A contrario dip picking for borehole imaging

Joris Costes, Gabriele Facciolo, Rafael Grompone von Gioi, Josselin Kherroubi, Enric Meinhardt-Llopis, Jean-Michel Morel

A contrario dip picking for borehole imaging

Joris Costes* and Gabriele Facciolo* and Rafael Grompone von Gioi* and Josselin Kherroubi† and Enric Meinhardt-Llopis* and Jean-Michel Morel*

ABSTRACT

We describe an algorithm to perform automatic dip picking on borehole images. One key element of the proposed method is a statistical validation, based on the *a contrario* theory, which is used to decide whether each candidate dip is to be accepted or not. The proposed method also uses a randomized Hough transform, which greatly improves the processing speed, allowing for a real-time detection of dips during image visualization. In addition, the same algorithm can be applied at different scales to provide a multi-resolution analysis of the structures. Our experiments show that the proposed algorithm produces reliable dip picking by an evaluation on three manually annotated boreholes: the proposed method detects from 60% to 90% of the dips annotated by an expert, depending on the complexity of the data.

INTRODUCTION

An essential tool in geological exploration are well logs, which record the measurement of physical quantities along the depth of a borehole. Modern measurement devices acquire two-dimensional data over the whole surface of the well. Meaningful information from the resulting *borehole images* can then be extracted using powerful image processing tools (Gonzalez et al., 2020; Jiang et al., 2020).

Dip picking is an important task in the geological interpretation of borehole images for structural, sedimentary, and fracture analysis (Lofts and Bourke, 1999). It consists in extracting the planar structures observable in the borehole image (known as *dips*) and characterizing them. Borehole images are high-resolution measurements allowing for dip characterization (Lowell et al., 1999). As shown in Figure 1, the imaging measurements are acquired on a cylinder-shaped borehole and result from the unrolling of this cylinder. As a consequence, the horizontal axis is the apparent azimuth angle and the vertical axis is the measured depth along the borehole trajectory. Any planar event crossing the borehole results approximately on a one period sinusoid trace on the unrolled image. As soon as this trace is extracted, the plane characteristics—dip inclination, azimuth and depth—can be directly computed.

The task of dip picking can be performed manually by visual inspection and annotation of the borehole images. This process is very time-consuming and requires a qualified team of experts to annotate a whole well dataset. Thus, several algorithms have been proposed in the literature to help in this task, either by computing the dips automatically, or in a semi-supervised way. Dip picking can be the basis of further geometrical analysis (Trice, 1999; Kherroubi et al., 2016).

The Hough transform (Hough, 1962; Duda and Hart, 1972; Illingworth and Kittler, 1988) is a feature extraction technique widely used to detect structures such as lines, ellipses, or sinusoids. In the Hough transform each pixel cast votes for the features compatible with its value and coordinates, the votes of all the pixels are accumulated in the parameter space of the sought structure. The final state of the accumulator space is the result of the transformation. Then, maxima in this transformed space are used to extract candidate structures.

In the pioneering work of Torres (1992), the Hough transform is used to extract sinusoidal structures in edge maps (Marr and Hildreth, 1980) extracted from borehole images. A later improvement of this method uses directly the gradient of the borehole images instead of an edge detector (Saito et al., 1999). Other methods also using the Hough transform are Hall et al. (1996) and van Ginkel et al. (2003). These methods produce an automatic dip picking, their main drawback being the computational

* Centre Borelli, École Normale Supérieure Paris-Saclay, Gif-sur-Yvette, France
† Schlumberger, Paris, France

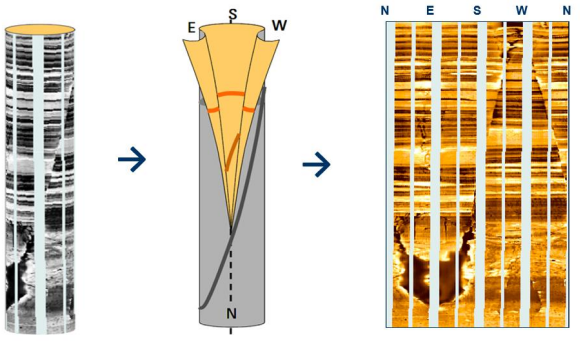


Figure 1: Borehole image: from cylinder to an unrolled image. On the right image, a typical colormap used by geologists is applied on the grayscale image. We can clearly see a fault crossing the middle of the image (the sinusoid with high amplitude), surrounded by many sinusoids with very small amplitude, thus nearly horizontal lines, corresponding to the sedimentary structure. Image courtesy of Schlumberger.

cost when a fine quantization of the parameter space is required.

To obtain an interactive user experience, Wang et al. (2007) discloses a user-guided dip picking method, which is based in defining a top and bottom reference curve. These constraints define a smaller subset of the parameter space that can be sampled in real time to find the optimal parameters of a family of sinusoids.

Assous et al. (2014) exploit the symmetries present in sinusoids to accumulate votes for finding likely structure locations. Sun and Pallottino (2003) use a shortest path algorithm to trace a structure across an entire borehole image, exploiting the connection between the left and right sides of the image. Wedge et al. (2015) propose a method for detecting sinusoids that group smaller edge elements, which also exploit the vertical and horizontal sinusoidal symmetries. In addition, the method provides interactive access to the detections.

In this paper, we propose a dip picking method that can process automatically the whole well data in one run, or be integrated in a semi-supervised human/computer interface where the user selects a region of interest. The proposed method involves a statistical validation method for each candidate sinusoid. Its criterion is based on the *a contrario* theory (Desolneux et al., 2008). This statistical test could be used directly to find a good set of sinusoids (in the sense of *number of false alarms*, as explained below) by an exhaustive exploration of the whole parameter space. However, to obtain a real-time algorithm, we use the randomized Hough transform (Xu et al., 1990) to initialize a local search in the parameter space. In addition, the relatively slow variation of dip inclination and azimuth allows for a local exploration of these two parameters, thus reducing even more the required number of random samples to produce reliable candidates.

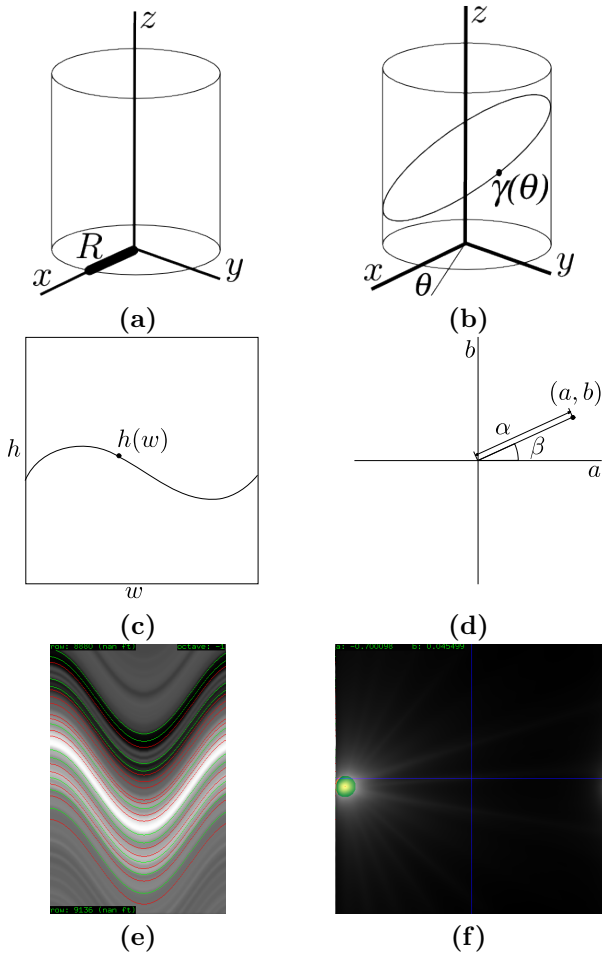


Figure 2: **(a)** Cylinder of radius R. **(b)** Intersection of the cylinder with a plane. **(c)** When the cylinder is unrolled, this ellipse becomes a sinusoid. See equation (4) for the equation of a sinusoid $h(w)$ in terms of its Hough parameters (a, b). **(d)** Sinusoid in Hough (a, b) coordinates. **(e)** Dip detection in a synthetic image (red for dark-to-bright and green for bright-to-dark detections). **(f)** Corresponding votes in Hough space.

To quantitatively assess the performance of our method, we evaluate its performance on three borehole images with manually annotated dips. We also conduct ablation studies to illustrate the impact of some key components of the proposed method. Our experiments show that the proposed method allows to extract dips reliably on full borehole images. This is supported by the fact that the method is currently used in production for a commercial service. This method could also be applied on any borehole imaging tool (whatever the measurement physics), in wireline logging or measurement while drilling (Drilling and Measurement). The same approach could also be used for extracting dips from slab core pictures: in that case, the Hough transform algorithm must be adapted to extract lines instead of sinusoids.

PROPOSED METHOD

For the sake of simplicity of exposition, we suppose that the borehole is a straight and vertical cylinder (i.e., the borehole does not change its trajectory); thus, the azimuth and depth have an absolute meaning. Furthermore, assuming that the borehole is a perfect cylinder (i.e., that the cross-section is a circle), planar events appear exactly sinusoidal in the image. In practice, boreholes are never straight, but the dip-picking is a local process that only needs the borehole to be locally straight. The method can easily be extended to the general case of a 3D borehole path, including a varying and not perfect circular cross-section; of course, the borehole parameters need to be known.

Let us consider a borehole as a cylinder of radius R (see Figure 2). The borehole image is obtained by unrolling the surface of the cylinder into an image of size $W \times H$. Thus, a pixel $(w, h) \in [0, W-1] \times [0, H-1]$ corresponds to the 3D point of coordinates

$$\left(R\cos\frac{2\pi w}{W},\ \ R\sin\frac{2\pi w}{W},\ \ \frac{2\pi R}{W}h \right). \tag{1}$$

Note that in the equation above, the third dimension is scaled by a factor $\frac{2\pi R}{W}$; this factor is required to obtain square pixels, having the same height and width in the 3D space.

Let us consider a (non-vertical) 3D plane of equation $z = ax + by + c$. If we assume that the plane is determined by an *inclination* angle α and an *azimuth* angle β, then the parameters correspond to $a = K\sin(\alpha)\cos(\beta)$ and $b = K\sin(\alpha)\sin(\beta)$, where K is a constant. This plane intersects the cylinder on an ellipse that can be parametrized as

$$\theta \mapsto (R\cos\theta,\ \ R\sin\theta,\ \ aR\cos\theta + bR\sin\theta + c). \tag{2}$$

When representing this curve on the image coordinates it becomes

$$\theta \mapsto \left(\frac{W\theta}{2\pi},\ \ \frac{W}{2\pi}\left(a\cos\theta + b\sin\theta + \frac{c}{R} \right) \right) \tag{3}$$

or, equivalently, as the graph of the function

$$h(w) = \frac{W}{2\pi}\left(a\cos\frac{2\pi w}{W} + b\sin\frac{2\pi w}{W} + \frac{c}{R} \right) \tag{4}$$

in the image plane (w, h). By replacing the expressions of a and b and using a trigonometric addition formula, the expression $a\cos\theta + b\sin\theta + c$ can be rewritten as to $K\sin(\alpha)\cos(\theta - \beta) + c$, parametrized by the inclination α and the azimuth β. Thus, the function $h(w)$ is a sinusoid of one period, whose phase determines the azimuth, and whose slope determines the inclination of the dip.

The task of dip picking consists in detecting these sinusoidal structures in the borehole images, and assigning an inclination, azimuth and depth to each. Here we propose an algorithm to do that automatically. The core of the proposed method is a statistical test that decides whether any candidate sinusoid is to be accepted or not; in other words, whether the sinusoid separates two groups of pixels whose intensities are significantly different. This criterion is based on the *a contrario* theory (Desolneux et al., 2008) and is described below.

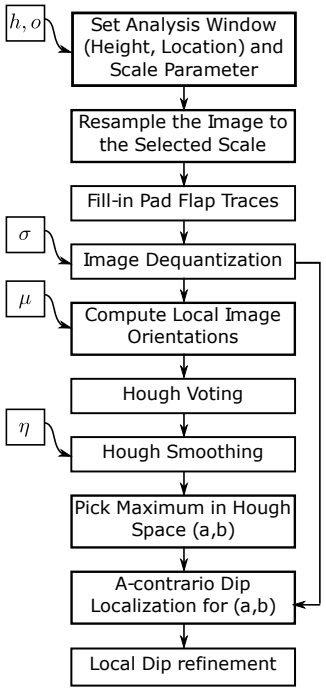


Figure 3: Workflow of the proposed method.

Such a statistical test could be used directly to find a good set of sinusoids by exhaustive exploration of the whole parameter space. In that case one would traverse the set of all possible sinusoids (a, b, c), and for each sinusoid compute the statistic test which requires evaluating the image at about W positions. Such an exhaustive algorithm would have a computational complexity of N^4, where N is the number of possible values for each of a, b, c or W. Here, we propose a faster heuristic search on this space, initializing the detection with plausible starting dips.

Algorithm overview

The overall structure of the proposed algorithm is laid out on Figure 3 and on Algorithm 1. The statistical test actually happens on the last steps of the method, because it is used only to validate and refine a rough set of dips obtained by a fast but somewhat imprecise Hough transform.

The algorithm finds all the dips inside a sub-window of the image specified by its vertical location h and scale octave o. The octave determines a vertically down-scaled version (by factors of 2^o) of the original image. Working on the down-scaled image allows to detect dips with large slopes that would otherwise not fit in a window. But the detection algorithm is the same regardless of the scale.

Algorithm 1: `dip-picker`

input : Image J of size $W \times H$
input : Valid data mask M $(W \times H)$
input : Location $h \in \{1, \dots, H\}$
input : Scale $\mathscr{o} \in \{1, \dots, \lfloor \log_2 H \rfloor\}$
input : Positive parameters $\sigma, \mu, \eta, \rho, \epsilon, \kappa, N_{rand}$
output : List of sinusoids L

$I \leftarrow$ `extract-and-resample-window`$(J, M, h, \mathscr{o})$
$I_f \leftarrow$ `fill-in-missing-data`(I, M)
$I_\sigma \leftarrow$ `gaussian-blur`(I_f, σ)
$G \leftarrow$ `local-orientation`(I_σ, μ)
$H \leftarrow$ `randomized-hough-voting`(G, N_{rand}, κ)
$H_\eta \leftarrow$ `gaussian-blur`(H, η)
$(a, b) \leftarrow \arg\max H_\eta(x, y)$
$L \leftarrow$ `locate-dips`$(I_\sigma, M, \rho, \varepsilon, a, b)$
$L \leftarrow$ `refine-dips-with-exclusion`$(I_\sigma, M, \rho, \varepsilon, L)$
return L

This window can be either selected interactively by the user, or slid along the whole image for an automatic analysis.

The algorithm relies on the local image orientation inside the target window. But two pre-processing are needed: filling-in pad/flap traces, and dequantization and filtering. Then, the local orientations are computed and used twice in the method: first to estimate the main dip inside the window (using the Hough transform), second to find the optimal placement of meaningful dips using an *a contrario* strategy.

In the following paragraphs we describe each of the steps of the algorithm in detail.

Filling-in pad/flap traces

In the case of pad-based borehole imaging tools, the images are incomplete due to blind spots between the pads/flaps of the sensor (see the vertical stripes on Figure 1). In the proposed method the decision of whether a dip is present or not only uses the available data, ignoring the pad/flap traces. Nevertheless, filling the missing data simplifies the pipeline for proposing candidates by avoiding the tracking of the missing parts. These traces could be inpainted using advanced methods that aim at reconstructing the missing structures, such as the one proposed in Zhang et al. (2019). In our case it is sufficient to handle the boundary conditions between the known and unknown data. This is done by solving the Laplace equation $\Delta u = 0$ inside the unknown region with boundary conditions given by the known data. This creates a smooth interpolation of the data (see Figure 4) which will not perturb the computation of the local orientation. This is the simplest possible method of data interpolation using PDEs (Weickert and Welk, 2006). Note that the quality of this filling is not really important as it is only used to simplify the computations of the intermediate gradients, and that in the final validation the interpolated data is not used.

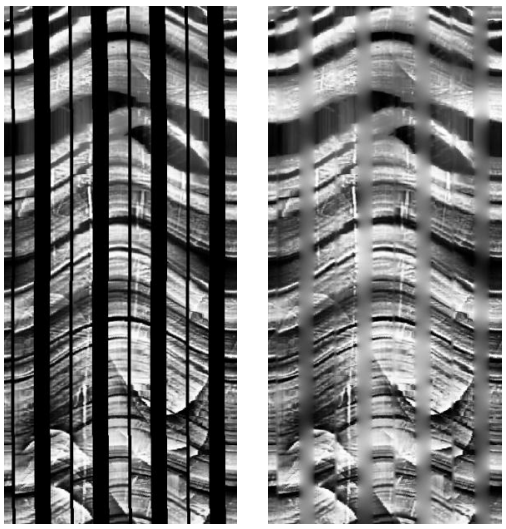


Figure 4: Example of Filling-in produced by the proposed Laplacian interpolation method. This creates a smooth interpolation of the data, which will not perturb the computation of the local orientation.

Dequantization and filtering

The next preprocessing step of the proposed method consists in filtering the image by a Gaussian kernel of size σ. This filtering serves two different goals: the dequantization of the local orientations and the selection of the "grain/resolution" size for the analysis.

The dequantization is essential for computing the Hough transform: if we set $\sigma = 0$, we do not obtain meaningful results, due to severe quantization artifacts. These artifacts arise because the gradient direction of a quantized image is not uniformly distributed as there is only a finite number of possible discrete orientations, and some orientations turn out to be much more likely than others (Desolneux et al., 2002). When computing the Hough transform, this non-uniformity appears as a set of concentric circles around the origin of coordinates (Figure 5). Smoothing the input images by a small Gaussian blur dequantizes the image and the artifacts on the transformed image disappear, allowing to correctly locate the maximum.

The parameter σ also allows to control the resolution of the analysis: using a larger σ permits to move the focus of the algorithm from thin strata to thicker structures. The impact of this parameter is discussed in detail in the section *Multiscale dip picking*.

Local image orientation

The proposed method relies on a point-wise estimation of the local orientation of the image structures. The simplest measure of local image orientation is the image gradient $\nabla I_\sigma = (\partial_x I_\sigma, \partial_y I_\sigma)$, which can be computed with a

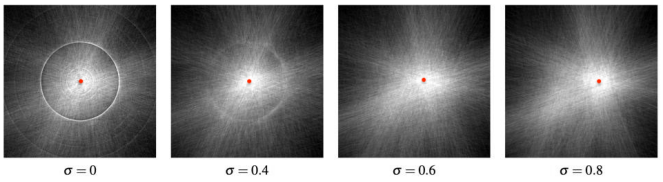


Figure 5: Importance of image dequantization. These images show the Hough transform of the same input image, pre-filtered with different Gaussian kernels of the indicated sigma. The circular artifacts are due to the quantized gray levels of the image. They interfere severely with the location of the maximum, which is indicated as a red circle in the images. The transform is dequantized by applying a small blur to the input images.

finite difference scheme. The main issue with the image gradient is that it encodes both orientation and direction (sign), which means that averaging gradients may cancel them out. Despite this, the gradient provides valuable information about the image and can be used successfully in the context of dip picking (Saito et al., 1999).

A better local image orientation descriptor is the structure tensor (Förstner and Gülch, 1987), which is a field representing a local average direction (modulo π) of the gradient vectors (modulo 2π). This is better than the blurred gradient because opposed vectors will vote for the same direction in Hough space (see next section). The structure tensor field S is computed from the image gradients field ∇I_σ by smoothing the singular matrix field $\nabla I_\sigma \otimes \nabla I_\sigma$, namely $S := G_\mu * (\nabla I_\sigma \otimes \nabla I_\sigma)$ where G_μ is a Gaussian kernel, and $\otimes$ is the outer product of a pair of vectors (rank-1 matrix).

Note that the Gaussian smoothing G_μ is not interchangeable with the previous smoothing G_σ used for removing the quantization artifacts. Indeed, the outer product results in the components I_x^2, $I_x I_y$ and I_y^2, which are non-linear operations; thus, we cannot remove the first smoothing step by using a stronger smoothing value μ. These two smoothing steps are both needed for different reasons, and if the quantization artifacts are not corrected before, the resulting structure tensor will present artifacts as those shown in Figure 5.

The local orientation is computed from the 2×2 matrix $S_\mu(i,j)$, as the eigenvector associated to its largest eigenvalue, and the strength of this orientation is the largest eigenvalue. These eigenvalues will be used to weight the Hough votes. The pseudo-code for computing the structure tensor and its largest eigenvector is detailed in Algorithm 2.

Algorithm 2: `local-orientation-by-structure-tensor`

input : Image $I_\sigma : \Omega \to \mathbf{R}$
input : Structure tensor filtering $\mu > 0$
output : Local orientation field $G : \Omega \to \mathbf{R}^2$

$I_x \leftarrow \frac{\partial}{\partial x} I_\sigma$ // finite difference derivatives
$I_y \leftarrow \frac{\partial}{\partial y} I_\sigma$
$P \leftarrow$ `gaussian-blur`(I_x^2, μ)
$Q \leftarrow$ `gaussian-blur`$(I_x I_y, \mu)$
$R \leftarrow$ `gaussian-blur`(I_y^2, μ)
for $p \in \Omega$ **do**
  $\left(\begin{smallmatrix} a & b \\ b & c \end{smallmatrix}\right) \leftarrow \left(\begin{smallmatrix} P(p) & Q(p) \\ Q(p) & R(p) \end{smallmatrix}\right)$ // structure tensor at p
  $T \leftarrow a + c$
  $D \leftarrow ac - b^2$
  $\alpha \leftarrow \frac{T + \sqrt{T^2 - 4D}}{2}$ // largest eigenvalue
  if $|b| > 0$ **then**
    $G(p) \leftarrow (b, \alpha - a)$ // eigenvector
  else
    $G(p) \leftarrow (a, 0)$
return G

Hough voting

The purpose of this step is to find the main bundle of sinusoids (the average dip) appearing on the selected window. This is done by means of the Hough transform. The Hough transform (Hough, 1962; Duda and Hart, 1972; Illingworth and Kittler, 1988) can be used to find instances of objects within a certain class of shapes by a voting procedure. The class of shapes must be described by a set of parameters. This defines a *parameter space* in which each point correspond to a particular shape. An *accumulator* is defined in the parameter space to count the number of votes for each particular parameter value. Then a voting procedure is carried out where each image feature increases the accumulator for the shapes compatible with the observation. The object candidates are obtained as local maxima in the accumulator.

Recall that a dip of parameters (a, b) corresponds to a certain sinusoid $h_{a,b}(w)$ as in equation (4). This dip is compatible with a local image orientation $(u, v) = S(w, h)$, given by the structure tensor at pixel (w, h), if the vector (u, v) is perpendicular to the curve $h_{a,b}(w)$, namely if the following condition holds

$$\big(1,\ h'_{a,b}(w)\big) \cdot (u, v) = 0. \tag{5}$$

Developing the derivative, this condition becomes

$$u \ - \ a\, v \sin\left(\frac{2\pi w}{W}\right) \ + \ b\, v \cos\left(\frac{2\pi w}{W}\right) \ = \ 0. \tag{6}$$

This equation determines a straight line in the (a, b)-plane, and the points of this straight line correspond to all the sinusoids compatible with the local gradient orientation (u, v) observed at (w, h).

The basic idea of Hough voting consists in choosing a finite discretization of the (a, b) parameter plane. Then, each point in the image votes for the corresponding straight line in the (a, b) space according to equation (6). The vote is weighted by the magnitude of the local orientation descriptor. Once all the points in the image domain have

voted, the site on the (a, b) plane with the most votes is selected as a candidate structure. The complexity of this algorithm is $O\left(N^{1.5}\right)$ where N is the number of pixels in the images (assuming that the size of Hough space is proportional to N).

An alternative voting strategy named *randomized voting* (Xu et al., 1990) reduces this complexity to $O(N)$. This technique considers N randomly chosen pairs of points in the image domain. Each pair of points in the image domain will vote for a single point in the Hough domain, namely the intersection of the straight lines corresponding to each of the two image points. The strength of the vote can be a function of the local orientation strengths of the two points, for instance the product, the sum, or the minimum. Alternatively, the vote can be weighted by the norm of the vector product of the corresponding line parameters. We choose the latter so that parallel lines contribute less.

The traditional Hough space for the sinusoid identification problem (Torres, 1992; Saito et al., 1999; Wang et al., 2007) has the three parameters (a, b, c) of a dip $z = ax + by + c$. The advantage of our proposal is that votes accumulate on a $2D$ space instead of a $3D$ one. This is much faster because a smaller number of votes is enough to fill-in the 2D space and to obtain stable and well-localized maxima. Intuitively, this strategy is aggregating the votes for all the heights in the given window; this is natural because the sinusoids are very often nearly parallel and would correspond to vertically aligned points in the 3D space.

The randomized voting strategy is detailed in Algorithm 3. This algorithm fills an image $H(a, b)$ with the Hough votes for each candidate pair (a, b). The table of votes is initialized to zero (step 1). Then N_{rand} pairs of points are randomly sampled on the image domain (steps 3 and 4) and the local orientations at these points are extracted (steps 5 and 6). Next, we find two straight lines ℓ_1,ℓ_2 that are compatible with these local orientations (steps 7 and 8). These lines are expressed in homogeneous coordinates which facilitates computing their intersection point as the cross product (step 9). The homogeneous coordinates of the intersection point are normalized to obtain its Euclidean coordinates (steps 10 and 11), which is identified to a site in the Hough domain. Then a weighted vote is cast for that site (steps 13 and 14).

Algorithm 3: `randomized-hough-voting`

input : Local orientations $G : [0, W]^2 \to \mathbf{R}^2$
input : Maximum values for a dip κ
input : Number of random trials N_{rand}
output : Hough votes $H : [-\kappa, \kappa]^2 \to \mathbf{R}$

1 $H \leftarrow 0$
2 **for** $n = 1, \ldots, N_{rand}$ **do**
3 $\quad p_1, q_1 \leftarrow$ `random-point`$([0, W]^2)$
4 $\quad p_2, q_2 \leftarrow$ `random-point`$([0, W]^2)$
5 $\quad u_1, v_1 \leftarrow G(p_1, q_1)$
6 $\quad u_2, v_2 \leftarrow G(p_2, q_2)$
7 $\quad \ell_1 \leftarrow \left(\sin\left(\frac{2\pi}{W} p_1\right), -\cos\left(\frac{2\pi}{W} p_1\right), u_1/v_1\right)$
8 $\quad \ell_2 \leftarrow \left(\sin\left(\frac{2\pi}{W} p_2\right), -\cos\left(\frac{2\pi}{W} p_2\right), u_2/v_2\right)$
9 $\quad x, y, z \leftarrow \ell_1 \times \ell_2$
10 $\quad a \leftarrow x/z$
11 $\quad b \leftarrow y/z$
12 $\quad$ **if** $(a, b) \in [-\kappa, \kappa]^2$ **then**
13 $\quad\quad H(a, b) \leftarrow H(a, b) + \|\ell_1 \times \ell_2\|$

Hough space discretization and smoothing

The Hough space is discretized as a $W \times W$ regular grid covering the square $[-\kappa, \kappa]^2$ of the (a, b)-plane. The range of values spanned by this grid is given by the maximum slope of a plane capable of being represented, usually $[-1, 1]^2$, which corresponds to 45°. In order to handle dips with larger slopes a multi-scale approach is applied, as detailed in the section *Multiscale dip picking*. The scales are vertically down-scaled versions of the original image by powers of two, called "octaves". In the o-th octave the (a, b)-plane extends to $[-2^o, 2^o]^2$.

We need to localize the dips very accurately in the (a, b) space, thus in practice we chose the same size $W \times W$ as the input window. The Hough space is used (Algorithm 1, step 7) to find the main dip direction on a window as the site (a, b) with the maximum number of votes. To improve the localization of this maximum, the Hough image $H(a, b)$ is smoothed with a Gaussian kernel of standard deviation η (which is a parameter of the method). Due to this smoothing, the actual resolution $W \times W$ of the Hough size is not a critical parameter as long as η is larger than the pixel size used to discretize H.

The *a contrario* theory

The *a contrario* theory (Desolneux et al., 2000, 2008) is a statistical framework used to set detection thresholds automatically in order to control the number of false detections. It is based on the non-accidentalness principle (Witkin and Tenenbaum, 1983; Lowe, 1985) which informally states that an observed structure is meaningful only when the relation between its parts is too regular to be the result of an accidental arrangements of independent parts. In the words of D. Lowe, "we need to determine the probability that each relation in the image could have arisen by accident, $P(a)$. Naturally, the smaller that this value is, the more likely the relation is to have a causal interpretation" (Lowe, 1985,p. 39).

A stochastic background model $\mathcal{H}_0$ needs to be defined, where the structure of interest is not present and can only arise as an accidental arrangement. For example, some geometrical feature detection methods such as line segments or elliptical arcs are based on the orientation of the image gradient (Grompone von Gioi et al., 2010; Pătrăucean et al., 2017). In such cases, the background model $\mathcal{H}_0$

assumes that the gradient orientations at each pixel are independent random variables, uniformly distributed in $[-\pi, \pi)$. Under this background model, a region of the image where the gradient orientation follows a regular structure would be a rare accident and is detected as such.

We also need to define a family of events of interest T. For feature detection the family of events is the set of all the geometrical events considered, i.e., all the line segments and elliptical arcs considered in the image domain. Then, we need to assess the accidentalness of a candidate feature. For example, if a line segment is present in an image, the gradient orientation at the corresponding position would be orthogonal to the line segment. Then, given a candidate line segment, one measures how well the image gradient corresponds to the candidate event, and evaluates the probability of observing such a good agreement by chance in the background model. A rough agreement could arise just by chance and thus does not correspond to an interesting event; conversely, a very good agreement would be rare and suggest the presence of a structure instead of just a lucky accident. In other words, when this probability is small enough, there exists evidence to reject the null hypothesis and declare the event meaningful. However, one needs to consider that multiple candidates are tested. If 100 tests were performed, for example, it would not be surprising to observe among them one event that appears with probability 0.01 under random conditions. The number of tests N_T needs to be included as a correction term, as it is done in the statistical multiple hypothesis testing framework (Gordon et al., 2007).

Following the *a contrario* methodology (Desolneux et al., 2000, 2008), we define the *number of false alarms* (NFA) of an event e observed up to an error $k(e)$ as:

$$\mathrm{NFA}(e) = N_T \cdot \mathbb{P}\Big[K_{\mathcal{H}_0}(e) \leq k(e)\Big], \tag{7}$$

where the right hand term is the probability of obtaining in the background model $\mathcal{H}_0$ an error $K_{\mathcal{H}_0}(e)$ smaller or equal to the observed one $k(e)$. The smaller the NFA, the more unlikely the event e is to be observed by chance in the background model $\mathcal{H}_0$; thus, the more meaningful. The *a contrario* approach prescribes accepting as valid detections the candidates with NFA $< \varepsilon$ for a predefined value ε. It can be shown (Desolneux et al., 2000, 2008) that under $\mathcal{H}_0$, the expected number of tests with NFA $< \varepsilon$ is bounded by ε. As a result, ε corresponds to the mean number of false detections under $\mathcal{H}_0$. In many practical applications, including the present one, the value $\varepsilon = 1$ is adopted. Indeed, it allows for less than one false detection on an image or on a set of images, which is usually quite tolerable.

A contrario dip detection

The maximum of the Hough transform gives the parameters (a, b) of the best bundle of sinusoids for the current analysis window. But we still need to determine if this detection is meaningful (in case we are observing a window without sinusoids) and then extract the individual dips. For this we propose to apply an *a contrario* validation (Desolneux et al., 2008). The test for validating individual dips is similar to the one used in Grompone von Gioi et al. (2010), but evaluating the image orientation along the sinusoid at each depth instead of along straight lines. For a given borehole image size, it selects a threshold on the number of pixels with gradient perpendicular to the sinusoid being tested.

Algorithms 4 and 5 describe the method. The main inputs are the image gradient at each point in the window of interest of size $W \times H$, and the parameters (a, b) of the sinusoid bundle to be validated. There are two parameters to be set: ρ and ε. The parameter ρ is a normalized angular tolerance. For example, a tolerance of 45° leads to $\rho = \frac{45}{180} = 0.25$. Here, the NFA corresponds to an upper bound of the expected number of dips, with at least as many correctly oriented gradients as the one currently considered, that will be observed by chance per window. The NFA provides a measure of the quality of the detection that is easier to interpret and to threshold than the quantity of well-aligned pixels. The lower the NFA, the less probable it corresponds to an accidental detection, thus the more probable it corresponds to a real dip. Automatic detections are made by keeping candidates with NFA $< \varepsilon$. The parameter ε corresponds to an upper bound to the number of false alarms that one can accept per window. The output of the algorithm is the set L of parameters of the validated sinusoids relative to the window being analyzed, see Figure 6.

The *a contrario* detection algorithm builds a list L of "accepted dips". The set L is initially empty (line 1 of Algorithm 5). Then, all the depths h in the range of the window will be tested one by one (line 2). For each depth, the number n of pixels along the sinusoid with defined values (i.e., belonging to the pad/flap traces) are counted. Among them, the number k of pixels with the image gradient orientation compatible with the sinusoid are also counted; both n and k are initialized to zero (lines 1 and 2 of Algorithm 4). To count these numbers, all the columns i of the image are evaluated (line 3) and for each one the corresponding coordinate $j(i)$ in the sinusoid at depth h is computed (line 4). Also, the vector $N(i)$ normal to the sinusoid at pixel $(i, j(i))$ is computed (line 5). If the borehole image is defined at $(i, j(i))$ as determined by a mask M (line 6), then the pixel provides information that may match or not the sinusoid; n is increased to count the point (line 7). If the angle between the normal to the sinusoid $N(i)$ and the actually observed image gradient $\nabla I_\sigma(i, j(i))$ is less than the tolerance $\rho \in (0, 1)$ (line 8), then the pixel is compatible and k is increased (line 9).

With n and k, the NFA can be computed (line 10) by:

$$\mathrm{NFA} = W^2 H \cdot B(n, k, \rho), \tag{8}$$

where $W \times W$ is the size of the discretization grid defined above (section "Hough space discretization"), H is the height of the observed window, and B is the tail of the

binomial distribution, defined as:

$$B(n,k,p) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \quad (9)$$

The binomial tail can be computed efficiently using the incomplete beta function.

Finally, if the NFA of the sinusoid being evaluated is smaller than ε (line 4 of Algorithm 5), the corresponding sinusoid (a, b, h) is added to the output set L (line 5). Figure 6 shows an example of the detected sinusoids.

Note that if there are no sinusoidal structures (compatible with a, b), then the previous algorithm should not detect any dip. If there are several candidate parameters (a, b), the algorithm is applied for each pair. If no dip is detected for any candidate plane orientation, then the window is declared to be without sinusoids.

Algorithm 4: `dip-nfa`

input : Image I_σ $(W \times H)$
input : Valid data mask M $(W \times H)$
input : The sinusoid parameters (a, b, h)
input : Number $0 < \rho < 1$
output : A positive number NFA

1 $n \leftarrow 0$ // total number of points
2 $k \leftarrow 0$ // number of well-aligned points
3 **for** $i = 0, \ldots, W-1$ **do** // (i,j)=graph of the sinusoid
4 $\quad j(i) \leftarrow a\cos\left(\frac{2\pi}{W}i\right) + b\sin\left(\frac{2\pi}{W}i\right) + h$
5 $\quad N(i) \leftarrow \left(-\frac{2\pi}{W}a\sin\left(\frac{2\pi}{W}i\right) + \frac{2\pi}{W}b\cos\left(\frac{2\pi}{W}i\right), 1\right)$
6 $\quad$ **if** M *is valid at* $(i, j(i))$ **then** // avoid pads/flaps missing data
7 $\quad\quad n \leftarrow n+1$
8 $\quad\quad$ **if** $\text{Angle}(N(i), \nabla I_\sigma(i, j(i))) < \rho\pi$ **then**
9 $\quad\quad\quad k \leftarrow k+1$
10 NFA $\leftarrow W^2 H \cdot B(n, k, \rho)$

Algorithm 5: `locate-dips`

input : Image I_σ $(W \times H)$
input : Valid data mask M $(W \times H)$
input : The sinusoid parameters (a, b)
input : Number $0 < \rho < 1$
input : Number $\varepsilon > 0$
output : A set of sinusoids L

1 $L \leftarrow \varnothing$
2 **for** *depth* $h = 0, \ldots, H-1$ **do**
3 $\quad$ NFA $\leftarrow$ `dip-nfa`$(I_\sigma, M, \rho, a, b, h)$
4 $\quad$ **if** NFA $< \varepsilon$ **then**
5 $\quad\quad L \leftarrow L \cup \{a, b, h\}$

Local dip refinement

The sinusoids detected by the previous step may not be very accurate because their (a, b) parameters are computed globally for the whole window (using the Hough transform). Indeed, the NFA test (Algorithm 4) reduces the set of candidate sinusoids in the current window by determining whether a sinusoid is meaningful or not. However, all of these sinusoids have the same parameters (a, b). The goal of the refinement step is to *slightly* adjust the three parameters of each sinusoid so that it better fits to the local geometry of the image. In what follows we propose a method to iteratively refine the parameters of each individual sinusoid by a simple discrete coordinate descent optimization (see Algorithm 6).

The refinement described in Algorithm 6 uses the NFA score as a measure of fit. It tests small variations of the parameters and selects the one yielding the smallest NFA value. This process is iterated until no improvement of the NFA can be achieved. Notice that the potentially infinite loop in this algorithm halts because the search space is finite (it is restricted to sinusoids that intersect the current window).

Algorithm 6: `refine-dip`

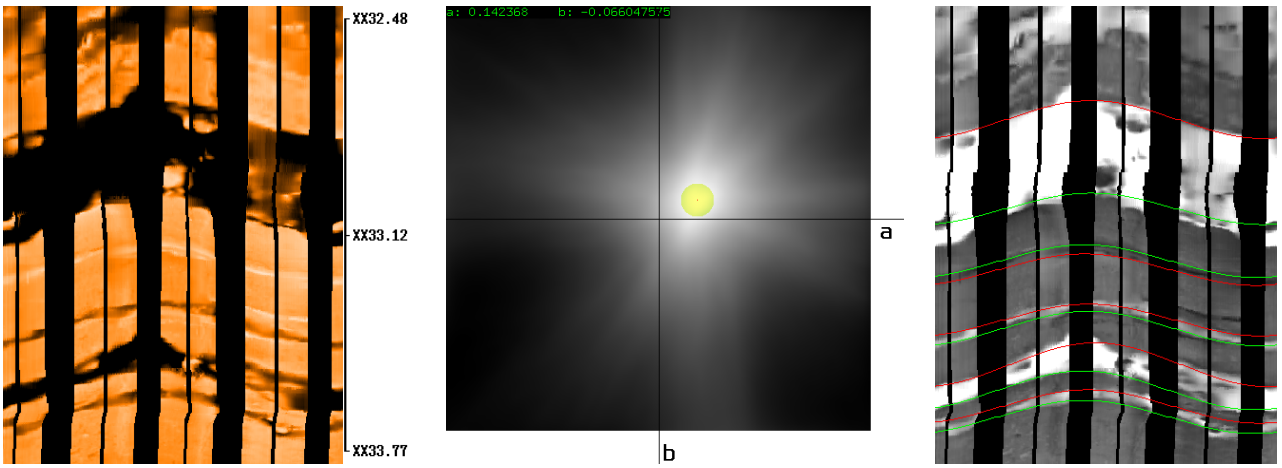
input : Image I_σ $(W \times H)$
input : Valid data mask M $(W \times H)$
input : Sinusoid to optimize $s = (a, b, h)$
output : A refined sinusoid s

1 **repeat**
2 $\quad N \leftarrow$ `neighbors`(s) // list of candidate neighbors positions
3 $\quad$ **for** $n \in N$ **do**
4 $\quad\quad$ **if** `dip-nfa`$(I_\sigma, M, n) <$ `dip-nfa`(I_σ, M, s) **then**
5 $\quad\quad\quad s \leftarrow n$ // select optimal neighbor
6 **until** s *does not change* // iterate until a local optimum is found

Redundancy reduction

After the local optimization step described in the previous section, we find that the output list of sinusoids is very redundant. There are two reasons for that. First, sinusoids that were very close may collapse to the same local minimum. Second, different local minima may be very close. To avoid these redundancies and to obtain a cleaner list of dips, we propose to apply an *exclusion criterion*. The criterion works by enforcing the following condition: each pixel in the input image is owned by at most one sinusoid, only the pixels owned by each sinusoid are used to compute its final NFA value.

This exclusion criterion is described by Algorithm 7. This is a greedy algorithm that traverses one time the list of all candidate sinusoids by increasing order of their NFA, from the most to the least meaningful detection (recall that the lowest NFA corresponds to the best match). The precise implementation of the algorithm depends on the function `exclude-around(G,s)`. This function finds



(a) **(b)** **(c)**

Figure 6: Example of dip picking results using the proposed method. **(a)** shows the input window. **(b)** shows the extracted Hough space with its detected maximum highlighted in yellow. **(c)** shows the validated detections highlighted (red and green denote opposed orientations of the sinusoid) over a grayscale version of the image.

the pixels around a thin band of fixed width around the sinusoid s, and marks them as *undefined* in the mask M. Thus, they will not be counted when updating the NFA value (see line 6 of Algorithm 4), just as if they were outside the image domain.

Algorithm 7: `refine-dips-with-exclusion`

input : Image I_σ $(W \times H)$
input : Valid data mask M $(W \times H)$
input : Number $\epsilon > 0$ (NFA threshold)
input : L list of sinusoids to be refined
output : L' list (possibly shorter) of refined sinusoids

1 $L' \leftarrow \varnothing$ // output accumulator
2 $L \leftarrow$ `sort-sinusoids-by-growing-nfa`(L, I_σ, M)
3 **for** $s \in L$ **do** // traverse sinusoids in order
4 $s' \leftarrow$ `refine-dip`(I_σ, M, s) // refine this sinusoid
5 **if** `dip-nfa`$(I_\sigma, M, s') < \epsilon$ **then** // if meaningful
6 $L' \leftarrow L' \cup \{s'\}$ // add it to list
7 $M \leftarrow$ `exclude-around`(M, s') // and mark neighboring pixels as undef.

Parameters

Table 1 summarizes the meaning and reasonable values for all the parameters of the proposed method.

The most important parameter is μ of the structure tensor filtering. Increasing μ, the precision of the detection increases, but the localization becomes worse; if it is too large (at around $\mu = 20$) it starts losing structures. In practice, μ is always set to 11.

The second most important parameter is σ. This is the resolution at which the local orientations are computed.

Table 1: Summary of the meaning and reasonable values for all the parameters of the proposed method.

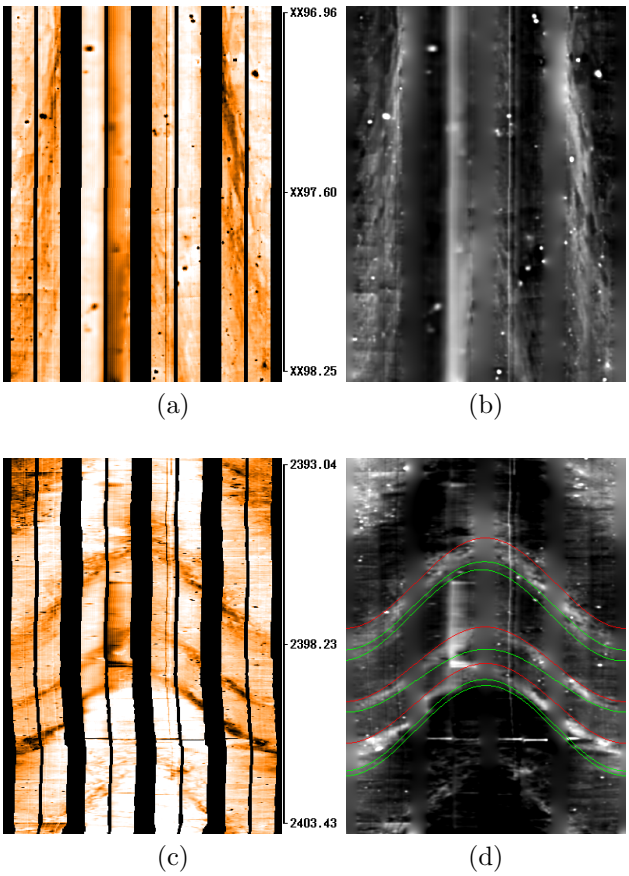
parameter	default value	meaning
σ	1.0	pre-filtering (dequantization)
μ	11.0	structure tensor filtering
κ	1.0	maximum values for a dip (a, b)
η	30.0	blur of the Hough space
ρ	0.25	normalized angular tolerance
ϵ	1.0	number of false alarms
N_{rand}	10^6	number of random trials

Setting it larger than 1.0 removes a lot of noise, aliasing and artifacts in the input images, but then small structures may be lost. Although it could be used to control the level of detail in the image, in practice it is always set to 1.0.

The parameter κ selects how much of the Hough space we want to consider. A value of $\kappa = 1.0$ allows for dips at a slope of 45°. If higher slopes are required, it is better to use a higher level of the pyramid than to increase κ. Decreasing κ may improve the precision of near-horizontal dips, at the price of ignoring high slopes. In practice, κ is always set to 1.0.

The value of the parameter η is not very important, provided it should be large enough to allow a rough estimation of the maximum in the Hough transform. In practice, η is always set to 30.0.

The value ρ of the validation step controls the balance between being tolerant to imprecise sinusoids at the cost of requiring larger observed parts of the sinusoids. The value 0.25 results in a good balance. Even if in some images a slightly smaller or slightly large value improves a little the result, in practice it is always set to $\rho = 0.25$.



(a) (b)

(c) (d)

Figure 7: Results of the multiscale analysis. The same part of the well is analyzed at octaves 0 (original resolution, input **(a)** and output **(b)**) and 3 (vertical zoom-out of factor 8, input **(c)** and output **(d)**). Notice that some very "tall" sinusoids only become visible at high enough scales. Depths are in meters.

The value of the parameter ϵ is not very important and does not have an important effect in the final result. In practice, ϵ is always set to 1.0.

The parameter N_{rand} affects linearly the running time of the algorithm. It should be as large as possible while being practical. In practice $N_{rand} = 10^6$ provides a good balance between quality of the result and computation time.

MULTISCALE DIP PICKING

The basic data structure for exploring such large images is the multi-scale pyramid. This data structure is straightforward: it is the union of several vertically down-scaled versions of the original image by factors that are successive powers of two, called "octaves". Notice that this notion of octave is different from the one commonly used in image processing: here we only perform a zoom-out in the vertical direction, thus all the octaves preserve the image width.

As observed in Figure 7, when the borehole direction is nearly parallel to the geological strata, the image contains nearly vertical structures. These patterns are actually sinusoids of very large amplitude, as can be seen by looking at the higher octaves. Our algorithm correctly finds these sinusoids once they appear entirely inside the selected region of interest.

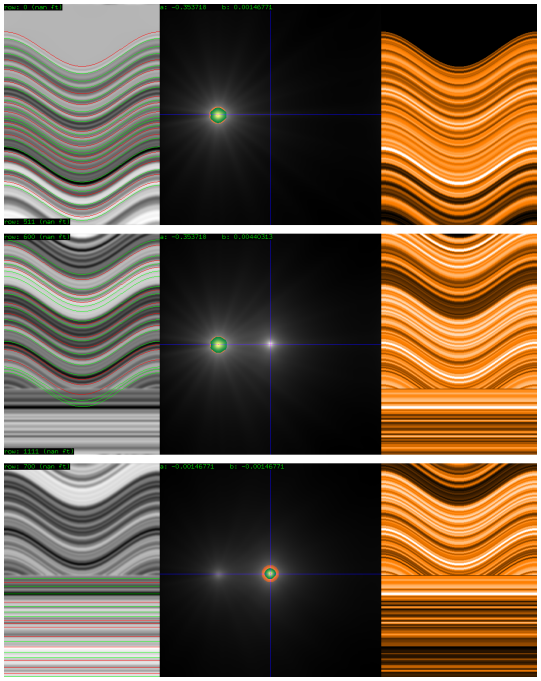


Figure 8: *A contrario* dip picking (with local refinement) on a synthetic image. In the left column the detections are highlighted (red for dark-to-bright and green for bright-to-dark detections) over a grayscale version of the image which is depicted on the right column in false color (the resistivity is single channel). The center column illustrates the smoothed Hough voting space, with the location of the maximum highlighted with a green dot. The three rows correspond to different dip configurations. In the second and the third row we see two dips in the same analysis window.

Automatic dip picking

The proposed algorithm can be used to process a whole borehole automatically. Two parameters need to be specified: a window size ω and the number of spatial scales O. The borehole image is processed at full resolution and at $O-1$ consecutive dyadic spatial scales. At each scale, the image is processed by windows of height ω, advancing by half-window steps of $\omega/2$. All the dips found at different windows and scales are put together. This can generate repeated dip detections, to filter them we apply the exclusion criterion described above.

In Figure 8 we show a typical run of our algorithm on three windows of a very well behaved synthetic image. In the first window, the correct dip is detected as the position of the maximum in Hough space. In the second window, when there is more than one dip (e.g., near

a "cross-bedding"), there are two competing maxima in Hough space, and the one corresponding to the dip with largest area on the region of interest is chosen. In the last window, the second maximum becomes larger and the other dip is detected. It is the task of the exclusion criterion to merge the whole set of dips without redundancy.

Computational cost

The first steps of the method all have a computational cost proportional to the number of pixels in the image or window being processed. This includes the extraction of the window, the filling-in of missing data, the Gaussian filtering and the computation of local orientations. The randomized Hough transform step has a computational cost proportional to the number of random trials N_{rand}. The *a contrario* validation and refinement step have a computational cost proportional to the number of candidates being evaluated, thus proportional to the number of windows evaluated. All in all, the computational complexity is $O(N)$, where N is the number of pixels in the borehole image.

In practice, the randomized Hough transform is very effective and the algorithm computes the result in real-time when applied to windows in a semi-supervised human/computer interactive interface. When applied automatically, a non-optimized code can process a whole borehole of 200,000 pixel rows, corresponding to about 1450 m (4800 feet), in less than 30 s on a modern laptop computer.

EXPERIMENTS

In this section we present experiments for evaluating the performance of the proposed algorithm. First we will carry out a quantitative evaluation comparing against manually annotated dips on three borehole images. Then we will perform an ablation study to analyze the impact of some components of the proposed algorithm. Lastly, we present a qualitative analysis of the results and discuss its limitations.

Quantitative evaluation

To evaluate the proposed method we shall compare against three manually annotated borehole images. These datasets (denoted A, B and C) are described below:

- A: 221 manually annotated dips along 30 m (97 ft) of borehole, from 4078 m to 4107 m (13380 to 13476 ft);
- B: 3766 manually annotated dips along 1485 m (4873 ft) of borehole, from 3110 m to 4596 m (10205 to 15079 ft);
- C: 194 manually annotated dips along 3345 m (1099 ft) of borehole, from 3037 m to 3372 m (9964 to 11063 ft).

The proposed multiscale algorithm was applied to the three borehole images using five octave scales and default parameters for the detection. This yielded a list of detections each with an associated NFA (which can be seen as a confidence measure).

To compare the manual annotations and our results we analyzed the dips on a rolling window along the well. We set the size of the rolling window to 6 m (20 ft) for the second and third boreholes (B and C), while for the first one (A) we used 0.6 m (2 ft) because the density of events is much higher for such a short borehole. The results can be seen in Figure 9. We computed and show the density of dips within each window, the dominant azimuth angle (computed as the mode of the azimuths), and the mean of the dip inclination. From the plots we can observe that the proposed method produces meaningful dips with parameters close to the manual annotations. As manual labeling is not exhaustive, it is to be expected that the proposed method can produce many more dip detections.

To further assess the pertinence of the detected dips we evaluated the error with respect to the manually annotated ones. For each manually annotated dip, we searched for the closest detected dip in terms of *root mean squared error* (RMSE) between the two graphs

$$d(h_1, h_2) = \sqrt{\frac{1}{W} \sum_{w=1,\dots,W} |h_1(w) - h_2(w)|^2}. \qquad (10)$$

Note that this error has distance units and can be used to set a threshold to identify false-positive detections. Here we considered all the correspondences to analyze the differences in the parameters of the two dips. We computed the difference of depth, azimuth, and inclination of the matched dips. The distributions of these errors for each one of the annotated borehole images are shown in Figure 10. We observe that all the histograms of errors are concentrated around zero. This means that close to most of the manual annotations, there was a detection with similar parameters. In the borehole B we observe a higher error, this is likely due to higher density and variability of events present in this borehole.

Ablation study

We performed three simple "ablation" experiments to assess the validity of the method by comparing its performance to manual dip annotations by expert geologists. Unfortunately, we cannot compare the performance of the proposed method to other publicly available methods, as we could not find any.

More precisely, we studied the impact of three crucial steps of the pipeline: the gradient dequantization preprocess (controlled by the parameter σ), the multi-scale selection scheme (controlled by the parameters σ), and the local dip refinement (implemented in Algorithm 6).

For that, we will define a *true positive* measure based on the RMSE error between matched dips of equation (10). This error measure has distance units: the distance between two sinusoids is defined as the average distance of the corresponding ellipses on the boundary of the cylin-

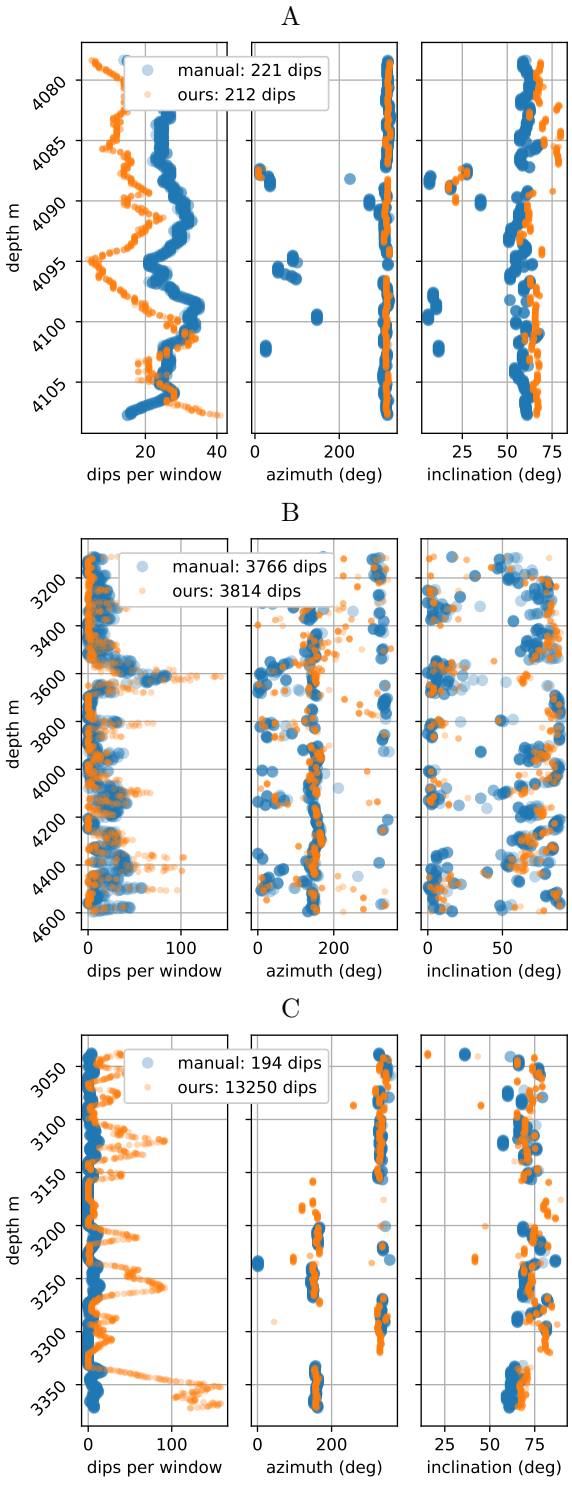


Figure 9: Analysis of dips on a sliding window. The three rows correspond to the three evaluation boreholes A, B and C with manual annotation. The plots in each column show: the number of dips on the rolling window, the dominant azimuth angle (which is computed as the the mode of the azimuths in the window), and the mean inclination within the window.

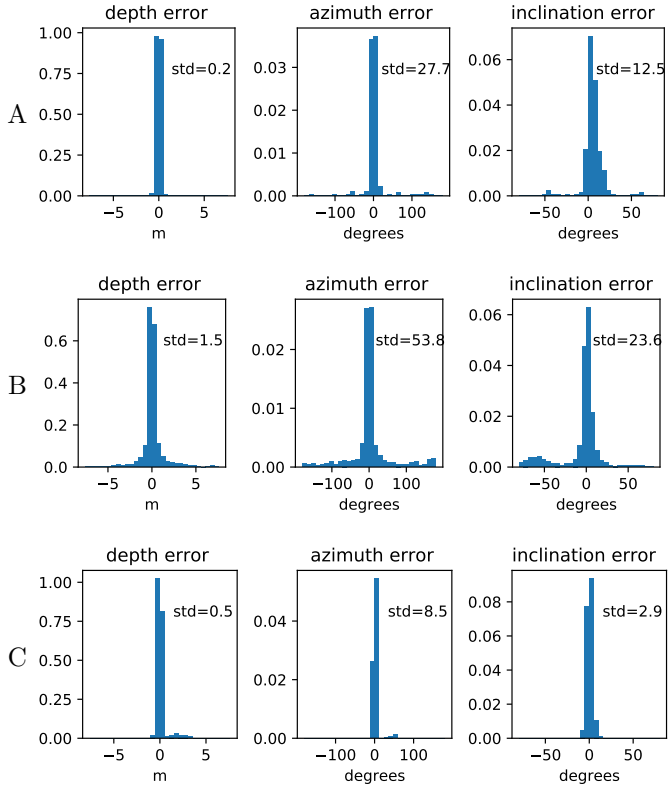


Figure 10: Analysis of errors in the detected dips. For each manually annotated dip, we matched it with the closest automatically detected dip. Then we analyzed the difference of depth, azimuth, and inclination of the matched dips. The three rows correspond to the three annotated boreholes (denoted A, B and C). The standard deviation of the errors are also shown in the plot.

der. We will say that a manually annotated dip is correctly found by an algorithm if the error with the matched dip is smaller than 30.48 cm (1 feet). Note that we cannot easily define a measure for false positives because the manually annotated dips are not exhaustive (large swaths of the image were not annotated).

This true positive score allows us to evaluate the performance of variants of the proposed algorithm obtained by removing different components. In Table 2 we present the results of these experiments. We consider reducing the number of scales (using only the first octave, the first two, or all the 5 octaves), removing the dequantization blur ($\sigma = 0$ for removing it), and removing the iterative local dip refinement (by setting the refinement iterations to 0 and 3). All these variants lead to a reduced performance, most notably removing the multiscale and the iterative refinement.

Qualitative evaluation and discussion

Let us now see results of our algorithm on some examples (see Figures 11 and 12). In all cases, the initial dip (a, b)

scales	dequant. σ	ref. iter.	A	B	C
all	1	100	**89%**	**59%**	**88%**
1 oct.	1	100	28%	34%	28%
2 oct.	1	100	81%	48%	76%
all	0	100	84%	52%	85%
all	1	3	86%	56%	85%
all	1	0	72%	45%	71%

Table 2: Ablation study. True positive rate for different variant of the algorithm evaluated on the three annotated borehole images A, B and C. The first three columns correspond to options of the proposed dip-picking algorithm: number of octaves considered in the multiscale scheme, dequantization parameter σ, and number of iteration of local dip refinement.

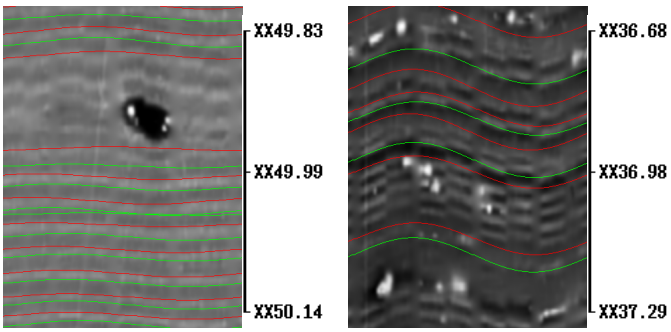
was computed by the Hough transform algorithm (line 6 of Algorithm 1), and the parameters (a, b) were used as input for the NFA computation. Notice that in most cases the correct dip is obtained, even for fragmented sinusoid as seen in Figure 11

The sequence of detections on Figure 12 shows the behavior of the algorithm around a fracture and a cross-bedding. Notice that, for each position of the sliding window, a single dip is selected by the Hough transform, the one that is present on the largest part of the window. This is the slope (a, b) that is given to the *a contrario* refining step, which determines the set of depths c where sinusoids are detected. For this particular example, the detections flip from one side to the other of the cross-bedding, which happens between the second and third images in the sequence.

Limitations

The proposed method is far from perfect and there is room for improvement. This can be seen in the quantitative evaluation were the performance on the dataset B is below the other two. This borehole is hard as it presents many events, this can be seen in the annotation in Figure 9, see in particular at depth 3627 m (11900 ft).

One can see some examples of wrong detections in Figure 12. There is an erroneous dip by the lower part of the second image. In this case, the Hough step proposed the wrong candidate due to the cross-bedding. Then, the refining step did not explore enough variations to fit precisely the actual dip. In addition, due to the blurred aspect of the edges, the gradient orientation roughly match the wrong dip, producing a barely meaningful detection. In most cases, this problem is solved by the exclusion criterion when all the dips at all windows and scales are put together; indeed, the dips detected in the third image for the same part of the image have better NFA value and will be preferred. But there are still cases where this strategy fails. A similar example of wrong detection, although less extreme, can be seen in the middle of the second image.



(a) (b)

Figure 11: *A contrario* dip picking on two challenging zones, one of them **(a)** with low contrast, the other **(b)** suffering from pad misalignment. In both cases correct dips are being identified, with just a couple of false alarms on the first one. Depths are in meters.

CONCLUSIONS

We have proposed a method to perform fast and automatic dip picking on borehole images, amenable to an embedded online implementation due to its lower computational requirements. The method is based on an *a contrario* criterion to control the number of false detections, coupled with a search space optimization based on Hough spaces. Our experiments show that the algorithm detects correctly the evolution of the dip along the borehole depth, and gives no dip when there are no apparent structures in the data. A quantitative evaluation on annotated borehole images shows that the proposed method can detect from 60% to 90% of the dips annotated by an expert in a completely unsupervised way. The same method can also be integrated in an interactive dip annotation tool to assist the annotation.

DATA AND MATERIALS AVAILABILITY

The dataset of three manually annotated borehole images and the supplementary material associated to this paper are available at `https://cmla.github.io/dip-picker/`.

ACKNOWLEDGEMENT

We want to thank Jean-Michel Ghidaglia for his warm encouragement towards publishing this work. We are also thankful for the support and fruitful discussions with Schlumberger experts: Isabelle Le Nir, Carlos Maeso, Tetsushi Yamada, Nicolas Veron, Alexis He, and Kristina Prokopetc. This project was carried out in the framework of the French CITEPH (EVOLEN) collaborative program under reference CITEPH-40B-2014. This work contains information provided by the Oil and Gas Authority and/or other third parties. We also thank the editors and the three anonymous reviewers for the valuable comments and suggestions.

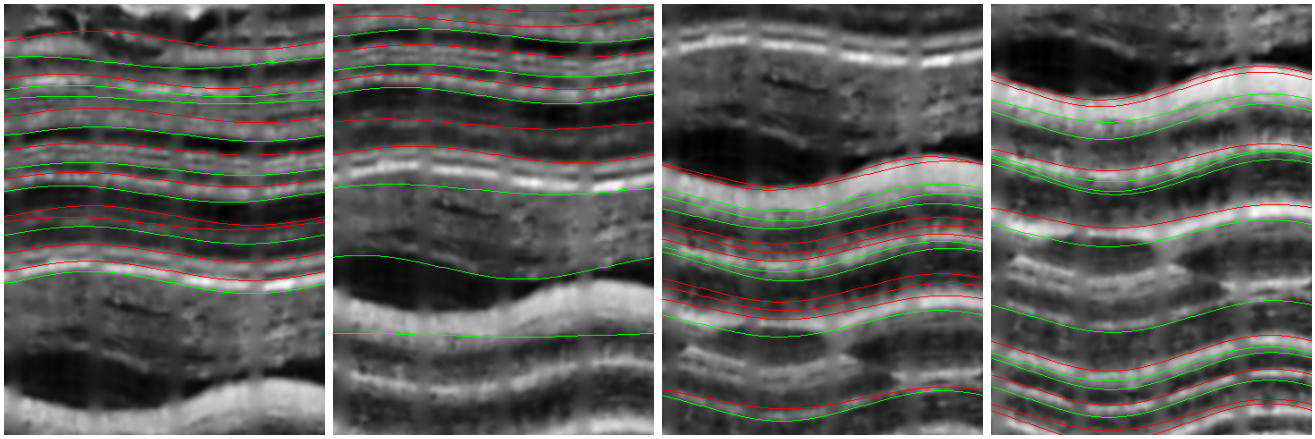


Figure 12: *A contrario* dip picking, on a sliding window traversing a change of inclination (from column 2 to 3) and a cross-bedding with no change of inclination (from column 3 to 4).

REFERENCES

Assous, S., P. Elkington, S. Clark, and J. Whetton, 2014, Automated detection of planar geologic features in borehole images: Geophysics, **79**, no. 1, D11–D19.

Desolneux, A., S. Ladjal, L. Moisan, and J.-M. Morel, 2002, Dequantizing image orientation: IEEE Transactions on Image Processing, **11**, 1129–1140.

Desolneux, A., L. Moisan, and J.-M. Morel, 2000, Meaningful alignments: International Journal of Computer Vision, **40**, 7–23.

———, 2008, From gestalt theory to image analysis: a probabilistic approach: Springer.

Duda, R. O., and P. E. Hart, 1972, Use of the hough transformation to detect lines and curves in pictures: Communications of the Association for Computing Machinery, **15**, 11–15.

Förstner, W., and E. Gülch, 1987, A fast operator for detection and precise location of distinct points, corners and centres of circular features: Proceedings of the ISPRS intercommission conference on fast processing of photogrammetric data, 281–305.

Gonzalez, A., L. Kanyan, Z. Heidari, and O. Lopez, 2020, Integrated multiphysics workflow for automatic rock classification and formation evaluation using multiscale image analysis and conventional well logs: Society of Petrophysicists and Well-Log Analysts.

Gordon, A., G. Glazko, X. Qiu, and A. Yakovlev, 2007, Control of the mean number of false discoveries, Bonferroni and stability of multiple testing: The Annals of Applied Statistics, **1**, 179–190.

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

Hall, M., M. Ponzi, and G. Maletti, 1996, Automatic extraction and characterisation of geological features and textures from borehole images and core photographs: SPWLA 37th Annual Logging Symposium.

Hough, P., 1962, Method and means for recognizing complex patterns. (US Patent 3,069,654).

Illingworth, J., and J. Kittler, 1988, A survey of the hough transform: Computer Vision, Graphics, and Image Processing, **44**, 87 – 116.

Jiang, J., R. Xu, S. C. James, and C. Xu, 2020, Deep-learning-based vuggy facies identification from borehole images: Society of Petroleum Engineers.

Kherroubi, J., C. Maeso, Y. Wang, H. Gamero-Diaz, et al., 2016, Lamination analysis from electrical borehole images: A quantitative workflow: Presented at the SPWLA 57th Annual Logging Symposium, Society of Petrophysicists and Well-Log Analysts.

Lofts, J. C., and L. T. Bourke, 1999, The recognition of artefacts from acoustic and resistivity borehole imaging devices, *in* Borehole Imaging: applications and case hisotries: Geological Society, London, Special Publications.

Lowe, D., 1985, Perceptual organization and visual recognition: Kluwer Academic Publishers.

Lowell, M. A., G. Williamson, and P. K. Harvey, eds., 1999, Borehole imaging: applications and case hisotries: Geological Society, London, Special Publications.

Marr, D., and E. Hildreth, 1980, Theory of edge detection: Proceedings of the Royal Society of London B: Biological Sciences, **207**, 187–217.

Pătrăucean, V., P. Gurdjos, and R. Grompone von Gioi, 2017, Joint a contrario ellipse and line detection: IEEE transactions on pattern analysis and machine intelligence, **39**, 788–802.

Saito, N., N. Bennett, and R. Burridge, 1999, Method of determining dips and azimuths of fractures from borehole images. (US Patent 5,960,371).

Sun, C., and S. Pallottino, 2003, Circular shortest path in images: Pattern Recognition, **36**, 709–719.

Torres, D., 1992, Method for determining dip and strike angles in borehole ultrasonic scanning tool data. (US Patent 5,162,994).

Trice, R., 1999, A methodology for applying a non-unique morphological classification to sine wave events picked

from borehole image log data, *in* Borehole Imaging: applications and case hisotries: Geological Society, London, Special Publications.

van Ginkel, M., M. Kraaijveld, L. van Vliet, E. Reding, P. Verbeek, and H. Lammers, 2003, Robust curve detection using a radon transform in orientation space: Lecture Notes in Computer Science: Proceedings of the 13th Scandinavian Conference on Image Analysis, **2479**, 125–132.

Wang, Y., A. Etchecopar, H. Onda, and G. Mathieu, 2007, Method of determining planar events from borehole or core images. (US Patent 7,236,887).

Wedge, D., E. Holden, M. Dentith, and N. Spadaccini, 2015, Automated structure detection and analysis in televiewer images: ASEG Extended Abstracts, **1**, 1–4.

Weickert, J., and M. Welk, 2006, Visualization and processing of tensor fields: Springer Berlin Heidelberg.

Witkin, A. P., and J. M. Tenenbaum, 1983, On the role of structure in vision, *in* Human and Machine Vision: Academic Press, 481–543.

Xu, L., E. Oja, and P. Kultanen, 1990, A new curve detection method: Randomized hough transform (rht): Pattern Recognition Letters, **11**, 331 – 338.

Zhang, T.-F., P. Tilke, E. Dupont, L.-C. Zhu, L. Liang, and W. Bailey, 2019, Generating geologically realistic 3d reservoir facies models using deep learning of sedimentary architecture with generative adversarial networks: Petroleum Science, **16**, 541–549.