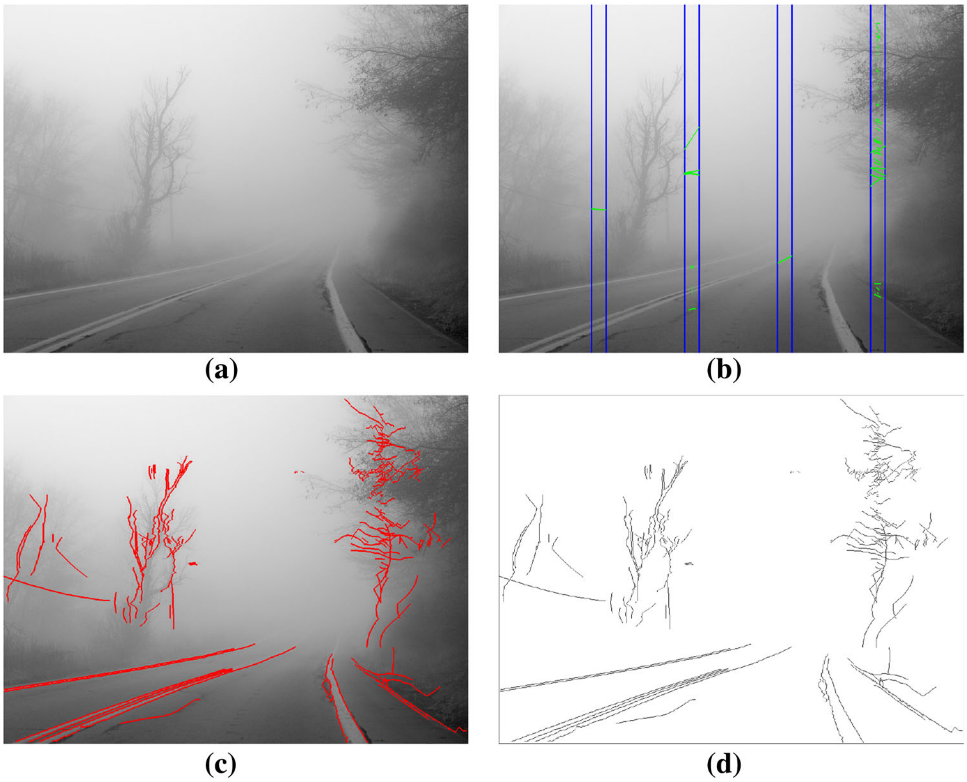


Fig. 8 Curved edge detection in a natural image ($\omega = 3, \sigma = 3$). **a** Original image (1024×768). **b** Detected edges in four equidistant strips of equal width ($J = 5$). From the detection of the lane markings in different strips, we see again that the more contrasted an edge is, the easier it is to be detected, that is, at a lower scale. **c** Edges tracked from the detected segments in **d** and from four equidistant horizontal strips (not shown). **d** final sketch

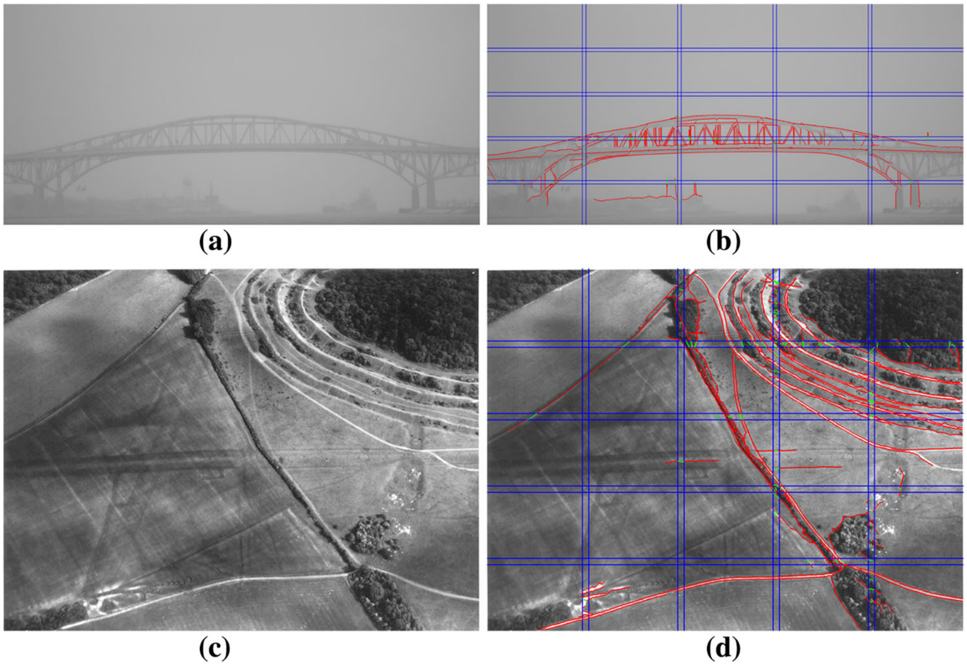


Fig. 9 Edge detection in the two images **a** with 4320×2000 pixels and **c** (2416×1836), respectively. All the detection strips had scale $J = 5$. For image **a**, the mask width and the noise level were set to $\omega = 4, \sigma = 3$ while for image **c**, $\omega = 5, \sigma = 20$. **a** Original. **b** Processed. **c** Original. **d** Processed

interval. Currently, we use a heuristic to define this set of coefficients.

An additional aspect of our method is the organization of the computation in an *easy to hard* manner. Easy high-contrast edges are detected early and tracked immediately. Weaker edges come later. The computational load of the *easy* part is much smaller than the total. In the future, we want to further extend this principle to organize the *type* of edges that are tracked. In other words, the computation would be organized so that in a natural manner high-contrast straight or very slowly varying edges are detected first, then in some order the more wiggly and low-contrast edges are brought into the computation. The advantage of this *easy to hard* organization of the computation is that if interrupted early, it may still provide much valuable information about the image. Furthermore, it is also possible to stop it, once relevant information, depending on the particular application, has already been found.

The need to keep the computation sublinear using the line integral algorithm led to the division of the interval into $2m$ subintervals requiring $O(n^{4\kappa-1/2})$ operations when $\kappa \in (1/2, 5/8)$ (see proof of Theorem 1). We believe there may be a recursive computation of the sums over all curves on the domain, using dyadic decompositions that could be achieved in $O(n^{3\kappa}\log n)$ operations—i.e., the number of curves up to a log factor—at the cost of a larger approximation width. This would then allow sublinear edge detection over the entire range $\kappa \in [0, 2/3)$, instead of the current upper bound of $5/8$.

Finally, a study of fundamental lower bounds on the number of operations needed to detect curved edges of various possible lengths, contrasts and smoothness parameters, and minimax lower bounds on the detectable edge SNR under computational constraints are challenging topics for further research.

Acknowledgements Funding was provided by Division of Mathematical Sciences (Grant No. 0706816).

Appendix 1: Consistency Test

Let $(y_k)_{1\le k\le L}$ be a Gaussian random vector

$$y_k = \mu_k + \xi_k, \quad k = 1, \dots, L$$

where the noise terms $(\xi_k)_{1\le k\le L}$ are i.i.d. of zero mean and known variance σ^2. Under the null hypothesis, its mean vector satisfies $\mu_1 = \mu_2 = \cdots = \mu_L$. Then the test statistic

$$T = \frac{1}{\sigma^2}\sum_{k=1}^{L}\left(y_k - \frac{1}{L}\sum_{i=1}^{L} y_i\right)^2$$

follows the Chi-squared distribution with $L-1$ degrees of freedom. We reject the null hypothesis when

$$T \ge L - 1 + 2\ln\delta^{-1} + 2\sqrt{(L-1)\ln\delta^{-1}}.$$

The false positive rate of this test is at most δ by the tail bound [18]

$$\forall t > 0, \ \ \mathbb{P}\left(\frac{T-(L-1)}{L-1} \ge 2t + 2\sqrt{t}\right) \le e^{-(L-1)t}.$$

Appendix 2: Endpoint Location

The problem of locating the endpoint t of a candidate edge given its values on an interval of length L can be formulated as follows: let $(y_k)_{1\le k\le L}$ be a Gaussian random vector

$$y_k = \mu \cdot 1_{k\le t} + \xi_k, \quad k = 1, \dots, L$$

where the noise terms $(\xi_k)_{1\le k\le L}$ are i.i.d., with zero mean and known variance σ^2, but the endpoint t and the contrast $\mu \ne 0$ are unknown. Their maximum likelihood estimates are

$$\begin{aligned}\max_{\mu,t}\log p(y_1,\dots,y_L) &= \min_{\mu,t}\left(\sum_{k=1}^{t}(y_k-\mu)^2 + \sum_{k=t+1}^{L} y_k^2\right)\\ &= \min_{\mu,t}\left(\mu^2 t - 2\mu\sum_{k=1}^{t} y_k\right).\end{aligned}$$

In particular, the estimated end point is

$$t^* = \operatorname*{argmax}_{1\le t\le L}\frac{\left(\sum_{k=1}^{t} y_k\right)^2}{t}.$$

Appendix 3: Proofs

Proof of Lemma 1 Let $p(x)$ be the degree r Taylor expansion of g around the middle point x_L of $I = [x_0, x_{2L}]$,

$$p(x) = \sum_{k=0}^{r}\frac{g^{(k)}(x_L)}{k!}(x - x_L)^k.$$

Since $L = \lambda(I)/2$, its approximation error satisfies

$$\sup_{x\in I}|p(x) - g(x)| \le \frac{b_{r+1}}{n^r}L^{r+1}.$$

It follows from the coefficient quantization formula (8)

$$\left|\frac{a_k}{M_q L^k} - \frac{g^{(k)}(x_L)}{k!}\right| \le \frac{1}{2M_q L^k}, \quad k = 0, \dots, r.$$

Hence, writing $p_q = \mathcal{P}^s_{I,M_q}(g)$, we find

$$\sup_{x\in I}|p(x) - p_q(x)| \le \frac{r+1}{2M_q}.$$

An application of the triangle inequality concludes the proof. □

Proof of Lemma 2 By definition, for each $g \in \mathcal{R}_{\boldsymbol{b},n}$, there is a $\boldsymbol{b}$-regular function f such that $g(x) = nf(x/n)$. According to (8), the coefficients of its symmetric approximator are

$$a_k = \left\lfloor \frac{f^{(k)}(c/n)\lambda(I)^k M_q}{k!2(2n)^{k-1}} + \frac{1}{2} \right\rfloor, \quad k = 1, \dots, r.$$

Since $|\lfloor z + \frac{1}{2} \rfloor| \leq |z| + \frac{1}{2}$ holds for all $z \in \mathbb{R}$, we deduce

$$|a_k| \leq \left| \frac{f^{(k)}(c/n)\lambda(I)^k M_q}{k!2(2n)^{k-1}} \right| + \frac{1}{2}, \quad k = 1, \dots, r.$$

Now, since f is $\boldsymbol{b}$-regular,

$$|a_k| \leq \frac{b_k \lambda(I)^k M_q}{2(2n)^{k-1}} + \frac{1}{2} \quad k = 1, \dots, r.$$

As the coefficients a_k are all integer valued, Eq. (12) follows. □

Proof of Lemma 3 First consider the case $b_1 > 0$ and $b_k = 0$ for $k \geq 2$. This corresponds to the assumption that image edges can be well approximated by straight segments. In this case, a_0 can have nM_q different values, whereas by Eq. (12), the coefficient a_1 can have approximately $b_1 M_q n^\kappa/2$ different values, and $a_k = 0$ for $k \geq 2$. Hence, in this case, the polynomial search space has $O(n^{1+\kappa})$ candidate curves which are all linear.

Next, consider the general case of curved smooth edges with $b_k > 0$, $k = 1, \dots, r$. Here, as above the coefficients a_0 and a_1 still have $O(n)$ and $O(n^\kappa)$ possible values. By Eq. (12), the higher-order coefficients have $O(n^{k\kappa}/n^{k-1})$ possibilities, of course provided that $1 + k(\kappa - 1) \geq 0$, otherwise they have a constant number of possible values, independent of n. Hence in the general case its size is $O\left(n^{\kappa+1+\sum_{k=2}^{r}(1+k(\kappa-1))_+}\right)$.

Let us analyze the behavior of this expression. For $\kappa \in [0, 1/2]$, for all $k \geq 2$, $1 + k(\kappa - 1) \leq 0$ and we obtain $O(n^{1+\kappa})$. For $\kappa \in [1/2, 2/3]$ the term with $k = 2$ also contributes and yields an overall size $O(n^{3\kappa})$. A value of $\kappa > 2/3$ yields a search space of size larger than $O(n^2)$ and thus not relevant for sublinear edge detection. Thus, Eq. (14) follows. □

Next, to prove Theorem 1 we shall make use of the following two auxiliary results:

Lemma 8 *Let $h(z)$ be a twice differentiable function defined on a closed interval $[z_1, z_2]$. Let $l(z)$ be its linear interpolant such that $l(z_i) = h(z_i)$, $i = 1, 2$. Then*

$$\sup_{z\in[z_1,z_2]} |h(z) - l(z)| \leq \frac{(z_2 - z_1)^2}{8} \sup_{z\in[z_1,z_2]} |h^{(2)}(z)|.$$

Proof Define for any $t \in (z_1, z_2)$

$$A_t(z) = h(z) - l(z) - \frac{h(t) - l(t)}{(t - z_1)(t - z_2)}(z - z_1)(z - z_2).$$

By construction, $A_t(z)$ is twice differentiable and has at least three roots $\{z_1, t, z_2\}$. Applying Rolle's theorem twice, there exists $\theta_t \in (z_1, z_2)$ such that $A''_t(\theta_t) = 0$ or equivalently

$$h''(\theta_t) - 2\frac{h(t) - l(t)}{(t - z_1)(t - z_2)} = 0.$$

It follows that

$$|h(t) - l(t)| \leq \frac{(z_2 - z_1)^2}{8} \sup_{z\in[z_1,z_2]} |h''(z)|.$$

□

Lemma 9 *Assume $\lambda(I) = 2md$ for some integers d and m. Let $(x'_k)_{0\leq k\leq 2m}$ be the $2m+1$ equally spaced integers with a spacing of d in the interval $I = [x'_0, x'_{2m}]$. For any function $g \in \mathcal{R}_{\boldsymbol{b},n}$, let p_q be its symmetric approximator of degree r on the same interval. Then the piecewise linear function ℓ that interpolates p_q at the grid points $\{x'_0, \dots, x'_{2m}\}$ satisfies*

$$\sup_{x\in I} |\ell(x) - p_q(x)| \leq \frac{1}{8M_q m^2} \sum_{k=2}^{r} k(k-1) \left\lfloor \frac{b_k \lambda(I)^k M_q}{2(2n)^{k-1}} + \frac{1}{2} \right\rfloor.$$

Proof of Lemma 9 Since the piecewise linear approximation $\ell(x)$ interpolates the polynomial p_q at $2m + 1$ equidistant points of the interval I, it follows from Lemma 8 that

$$\sup_{x\in I} |p_q(x) - \ell(x)| \leq \frac{d^2}{8} \sup_{x\in I} |p_q^{(2)}(x)| \tag{29}$$

Combining this with the definition of $p_q(x)$, Eq. (8) and the fact that $L = md$ gives that

$$\sup_{x\in I} |p_q(x) - \ell(x)| \leq \frac{1}{8M_q m^2} \sum_{k=2}^{r} |a_k| k(k-1).$$

The bound (12) on the coefficients a_k yields the lemma. □

Proof of Theorem 1 Let $g \in \mathcal{R}_{\boldsymbol{b},n}$, let p_q be its symmetric approximator and let $\ell(x)$ be its piecewise linear interpolant. By the triangle inequality in the interval I,

$$\|g - \ell\|_\infty \leq \|g - p_q\|_\infty + \|p_q - \ell\|_\infty$$

By Lemma 1, the first term on the right-hand side is bounded by (16), whereas by Lemma 9 the second term is bounded by (17). Hence, Eq. (15) of the theorem readily follows.

Next, let us consider how large can the spacing d be. On the one hand we would like d to be as large as possible, as this leads to significant gains in the runtime of the line integral algorithm [4]. In details, this method calculates the edge responses of all the linear candidate curves in a $d \times n$ strip, in $O(nd \log d)$ operations instead of $O(nd^2)$ by the naive method. While this recursion may induce an approximation error [4], at high noise levels its effect is negligible.

Now on the other hand, a larger spacing d yields a larger approximation error of the piecewise linear interpolation. For large values of n, the asymptotically dominant error term in (17) is the first summand with $k = 2$. This error term is approximately equal to $b_2 d^2/(4n)$. More precisely,

$$\left| \frac{1}{4M_q m^2} \left\lfloor \frac{b_2 \lambda(I)^2 M_q}{4n} + \frac{1}{2} \right\rfloor - \frac{b_2 d^2}{4n} \right| \le \frac{1}{8M_q m^2}.$$

For this term to be small, the spacing d should thus grow no faster than $O(\sqrt{n})$.

Armed with these results, we now consider the number of operations to calculate all the edge responses of linearly interpolated candidate curves in the polynomial search space. We first calculate all the linear responses in the $2m$ contiguous sub-strips of width d. Then, for each candidate curve we sum the corresponding responses to form the final output.

The first step costs $O(mnd \log d)$ operations. For the second step, consider first the case $\kappa \in (1/2, 2/3)$. According to (14), the number of polynomial candidate curves is $O(n^{3\kappa})$. Hence computing their edge responses requires $O(mn^{3\kappa})$ additional operations. Since $\lambda(I) = 2md$, the overall complexity is $O(mnd \log d + mn^{3\kappa}) = O(n^{1+\kappa} \log d + n^{4\kappa}/d)$. To minimize the complexity without sacrificing accuracy, we thus choose $d = O(\sqrt{n})$. At this value of d the first term is negligible, and the overall complexity is $O(n^{4\kappa - 1/2})$. For $\kappa \in [0, 1/2]$, a similar analysis shows that with $d = O(n^\kappa)$, the complexity is $O(n^{1+\kappa} \log n)$. As a result, the sublinear constraint implies an upper bound on the strip width $\kappa \in [0, 5/8)$. □

Proof of Lemma 5 As soon as the reference curve ℓ_J^* defined in Eq. (32) satisfies

$$|\mathbb{E}R(\ell_J^*)| \ge \tau_d + c_\beta \tag{30}$$

where c_β is the standard normal distribution's β-quantile, the probability of the event $\{s^* = +\infty\}$ of all candidate curves not firing is less than β. A sufficient condition for Eq. (30) to hold can be obtained using Eq. (7)

$$|\mathbb{E}R(\ell_J^*)| \ge \frac{\mu_{\Gamma(g)}(\omega - \gamma^*)\sqrt{L}}{\sigma\sqrt{2\omega}} \ge \tau_d + c_\beta$$

with $L = 2^J + 1$ the number of columns of the full strip. Rearranging this inequality, we find

$$\frac{\mu_{\Gamma(g)}}{\sigma} \ge \frac{\sqrt{2\omega}(\tau_d + c_\beta)}{(\omega - \gamma^*)\sqrt{L}}.$$

The strip width $L = O(n^\kappa)$ and the detection threshold $\tau_d = O(\sqrt{\ln n})$ thus imply the asymptotic minimal detectable edge SNR $= O(n^{-\kappa/2}\sqrt{\ln n})$. □

Proof of Theorem 2 If a pixel in the middle column fires, its signal must be nonzero on the high probability event

$$\left\{ \forall h \in \mathcal{T}_J,\ \ |R(h) - \mathbb{E}R(h)| \le \sqrt{2 \ln \frac{|\mathcal{T}_J|}{\alpha}} \right\}. \tag{31}$$

Consequently, its covering ratio is 1 with high probability.

Next, consider the case where $1 \le s^* < \infty$. The goal is to show that, asymptotically, the curve $\bar{\ell}_{s^*}$ enjoys a covering ratio close to $1 - \gamma^*/\omega$ with high probability. To this end, we first lower bound the covering ratio in terms of edge responses and then prove that when n is large, this lower bound approaches $1 - \gamma^*/\omega$ with high probability.

First, we define the following $J + 1$ *reference curves*

$$\ell_s^* = \operatorname*{argmax}_{\rho(g,\ell_s)=1} |\mathbb{E}R(\ell_s)|, \qquad s = 0, 1, \ldots, J. \tag{32}$$

Their existence is guaranteed by Assumption (20). Let L_s be the number of columns in the scale s sub-strip. Eq. (7) implies

$$|\mathbb{E}R(\ell_s^*)|\sigma\sqrt{2\omega L_s} \ge (\omega - \gamma^*)\mu_{\Gamma(g)} L_s. \tag{33}$$

By definition, any scale s candidate curve ℓ_s also satisfies

$$|\mathbb{E}R(\ell_s)|\sigma\sqrt{2\omega L_s} \le \omega\mu_{\Gamma(g)}\rho(g, \ell_s) L_s \tag{34}$$

Combining (33) and (34) gives the following deterministic lower bound on the covering ratio

$$\rho(g, \ell_s) \ge \frac{\omega - \gamma^*}{\omega} \left| \frac{\mathbb{E}R(\ell_s)}{\mathbb{E}R(\ell_s^*)} \right|, \quad s = 0, \ldots, J. \tag{35}$$

Two straightforward triangle inequalities then lead to

$$\rho(g, \ell_s) \ge \frac{\omega - \gamma^*}{\omega} \frac{|R(\ell_s)| - |R(\ell_s) - \mathbb{E}R(\ell_s)|}{|R(\ell_s^*)| + |R(\ell_s^*) - \mathbb{E}R(\ell_s^*)|}$$

As the relation $|R(\bar{\ell}_s)| \ge |R(\ell_s^*)|$ always holds, we obtain

$$\rho(g, \bar{\ell}_s) \ge \frac{\omega - \gamma^*}{\omega} \frac{|R(\bar{\ell}_s)| - |R(\bar{\ell}_s) - \mathbb{E}R(\bar{\ell}_s)|}{|R(\bar{\ell}_s)| + |R(\ell_s^*) - \mathbb{E}R(\ell_s^*)|}$$

which then implies

$$\rho(g, \bar{\ell}_s) \geq \frac{\omega - \gamma^*}{\omega}\left(1 - \frac{v_s^* + \bar{v}_s}{|R(\bar{\ell}_s)|}\right),$$

where $v_s^* := |R(\ell_s^*) - \mathbb{E}R(\ell_s^*)|$ and $\bar{v}_s := |R(\bar{\ell}_s) - \mathbb{E}R(\bar{\ell}_s)|$.

To get the desired result, it suffices to show that when n is large, on a high probability event restricted to $\{s^* < +\infty\}$, both $v_{s^*}^*$ and $\bar{v}_{s^*}$ are small compared to $|R(\bar{\ell}_{s^*})|$.

Though unknown, the reference curves ℓ_s^* are deterministic. Hence, it follows from Lemma 4 that $\max_{0\leq s\leq J} v_s^*$ is $O_P(\sqrt{\ln J}) = O_P(\sqrt{\ln \ln n})$. Since $v_{s^*}^* \leq \max_{0\leq s\leq J} v_s^*$, $v_{s^*}^*$, it is asymptotically negligible compared to the detection threshold $\tau_d = O(\sqrt{\ln n})$ with high probability. So is it to $|R(\bar{\ell}_{s^*})|$ which by definition is larger than τ_d.

In contrast, at each scale, the candidate curve with the maximal absolute edge response is random. To control $\bar{v}_{s^*}$, we show such candidate curves can only belong to a certain set whose size we upper bound. Then Lemma 4 can be used. To this end, we prove that thanks to the offset Δ in the detection threshold, it is with high probability that a firing candidate curve enjoys a minimal covering with the edge g (Lemma 10) and there are at most $O((\ln n)^3)$ such candidate curves at any fixed scale (Theorem 3). Theorem 2 now follows. □

Lemma 10 *On a high probability event, any candidate curve ℓ firing at scale $s^* \in \{1, \ldots, J\}$, namely $|R(\ell)| > \tau_d$, satisfies*

$$\rho(g, \ell) \geq \frac{(\omega - \gamma^*)^2 \Delta}{2\sqrt{3}\omega^2 \tau_d}. \tag{36}$$

Theorem 3 *Let $g \in \mathcal{R}_{\boldsymbol{b},n}$. For any $\kappa \in [0, 5/8)$, in a strip of $O(n^\kappa)$ columns, there is a constant C independent of n and κ such that*

$$|\{\ell \in \mathcal{S}(\boldsymbol{b}, I, M_q, m, n) \mid \rho(g, \ell) > \rho\}| \leq \frac{C}{\rho^6}.$$

Proof of Lemma 10 Equations (33) and (34) not only yield a lower bound on the covering ratio (35), they also bound the maximal expected edge responses of two successive scales

$$|\mathbb{E}R(\ell_s^*)| \leq \frac{\sqrt{3}\omega|\mathbb{E}R(\ell_{s-1}^*)|}{(\omega - \gamma^*)}, \quad s = 1, \ldots, J \tag{37}$$

This holds, because by definition, $L_s \leq 3L_{s-1}$, for all $s \geq 1$, with equality at $s = 1$. Combining Eqs. (37) and (35) gives

$$\rho(g, \ell_s) \geq \frac{\omega - \gamma^*}{\omega}\left|\frac{\mathbb{E}R(\ell_s)}{\mathbb{E}R(\ell_s^*)}\right| \geq \frac{(\omega - \gamma^*)^2}{\sqrt{3}\omega^2}\left|\frac{\mathbb{E}R(\ell_s)}{\mathbb{E}R(\ell_{s-1}^*)}\right|.$$

Next, applying the triangle inequality gives, $\forall s \in \{1, \ldots, J\}$

$$\rho(g, \ell_s) \geq \frac{(\omega - \gamma^*)^2(|R(\ell_s)| - \max_{\ell' \in \mathcal{T}_J} |R(\ell') - \mathbb{E}R(\ell')|)}{\sqrt{3}\omega^2(|R(\bar{\ell}_{s-1})| + \max_{0\leq s\leq J} |R(\ell_s^*) - \mathbb{E}R(\ell_s^*)|)},$$

Let ℓ be any candidate curve firing at scale $s^* \in \{1, \ldots, J\}$. Then $|R(\ell)| > \tau_d$, and all candidate curves up to scale $s^* - 1$ failed to fire. Hence, $\max_{s<s^*} |R(\bar{\ell}_s)| \leq \tau_d$, and

$$\rho(g, \ell) \geq \frac{(\omega - \gamma^*)^2(\tau_d - \max_{\ell' \in \mathcal{T}_J} |R(\ell') - \mathbb{E}R(\ell')|)}{\sqrt{3}\omega^2(\tau_d + \max_{0\leq s\leq J} |R(\ell_s^*) - \mathbb{E}R(\ell_s^*)|)}.$$

On the event (31) which according to Lemma 4 has probability at least $1 - \alpha$, we thus have

$$\rho(g, \ell) \geq \frac{(\omega - \gamma^*)^2 \Delta}{2\sqrt{3}\omega^2 \tau_d}.$$

□

Proof of Theorem 3 The proof of Lemma 3 effectively shows that the sublinear runtime requires $a_k = 0$ for $k > 2$ when $n \to \infty$. We thus assume $r = 2$ in the following. Let $p_q \in \mathcal{S}_p(\boldsymbol{b}, I, M_q, n)$ be such that $\ell = \mathcal{I}_m(p_q)$ intersects the signal tube of an edge $g \in \mathcal{R}_{\boldsymbol{b},n}$ at point $x \in I$, that is $|\ell(x) - g(x)| \leq \omega - 1$. By Theorem 1, and the triangle inequality

$$\begin{aligned}|p_q(x) - \mathcal{P}^s_{I,M_q}(g)(x)| \leq |p_q(x) - \ell(x)| + |\ell(x) - g(x)| \\ + |g(x) - \mathcal{P}^s_{I,M_q}(g)(x)| \leq W\end{aligned}$$

where $W := \omega - 1 + E_1(I) + E_2(I, m)$. Let

$$A(p_q) = \sum_{k=1}^{L} 1_{|\mathcal{P}^s_{I,M_q}(g)(x_k) - p_q(x_k)| \leq W}$$

where L is the number of strip columns. Hence, we find

$$A(p_q) \geq \sum_{k=1}^{L} 1_{|g(x_k) - \ell(x_k)| \leq \omega - 1} = \rho(g, \ell)L.$$

To prove the theorem, it suffices to show that there is some constant C independent of n and κ such that for any $\rho > 0$,

$$\left|\left\{p_q \in \mathcal{S}_p(\boldsymbol{b}, I, M_q, n),\ A(p_q) > \rho L\right\}\right| \leq \frac{C}{\rho^6}.$$

To this end, we study two separate cases $\rho L \geq 6$ and $\rho L < 6$.

First assume $\rho L \geq 6$. Since $d_q := \mathcal{P}^s_{I,M_q}(g) - p_q$ is a quantized polynomial of degree at most 2, the set of integers $\{x_k \in I,\ |d_q(x_k)| \leq W\}$ can be divided into at most 2 groups of contiguous integers on which d_q is monotone. The largest group must therefore have at least $\lceil \rho L/2 \rceil$ elements. Let $I' \subset$

I denote the smallest interval containing these integers. Its length satisfies

$$\lambda(I') \geq (\rho L/2 - 1)_+ \overset{(i)}{\geq} \rho L/3 > \rho\lambda(I)/3 \tag{38}$$

where inequality (i) holds thanks to the assumption $\rho L \geq 6$.

The polynomial d_q is uniformly bounded by W over I', which implies a bound on its coefficients. It can be proved, for instance, by Markov's theorem (see, e.g., [1]). □

Theorem 4 (Markov) *Consider the universal constants*

$$C_{r,k} = \begin{cases} \prod_{j=0}^{k-1} \frac{r^2-j^2}{2j+1}, & k > 0 \\ 1, & k = 0 \end{cases}$$

Then for any polynomial $p(x)$ of degree r and any $k \leq r$

$$\sup_{x\in[-1,1]} |p^{(k)}(x)| \leq C_{r,k} \sup_{x\in[-1,1]} |p(x)|.$$

The idea is thus to turn the bounds on the coefficients into a bound on the number of such polynomials. Specifically, let $I' = [y_0, y_1]$ and apply Markov's theorem to the polynomial

$$\bar{d}_q(x) = d_q\left(\frac{y_1 - y_0}{2}x + \frac{y_1 + y_0}{2}\right)$$

which is uniformly bounded on the interval $[-1, 1]$ by W. We obtain for $k = 1, 2$

$$\sup_{x\in I'} |d_q^{(k)}(x)| \leq C_{2,k}\frac{2^k W}{\lambda(I')^k} < C_{2,k}\frac{6^k W}{\rho^k\lambda(I)^k}, \tag{39}$$

where the last inequality follows from Eq. (38). Let c denote the interval I's midpoint and $y_* = \text{argmin}_{y\in I'}|y - c|$. It is clear $|y_* - c| < \lambda(I)/2$. Hence for all $k = 0, 1, 2$

$$\begin{aligned} |d_q^{(k)}(c)| &\leq |d_q^{(k)}(c) - d_q^{(k)}(y_*)| + |d_q^{(k)}(y_*)| \\ &\leq \frac{2^k}{\lambda(I)^k}\sum_{i=k}^{2}\frac{|d_q^{(i)}(y_*)|\lambda(I)^i}{(i-k)!2^i}. \end{aligned} \tag{40}$$

Substituting the estimates of Eq. (39) into (40), we obtain

$$\begin{aligned} |d_q^{(k)}(c)| &\leq \frac{2^k}{\lambda(I)^k}\sum_{i=k}^{2}\frac{|d_q^{(i)}(y_*)|\lambda(I)^i}{(i-k)!2^i} \\ &< \frac{2^k W}{\lambda(I)^k}\sum_{i=k}^{2}\frac{C_{2,i}3^i}{(i-k)!\rho^i}. \end{aligned}$$

Hence if we write d_q as

$$d_q(x) = \sum_{k=0}^{2}\frac{\bar{a}_k}{M_q}\left(\frac{x-c}{\lambda(I)/2}\right)^k,$$

its quantized coefficients $(\bar{a}_k)_{0\leq k\leq 2}$ can be bounded

$$|\bar{a}_k| = \frac{M_q\lambda(I)^k}{k!2^k}|d_q^{(k)}(c)| \leq \frac{M_q W}{\rho^2 k!}\sum_{i=k}^{2}\frac{C_{2,i}3^i}{(i-k)!}.$$

Hence when $\rho L \geq 6$, the number of polynomials can be upper bounded by $C\frac{(M_q W)^3}{\rho^6}$ for a universal constant C.

Turn to the case $\rho L < 6$. If a candidate curve ℓ satisfies $|\ell(x) - g(x)| \leq \omega - 1$ at some $x \in I$, we deduce

$$|\ell(x) - g(c)| \leq \omega - 1 + \frac{b_1\lambda(I)}{2}.$$

As a result, we are interested in bounding the size of

$$\mathcal{B} = \left\{ p_q \;\middle|\; \min_{x\in I}|\mathcal{I}_m(p_q)(x) - g(c)| \leq \omega - 1 + \tfrac{b_1\lambda(I)}{2} \right\}.$$

To this end, writing a generic element in the set $\mathcal{B}$ as

$$p_q(x) = \sum_{k=0}^{2}\frac{a_k}{M_q}\left(\frac{x-c}{\lambda(I)/2}\right)^k,$$

we aim to bound a_0. Applying the triangle inequality gives

$$\begin{aligned} &|p_q(c) - g(c)| \\ &\leq \min_{x\in I}\Big(|p_q(c) - p_q(x)| + |p_q(x) - \ell(x)| + |\ell(x) - g(c)|\Big) \\ &\leq \frac{1}{M_q}\sum_{k=1}^{2}|a_k| + E_2(I, m) + \omega - 1 + \frac{b_1\lambda(I)}{2} \\ &\leq b_1\lambda(I)(1 + O(1)) \end{aligned}$$

thanks to Eq. (12). Multiplying both sides by M_q, we find that $|a_0 - g(c)M_q| \leq M_q b_1\lambda(I)(1 + O(1))$, and

$$|\mathcal{B}| \leq M_q b_1\lambda(I)\,(1 + O(1))\prod_{k=1}^{2}\left(\frac{b_k\lambda(I)^k M_q}{(2n)^{k-1}} + 2\right).$$

As the condition $\rho L < 6$ implies $\lambda(I) < \frac{6}{\rho}$, we conclude $|\mathcal{B}| \leq \frac{C'}{\rho^3}$ for some constant C' independent of n and κ.

Proof of Lemma 7 From Eq. (24), for all $k = 1, \ldots, r$

$$\left|\frac{a_k^{(i)}k!}{M_q\lambda(I_i')^k} - g^{(k)}\left(x_i'\right)\right| \leq \frac{k!}{2M_q\lambda(I_i')^k}, \quad i = 0, 1.$$

An application of the triangle inequality leads to

$$\begin{aligned} \left|\frac{a_k^{(0)}k!}{M_q\lambda(I_0')^k} - \frac{a_k^{(1)}k!}{M_q\lambda(I_1')^k}\right| &\leq \frac{k!}{2M_q\lambda(I_0')^k} + \frac{k!}{2M_q\lambda(I_1')^k} \\ &\quad + \left|g^{(k)}\left(x_1'\right) - g^{(k)}\left(x_0'\right)\right|. \end{aligned}$$

Since $\left|g^{(k)}\left(x_1'\right) - g^{(k)}\left(x_0'\right)\right| \le \min\left(2\|g^{(k)}\|_\infty, \lambda(I_0')\|g^{(k+1)}\|_\infty\right)$, Eq.(28) of the lemma readily follows.

Regarding the constant term $a_0^{(1)}$, thanks to the uniform approximation error bound (16), it satisfies

$$\begin{aligned}\left|\frac{a_0^{(1)}}{M_q} - p_q^{(0)}(x_1')\right| &\le \left|\frac{a_0^{(1)}}{M_q} - g(x_1')\right| + \left|g(x_1') - p_q^{(0)}(x_1')\right| \\ &\le \tfrac{r+2}{2M_q} + \tfrac{b_{r+1}}{n^r}\lambda(I_0')^{r+1}.\end{aligned}$$

□

Proof of Eq. (7) We start by showing that the condition $\sup_{x\in[x_1,x_L]}|h(x) - g(x)| \le \gamma$ implies that

$$\max_{1\le k\le L} |\lfloor h(x_k)\rfloor - \lfloor g(x_k)\rfloor| \le \gamma. \tag{41}$$

To this end, note that the following holds for all $(z_1, z_2) \in \mathbb{R}^2$

$$\begin{aligned}|\lfloor z_1\rfloor - \lfloor z_2\rfloor| &\overset{(i)}{\le} |z_1 - z_2| + |(z_2 - \lfloor z_2\rfloor) - (z_1 - \lfloor z_1\rfloor)| \\ &\overset{(ii)}{<} |z_1 - z_2| + 1.\end{aligned}$$

(i) follows from the triangle inequality. As $z_2 - \lfloor z_2\rfloor$ and $z_1 - \lfloor z_1\rfloor$ take values in $[0, 1)$, their difference is strictly smaller than 1 in absolute value, hence inequality (ii). Consequently,

$$\begin{aligned}\max_{1\le k\le L} |\lfloor h(x_k)\rfloor - \lfloor g(x_k)\rfloor| &< 1 + \max_{1\le k\le L} |h(x_k) - g(x_k)| \\ &\le 1 + \sup_{x\in[x_1,x_L]} |h(x) - g(x)| \\ &\le 1 + \gamma.\end{aligned}$$

Since $\max_{1\le k\le L} |\lfloor h(x_k)\rfloor - \lfloor g(x_k)\rfloor|$ is an integer, its being strictly smaller than $\gamma + 1$ can only imply Eq. (41).

To prove Eq. (7), we use Eq. (6) which gives

$$\mathbb{E}R(h) = \frac{1}{\sigma\sqrt{L}}\sum_{k=1}^{L} \mathbb{E}\left[R(x_k, \lfloor h(x_k)\rfloor)\right]. \tag{42}$$

If there is only one edge, all the summands in (42) have the same sign. Hence, we deduce from Eqs. (5) and (41) that

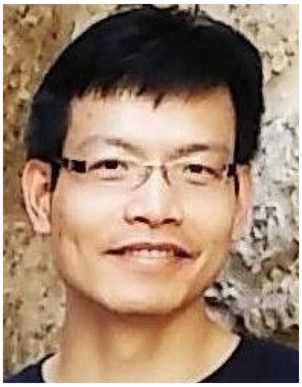
$$\begin{aligned}|\mathbb{E}R(h)| &= \frac{1}{\sigma\sqrt{L}}\sum_{k=1}^{L} |\mathbb{E}\left[R(x_k, \lfloor h(x_k)\rfloor)\right]| \\ &\ge \frac{\mu_{\Gamma(g)}\sqrt{L}(\omega - \gamma)_+}{\sigma\sqrt{2\omega}}.\end{aligned}$$

□

References

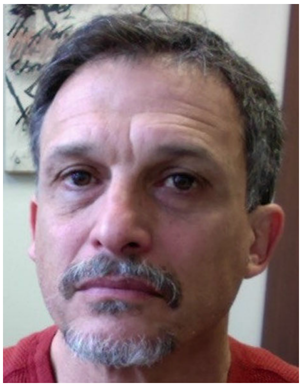
1. Achieser, N.I.: Theory of approximation. Dover Publications (1992)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. **33**(5), 898–916 (2011)
3. Arias-Castro, E., Efros, B., Levi, O.: Networks of polynomial pieces with application to the analysis of point clouds and images. J. Approx. Theory **1**(162), 94–130 (2010)
4. Brandt, A., Dym, J.: Fast calculation of multiple line integrals. SIAM J. Sci. Comput. **20**(4), 1417–1429 (1999)
5. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. **8**(6), 679–698 (1986)
6. Deriche, R.: Using Canny's criteria to derive a recursively implemented optimal edge detector. Int. J. Comput. Vis. **1**(2), 167–187 (1987)
7. Desolneux, A., Moisan, L., Morel, J.M.: Edge detection by Helmholtz principle. J. Math. Imaging Vis. **14**(3), 271–284 (2001)
8. Dollár, P., Zitnick, C.: Structured forests for fast edge detection. In: IEEE International Conference on Computer Vision, pp. 1841–1848 (2013)
9. Gonzalez, R.C., Woods, R.E.: Digital image processing, 2nd edn. Prentice Hall (2002)
10. Haupt, J., Castro, R., Nowak, R.: Distilled sensing: adaptive sampling for sparse detection and estimation. IEEE Trans. Inform. Theory **57**(9), 6222–6235 (2011)
11. Horev, I., Nadler, B., Arias-Castro, E., Galun, M., Basri, R.: Detection of long edges on a computational budget: a sublinear approach. SIAM J. Imaging Sci. **8**(1), 458–483 (2015)
12. Kimmel, R., Bruckstein, A.M.: Regularized Laplacian zero crossings as optimal edge integrators. Int. J. Comput. Vis. **53**(3), 225–243 (2003)
13. Kiryati, N., Eldar, Y., Bruckstein, A.M.: A probabilistic hough transform. Lect. Notes Comput. Sci. **24**(4), 303–316 (1991)
14. Kiryati, N., Kälviäinen, H., Alaoutinen, S.: Randomized or probabilistic hough transform: unified performance evaluation. Pattern Recogn. Lett. **21**(13), 1157–1164 (2000)
15. Kleiner, I., Keren, D., Newman, I., Ben-Zwi, O.: Applying property testing to an image partitioning problem. IEEE Trans. Pattern Anal. **33**(2), 256–265 (2011)
16. Konishi, S., Yuille, A., Coughlan, J., Zhu, S.C.: Statistical edge detection: learning and evaluating edge cues. IEEE Trans. Pattern Anal. **25**(1), 57–74 (2003)
17. Korman, S., Reichman, D., Tsur, G., Avidan, S.: Fast-match: fast affine template matching. In: Proceedings of the CVPR IEEE, pp. 2331–2338 (2013)
18. Laurent, B., Massart, P.: Adaptive estimation of a quadratic functional by model selection. Ann. Statist. **28**(5), 1302–1338 (2000)
19. Lebrun, M., Colom, M., Buades, A., Morel, J.: Secrets of image denoising cuisine. Acta Numer. **21**, 475–576 (2012)
20. Liu, C., Freeman, W., Szeliski, R., Kang, S.: Noise estimation from a single image. In: Proceedings of the CVPR IEEE, vol. 1, pp. 901–908. IEEE (2006)
21. Marr, D., Hildreth, E.: Theory of edge detection. Philos. Trans. R. Soc. B **207**(1167), 187–217 (1980)
22. Ofir, N., Galun, M., Nadler, B., Basri, R.: Fast detection of curved edges at low SNR. In: Proceedings of the CVPR. IEEE (2016)
23. Rao, K.R., Ben-Arie, J.: Optimal edge detection using expansion matching and restoration. IEEE Trans. Pattern Anal. **16**(12), 1169–1182 (1994)
24. Raskhodnikova, S.: Approximate testing of visual properties. In: Lecture Notes in Computer Science, pp. 370–381. Springer (2003)
25. Rousseeuw, P., Croux, C.: Alternatives to the median absolute deviation. J. Am. Stat. Assoc. **88**(424), 1273–1283 (1993)

26. Rubinfeld, R., Shapira, A.: Sublinear time algorithms. SIAM J. Discrete Math. **25**(4), 1562–1588 (2011)
27. Shen, J., Castan, S.: An optimal linear operator for step edge detection. Graph. Models Image Process. **54**(2), 112–133 (1992)
28. Tsur, G., Ron, D.: Testing properties of sparse images. In: International Journal in Foundations of Computer Science, pp. 468–477. IEEE (2010)
29. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: On straight line segment detection. J. Math. Imaging Vis. **32**(3), 313–347 (2008)
30. von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. **32**(4), 722–732 (2008)
31. Willett, R., Martin, A., Nowak, R.: Backcasting: adaptive sampling for sensor networks. In: Lecture Notes in Computer Science, pp. 124–133. ACM (2004)
32. Xu, L., Oja, E., Kultanen, P.: A new curve detection method: randomized hough transform (RHT). Pattern Recogn. Lett. **11**(5), 331–338 (1990)



Yi-Qing Wang received the diplôme d'ingénieur from Ecole Polytechnique, Palaiseau, France, in 2008, the M.Sc. degree in probability from Université Pierre et Marie Curie, Paris, France, in 2009 and the M.Sc. degree in Applied Mathematics from Ecole Normale Supérieure de Cachan, France, in 2012. He obtained his Ph.D. degree in March, 2015 from Ecole Normale Supérieure de Cachan. Currently, he is a postdoctoral researcher in the Department of Statistics at UCLA. His research interests include statistical learning and its applications to image processing and computer vision.

Alain Trouvé is currently Professor at the Center of Mathematics and Their Application at ENS Paris-Saclay. He did his Ph.D. in Stochastic Optimization and Bayesian Image Analysis under the supervision of Robert Azencott. His main research interests are computational vision and shape analysis with a particular emphasis on the use of Riemannian geometry and infinite dimensional group actions driven by applications in computational anatomy and medical imaging.



Yali Amit received his Ph.D. in Mathematics at the Weizmann Institute, Israel in 1988. He spent 3 years as a visiting assistant professor in the Division of Applied Math at Brown University where he started working on image analysis. In 1991, he joined the Department of Statistics at the University of Chicago. In 2000, he was appointed full Professor with a joint appointment in Statistics and Computer Science. He served as chair of the Statistics Department between 2010–2016. His research interests have focused on object detection and recognition, speech recognition, machine learning, the biological visual system, and computational models for learning and memory in neural networks.

Boaz Nadler holds a Ph.D. in Applied Mathematics from Tel-Aviv University, Israel. After 3 years as a Gibbs instructor/assistant professor at the Mathematics Department at Yale, he joined the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, where he is currently associate professor. His research interests are in mathematical statistics, machine learning, and applications in optics, signal and image processing.