

Measure of Circularity for Parts of Digital Boundaries and its Fast Computation[★]

Tristan Roussillon¹

Université de Lyon,

Université Lyon 2, LIRIS, UMR5205, F-69676, FRANCE

Isabelle Sivignon

Université de Lyon, CNRS

Université Lyon 1, LIRIS, UMR5205, F-69622, FRANCE

Laure Tougne

Université de Lyon,

Université Lyon 2, LIRIS, UMR5205, F-69676, FRANCE

Abstract

This paper focuses on the design of an effective method that computes the measure of circularity of a part of a digital boundary. An existing circularity measure of a set of discrete points, which is used in computational metrology, is extended to the case of parts of digital boundaries. From a single digital boundary, two sets of points are extracted so that the circularity measure computed from these sets is representative of the circularity of the digital boundary. Therefore, the computation consists of two steps. First, the inner and outer sets of points are extracted from the input part of a digital boundary using digital geometry tools. Next, the circularity measure of these

sets is computed using classical tools of computational geometry. It is proved that the algorithm is linear in time in the case of convex parts thanks to the specificity of digital data, and is in $\mathcal{O}(n \log n)$ otherwise. Experiments done on synthetic and real images illustrate the interest of the properties of the circularity measure.

Key words: circularity, compactness, digital circle, discrete geometry, computational geometry

1 Introduction

Accurately locate circles and accurately measure deviation with a circular template are common problems in many fields of science and engineering. The fields of application are as diverse as geology [1], archeology [2], computer vision such as raster-to-vector conversion [3] or video processing [4], computational metrology to test the quality of manufactured parts [5–12], image processing and discrete geometry to recognize digital circles [13–20].

This paper focuses on the design of an effective method that computes the measure of circularity of a part of a digital boundary previously extracted from a digital image. The circularity measure of a given part of a digital boundary is a quantity that increases with deviation from a piece of digital circle, called a digital arc. The reader may find in the literature terms as diverse as compactness [21,13], roundness [22,6,8–11], out-of-roundness [5,6,23], but we prefer “circularity” [24,7] because it recalls the template with which the data are compared to, that is the circle.

* Work partially supported by the GEODIB ANR project (ANR-06-BLAN-0225)
Email address: `tristan.roussillon@univ-lyon2.fr` (Tristan Roussillon).

¹ Author supported by a grant from the DGA

16 Although plenty of papers present methods for assessing the circularity of a set
17 of points, as far as we know, only one paper dealt with the circularity of digital
18 boundaries, more than twenty years ago. In [13], a digital disk recognition
19 algorithm in $\mathcal{O}(n^2)$ is presented in the first part, and a digital compactness
20 evaluation algorithm for digital convex objects in $\mathcal{O}(n^3\sqrt{n})$ is presented in
21 the second part (where n is the number of pixels of the digital boundary).
22 The digital compactness measure is defined as the ratio between area A of the
23 shape and area A' of the smallest enclosing digital disk (where “the smallest”
24 is expressed in area unit, that is in number of pixels). As a smallest enclosing
25 digital disk may not be unique and as the smallest enclosing euclidean disk
26 may not be a smallest enclosing digital disk, areas of many digital disks have
27 to be compared. This is why the computational cost is rather high. This first
28 attempt shows that the problem is not trivial.

29 Moreover, naive methods that consist to find an easy-to-compute point that is
30 expected to be the centre of a circle separating the shape from the background
31 are only approximative. For instance, in [25], the barycentre of a set of pixels
32 is assumed to be the centre of a separating circle, but Fig. 1 shows that if the
33 barycentre of a set of pixels is computed, pixels that do not belong to the set
34 may be closer to the barycentre than pixels that belong to the set, even if it
35 turns out that the set of pixels can be separated from the pixels that do not
36 belong to the set.

37 A well-known circularity measure in the Euclidean plane is $4\pi A/P^2$ where A
38 is the area and P the perimeter. The digital equivalent of this circularity mea-
39 sure was introduced by [21], but even with a convergent perimeter estimation
40 based on digital straight segment recognition (see [26] and [27]) the measure is
41 theoretically unsatisfactory: digital circles may have different values that are

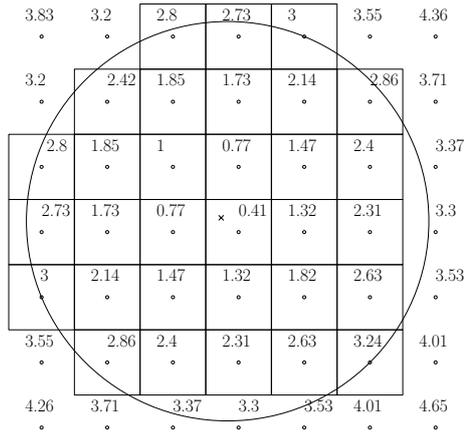


Fig. 1. A digital disk is depicted with pixels. In each pixel, the distance of its centre to the barycentre of the digital disk (located with a cross) is written. Some pixels that do not belong to the disk are closer (3.2) to the barycentre than some pixels that belong to the disk (3.24)

42 always strictly less than 1. Moreover, this kind of measure has several other
 43 drawbacks in practice: (i) it is not perfectly scale invariant, (ii) it is not easy
 44 to interpret (iii) it is not computable on parts of a digital boundary and (iv) it
 45 is not able to provide the parameters of a circle that is close to the data. This
 46 measure may be used for a coarse and quick approximation of the circularity
 47 of a digital boundary, but in the general case, another measure is needed.

48 Three kinds of methods may be found in the literature:

- 49 (1) Methods based on the circular Hough transform [28–30] allow extraction,
 50 detection and recognition of digital arcs. Even if these methods are ro-
 51 bust against shape distortions, noise and occlusions, they require massive
 52 computations and memory, and thresholds tuning. As the digital bound-
 53 ary is assumed to be extracted from the digital image in this paper, the
 54 following methods are more appropriate.
- 55 (2) Methods based on the separating circle problem in discrete and com-

56 computational geometry [14–20] allow the recognition of digital arcs. These
57 algorithms are not robust since one point can forbid the recognition of
58 a digital arc. They need to be modified to measure the extent of the
59 deviation with a digital arc.

60 (3) Methods based on circle fitting are widely used. In computer vision [31–
61 33,10,3,4], a circle is fitted to a set of pixels with the least square norm.
62 In computational metrology [5,22,6,7,23,12], a circle is fitted to a set of
63 points sampled on the boundary of a manufactured part by a Coordinate
64 Measurement Machine (CMM) generally with the least L_∞ norm (or
65 Chebyshev or MinMax norm) because it is recommended by the American
66 National Standards Institute (ANSI standard, B89.3.1-1972, R2002), but
67 sometimes with the least square norm, like in [34].

68 In this paper, a preliminary work presented in [35] is extended. Given a part
69 of a digital boundary, the objective is to compute a circularity measure ful-
70 filling some properties that will be enumerated in Section 2.2, as well as the
71 parameters of one separating circle if it is a digital arc or the parameters of
72 the closest circle otherwise. The proposed method is original because it is ap-
73 plied on digital boundaries like in [13] and it links both methods based on the
74 separating circle problem and methods based on circle fitting.

75 We formally define a circularity measure for parts of digital boundaries in
76 Section 2. From one digital boundary, two sets of points are extracted so that
77 the circularity measure computed from these sets is representative of the cir-
78 cularity of the digital boundary. Thanks to this trick, in spite of the specificity
79 of the digital boundaries, an algorithm that only uses classical tools of com-
80 putational geometry is derived in Section 3. Moreover, we show in Section 4
81 that the size of the two sets of points can be reduced in order to decrease the

82 burden of the computation. Some experiments are done on synthetic digital
83 boundaries and on real-world digital images in Section 5. The paper ends with
84 some concluding words and future works in Section 6.

85 **2 Circularity measure for parts of digital boundary**

86 *2.1 Data*

87 A binary image I is viewed as a subset of points of \mathbb{Z}^2 that are located inside
88 a rectangle of size $M \times N$. A digital object $O \in I$ is a 4-connected subset of
89 \mathbb{Z}^2 without hole (Fig. 2.a). Its complementary set $\bar{O} = I \setminus O$ is the so-called
90 background. The digital boundary C (resp. \bar{C}) of O (resp. \bar{O}) is defined as the
91 8-connected list of digital points having at least one 4-neighbour in \bar{O} (resp.
92 O) (Fig. 2.b). Without loss of generality, let us suppose that C is clockwise
93 oriented. Each point of C is numbered according to its position in the list. The
94 starting point, which is arbitrarily chosen, is denoted by C_0 . The last point is
95 denoted by C_{n-1} , where n is equal to the number of points in C . A connected
96 part C_{ij} of C is the list of digital points from the i -th point to the j -th point
97 of C (Fig. 2.c).

98 A digital disk is defined as a digital object whose points are separable from the
99 background by an Euclidean circle [13] (Fig. 2.d). A digital circle is defined as
100 the boundary of a digital disk (Fig. 2.e) and a connected part of it is defined
101 as a digital arc (Fig. 2.f).

102 The goal of the following subsection is to define a measure of how much a
103 given part of digital boundary is far from a digital arc.

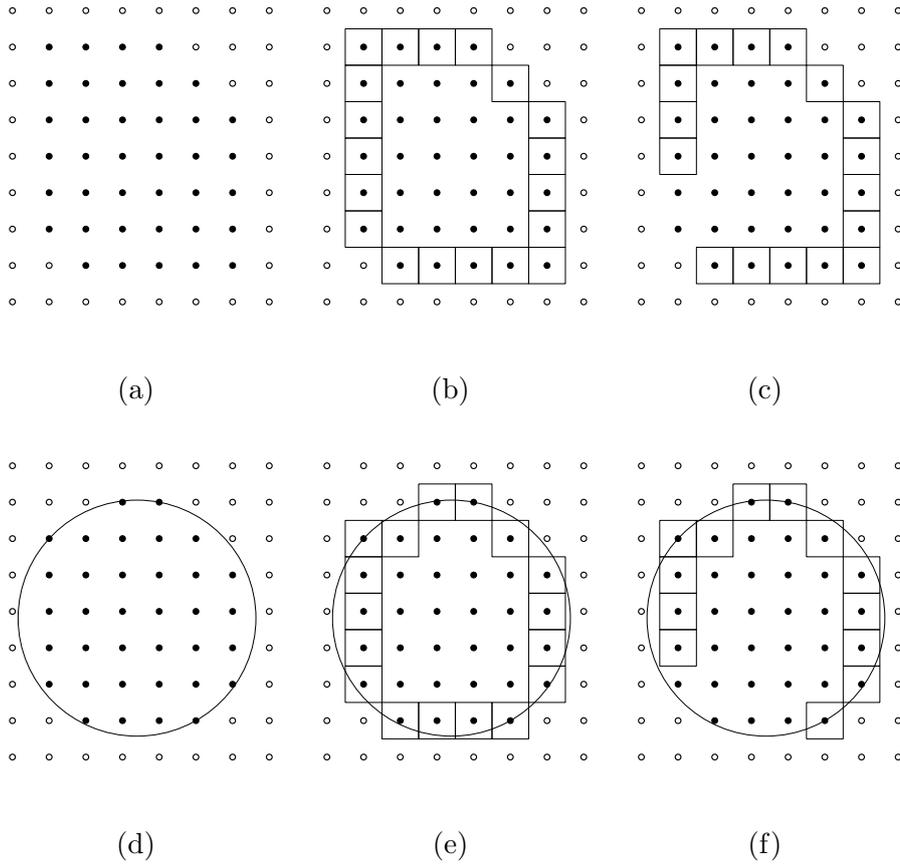


Fig. 2. (a) A digital object is depicted with black disks. The set of squares depicts the whole (b) or a part of the (c) digital boundary. (d) A digital object that is a digital disk. (e) A digital boundary that is a digital circle. (f) A part of a digital boundary that is a digital arc.

104 *2.2 Circularity measure of a part of a digital boundary*

105 A circularity measure for parts of digital boundaries is naturally expected to
 106 fulfil the following properties:

- 107 (1) be robust to translation, rotation, scaling.
- 108 (2) range from 0 to 1, equal 1 for a digital arc.
- 109 (3) be intuitive. For instance, it is naturally expected to increase as the num-
 110 ber of sides of regular polygons increases or as the eccentricity of ellipses

	least square norm	Chebyshev norm
radial distances	mean square error [31,33,34]	minimum width annulus [5,22,6,7,9,10,12]
areal distances	modified mean square error [32]	minimum area annulus [5,23]

Table 1

Some references for the four most used instances of the problem of fitting a circle to a set of points

111 decreases or as the amount of noise decreases. It is also expected that the
112 measure is robust: for example, the measure of a noisy digital circle has
113 to be higher than the measure of a digital triangle or a digital square, if
114 the noise is limited and does not affect the form.

115 In metrology, the circularity of an arbitrary set of points in the plane is defined
116 from the minimum cost of fitting a circle to the set given a certain norm. The
117 most often used norm is either L_2 (least square norm) or L_∞ (MinMax or
118 Chebyshev norm). Moreover, for both norms, the quantity that is minimized
119 is either the sum of the radial distances or the sum of the areal distances. The
120 four instances of the problem of fitting a circle to a set of points have been
121 thoroughly studied for a long time as it is shown in Table 1.

122 Fitting a circle to the points of a digital boundary with any of the above
123 techniques does not lead to a satisfactory measure, because property 2 does
124 not hold.

125 In the aim of fulfilling property 2, two sets of points, denoted by \mathcal{S} and \mathcal{T} , are

126 extracted from the digital boundary, so that: (i) $\mathcal{S} \subseteq O$, (ii) $\mathcal{T} \subseteq \bar{O}$, (iii) \mathcal{S}
 127 and \mathcal{T} are separable by a circle if and only if C is a digital circle.

128 Let $\mathcal{S} = C$ and $\mathcal{T} = \bar{C}$. According to the definitions introduced in Section 2.1),
 129 the three previous criteria are obviously fulfilled.

130 Let the minimum signed area annulus \mathbf{A} of centre ω , inner radius r_1 and outer
 131 radius r_2 be such that the outer disk contains all the points of \mathcal{S} and the inner
 132 disk does not contain any point of \mathcal{T} :

Find \mathbf{A} that minimizes $(r_2^2 - r_1^2)$

subject to

$$\begin{cases} \forall S \in \mathcal{S}, (S_x - \omega_x)^2 + (S_y - \omega_y)^2 \leq r_2^2 \\ \forall T \in \mathcal{T}, (T_x - \omega_x)^2 + (T_y - \omega_y)^2 > r_1^2 \end{cases} \quad (1)$$

133 Notice that the problem of finding a minimum signed area annulus enclosing a
 134 first set of points but not a second set of points is more general than, but may
 135 be reduced to the usual problem of finding a minimum area annulus enclosing
 136 a set of points (right bottom case of Table 1).

137 The circularity measure of \mathcal{S} and \mathcal{T} is the squared ratio between r_1 and r_2 :

$$circ(\mathcal{S}, \mathcal{T}) = \frac{r_1^2}{r_2^2} \quad (2)$$

138 Now, we define the circularity measure of C as the circularity measure of \mathcal{S}
 139 and \mathcal{T} :

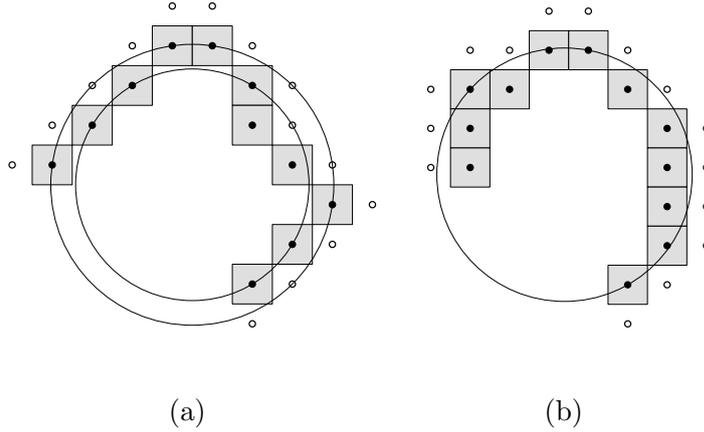


Fig. 3. Two parts of two digital boundaries are depicted with gray squares. \mathcal{S} (resp. \mathcal{T}) is the set of black disks (resp. white disks). In (a), the minimum area annulus has an area of 4 and the circularity measure equals $8.5/12.5 = 0.68$. However in (b), it has a null area and the circularity measure equals 1, because the part of digital boundary is a digital arc.

$$\begin{cases} circ(C) = circ(\mathcal{S}, \mathcal{T}) & \text{if } (circ(\mathcal{S}, \mathcal{T}) < 1) \\ circ(C) = 1 & \text{otherwise} \end{cases} \quad (3)$$

140 If the signed area $\pi(r_2^2 - r_1^2)$ of \mathbf{A} is strictly less than 0, \mathcal{S} and \mathcal{T} are separable
 141 by a circle and $circ(\mathcal{S}, \mathcal{T}) > 1$, but if $\pi(r_2^2 - r_1^2) \geq 0$, \mathcal{S} and \mathcal{T} are not sepa-
 142 rable by a circle and $circ(\mathcal{S}, \mathcal{T}) \leq 1$ (Fig. 3). As a consequence the circularity
 143 measure defined in equation 3 fulfils property 2. Moreover, it is clear that the
 144 measure is also intuitive and is robust to rigid transformations such that it
 145 fulfils properties 1 and 3.

146 **3 Computation of $\text{circ}(\mathcal{S}, \mathcal{T})$**

147 This section focuses on the computation of $\text{circ}(\mathcal{S}, \mathcal{T})$. First, we show that
 148 this computation may be achieved by linear programming in a space of di-
 149 mension 4. Next, we derive a simple geometric algorithm working in a space
 150 of dimension 3 only.

151 *3.1 Linear programming problem*

152 Developing the set of constraints of equation 1, we get:

$$\begin{cases} \forall S \in \mathcal{S}, -2aS_x - 2bS_y + f(S_x, S_y) + c_2 \leq 0 \\ \forall T \in \mathcal{T}, -2aT_x - 2bT_y + f(T_x, T_y) + c_1 > 0 \end{cases} \quad (4)$$

where

$$\begin{cases} a = \omega_x, & b = \omega_y, \\ c_1 = (a^2 + b^2 - r_1^2) & c_2 = (a^2 + b^2 - r_2^2) \\ f(x, y) = x^2 + y^2 \end{cases}$$

153 Instead of characterizing a circle by its centre and its radius, we characterize
 154 a circle by its centre and the power of the origin with respect to the circle.
 155 Thanks to this change of variables, solving equation 1 is equivalent to solving a
 156 linear program with four variables and $|\mathcal{S}| + |\mathcal{T}|$ constraints (where $|\cdot|$ denotes
 157 the cardinality of a set).

158 This kind of reformulation into a linear programming problem has been done,
 159 for instance, in computational geometry for the smallest enclosing circle [36]

160 or the smallest separating circle [14], in discrete geometry for digital circle
161 recognition [19] and in engineering for the quality control of manufactured
162 parts [23].

163 The technique of Megiddo [37] is linear in time in the number of constraints.
164 Unfortunately, Megiddo's algorithm is not easy to implement and the constant
165 is large and is exponential in the dimension, which is equal to 4 here. In a space
166 of dimension 4, Megiddo's algorithm cannot be used in practice. That's why
167 we propose in this section a simple geometric algorithm that works in a space
168 of dimension 3 only.

169 As an annulus is a pair of concentric circles that are characterized by three
170 parameters each, we interpret equation 4 in a 3D space that we call *abc*-
171 space. Indeed, c_1 and c_2 , having the same meaning, are both represented in
172 the c -axis. From now on, in addition to the original plane, called xy -plane,
173 containing the points of \mathbb{Z}^2 , we work in the *abc*-space as well as in its dual
174 space, called *xyz*-space.

175 3.2 *abc-space vs xyz-space*

176 By definition $0 \leq r_1 \leq r_2$, whereas $a^2 + b^2 \leq c$, which implies that the *abc*-
177 space is a copy of \mathbb{R}^3 from which the interior of the paraboloid of equation
178 $c = a^2 + b^2$ has been excluded. A point on the paraboloid maps to a circle
179 of null radius in the xy -plane. A point that is out of the paraboloid maps
180 to a circle whose radius is equal to the vertical distance between the point
181 and the paraboloid in the xy -plane (Fig. 4.a). It is clear that two points with
182 the same projection in the ab -plane corresponds to two concentric circles in

183 the xy -plane. Minimizing the area of an annulus bounded by such a pair of
184 concentric circles is tantamount to minimize the vertical distance between the
185 two corresponding points in the abc -space.

186 In the xyz -space, all the points of \mathbb{Z}^2 are lifted along an extra axis (the z -
187 axis) according to the bivariate function f . Let $\mathcal{S}' = \{S'(S'_x, S'_y, S'_z)\}$ (resp.
188 $\mathcal{T}' = \{T'(T'_x, T'_y, T'_z)\}$) be the set of points of \mathcal{S} (resp. \mathcal{T}) that are vertically
189 projected onto the paraboloid of equation $z = f(x, y) = x^2 + y^2$. Any plane
190 in the xyz -space passing through some points of \mathcal{S}' or \mathcal{T}' cuts the paraboloid.
191 The projection on the xy -plane of the intersection between the plane and the
192 paraboloid is a circle that passes through the corresponding points of \mathcal{S} and
193 \mathcal{T} (Fig. 4.b). The intersection between the paraboloid and a pair of parallel
194 planes projects to a pair of concentric circles on the xy -plane. Minimizing the
195 area of an annulus bounded by such a pair of concentric circles is tantamount
196 to minimize the vertical distance between the two corresponding planes in
197 the xyz -plane. This kind of transformation is well known in computational
198 geometry since [38] and has already been used in [36] to solve the smallest
199 enclosing circle or in [14] to solve the smallest separating circle problem.

200 The understanding of the constraints is more straightforward in the xyz -plane
201 and that is why we will preferably work in this space in the following subsec-
202 tion.

203 3.3 *Pair of parallel planes*

204 We have to compute a pair of parallel planes such that the upper plane is
205 above the points of \mathcal{S}' and the lower plane is below the points of \mathcal{T}' in order

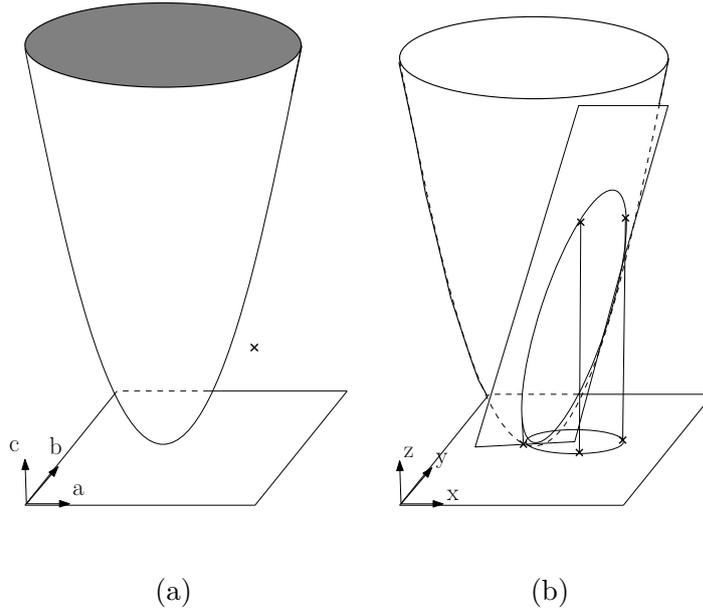


Fig. 4. (a) A point outside the paraboloid of equation $c = a^2 + b^2$ in the abc -space corresponds to a circle in the xy -plane and conversely. (b) A plane that cuts the paraboloid of equation $z = x^2 + y^2$ in the xyz -space corresponds to a circle in the xy -plane and conversely.

206 to solve equation 4 and derive a circularity measure.

207 Obviously, \mathcal{S}' and \mathcal{T}' may be reduced to their convex hull denoted by $CH(\mathcal{S}')$
 208 and $CH(\mathcal{T}')$. In addition, the property of convexity makes the next step that
 209 consists in minimizing the vertical distance between the two parallel planes of
 210 support more efficient.

211 We do not detail the classical convex hull computation algorithm that may
 212 run in $\mathcal{O}(m \log m)$, where $m = |\mathcal{S}'| + |\mathcal{T}'|$ [39,40].

213 An elementary way to compute the pair of parallel planes of support minimiz-
 214 ing their vertical distance is to compute the intersection depth between the two
 215 polyhedra $CH(\mathcal{S}')$ and $CH(\mathcal{T}')$ denoted by $h = \min Height(CH(\mathcal{S}'), CH(\mathcal{T}'))$.
 216 $Height(CH(\mathcal{S}'), CH(\mathcal{T}'))$ is a function that returns the distance between the

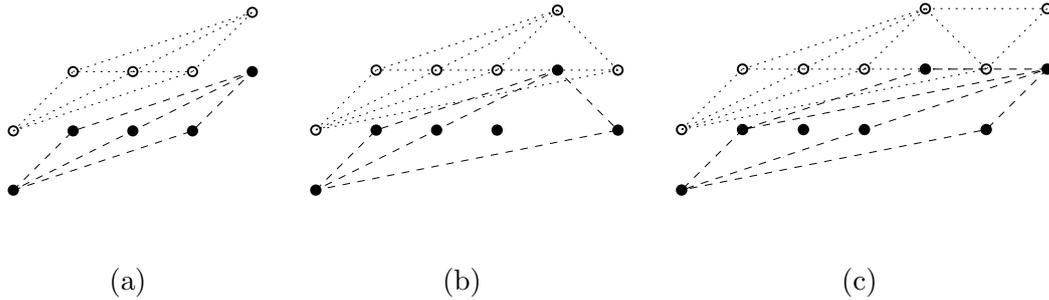


Fig. 5. \mathcal{S} (black disks) and \mathcal{T} (white disks) are separable by a straight line in (a), by a circle in (b) and are not separable by a circle in (c). Note that G^* , which is the intersection between $G_{\mathcal{S}}$ (in dashed lines) and $G_{\mathcal{T}}$ (in dotted lines), has respectively 0, 4 and 3 nodes in (a), (b) and (c).

217 two polyhedra along the z -axis for each point of the domain of the function.
 218 Notice that $Height(CH(\mathcal{S}'), CH(\mathcal{T}'))$ is not defined everywhere. Indeed, the
 219 domain of this function is the intersection of the projection on the xy -plane
 220 of $CH(\mathcal{S}')$ and $CH(\mathcal{T}')$, that is $CH(\mathcal{S}) \cap CH(\mathcal{T})$.

221 To compute h , the brute force algorithm consists in computing the planar
 222 graph G^* that is the intersection between $G_{\mathcal{S}}$ and $G_{\mathcal{T}}$ (Fig. 5). If $|G^*| = 0$,
 223 then $CH(\mathcal{S}) \cap CH(\mathcal{T}) = \emptyset$. In this degenerate case, \mathcal{S}' and \mathcal{T}' are separable by
 224 a plane that is orthogonal to the xy -plane, \mathcal{S} and \mathcal{T} are separable by a circle
 225 of infinite radius, that is a straight line, so the part of digital boundary from
 226 which \mathcal{S} and \mathcal{T} have been computed is a digital straight segment (Fig. 5.a). If
 227 $|G^*| > 0$, it remains to compute the height function for each vertex of G^* and
 228 take the minimum.

229 The brute force algorithm runs in $\mathcal{O}(m^2)$ since G^* has at most m^2 vertices.
 230 However, it is possible to take advantage of the convexity of the height function
 231 to get an algorithm in $\mathcal{O}(m \log m)$ (see [39, pages 310-315] for this algorithm).

232 Although our algorithm is more general than a simple digital circle test, its

233 complexity in $\mathcal{O}(m \log m)$ is better than the quadratic complexity of the meth-
 234 ods presented in [15,16,20]. These methods cannot be efficient because they
 235 only deal with 2D projections of 3D polyhedrons.

236 Once the pair of parallel planes of support are known, we have $\text{circ}(\mathcal{S}, \mathcal{T}) =$
 237 $\frac{r_1^2}{r_2^2}$, where r_1^2 and r_2^2 are derived from the coefficients of the pair of parallel
 238 planes. From equation 4, it is obvious to get the following equations: $r_1^2 =$
 239 $a^2 + b^2 - c_1$ and $r_2^2 = a^2 + b^2 - c_2$.

240 Since h is the signed area of the annulus \mathbf{A} , if $h < 0$, \mathcal{S} and \mathcal{T} are separable
 241 by a circle and $\text{circ}(\mathcal{S}, \mathcal{T}) > 1$ but if $h \geq 0$, \mathcal{S} and \mathcal{T} are not separable by a
 242 circle and $\text{circ}(\mathcal{S}, \mathcal{T}) \leq 1$.

Algorithm 1 sums up the current section.

Algorithm 1 CircularityComputation(\mathcal{S}, \mathcal{T})

Input: \mathcal{S} and \mathcal{T} , two sets of points

Output: $\text{circ}(\mathcal{S}, \mathcal{T})$

- 1: Compute \mathcal{S}' (resp. \mathcal{T}'), the set of the vertical projections of the points of
 \mathcal{S} (resp. \mathcal{T}) onto the elliptic paraboloid of equation $z = x^2 + y^2$
 - 2: Compute the 3D convex hull of \mathcal{S}' and \mathcal{T}' [40]
 - 3: Compute the pair of parallel planes of support [39, pages 310-315]
 - 4: Compute r_1^2 and r_2^2 from the coefficients of the parallel planes (a, b, c_1, c_2) :
 $r_1^2 = a^2 + b^2 - c_1$ and $r_2^2 = a^2 + b^2 - c_2$
 - 5: **return** r_1^2/r_2^2
-

243

244 **4 Minimization of the size of \mathcal{S} and \mathcal{T}**

245 In the aim of decreasing the burden of the computation of $\text{circ}(\mathcal{S}, \mathcal{T})$, which
 246 depends on the size of \mathcal{S} and \mathcal{T} , we search for $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$ such that $\hat{\mathcal{S}} \subseteq \mathcal{S}$,
 247 $\hat{\mathcal{T}} \subseteq \mathcal{T}$, $|\hat{\mathcal{S}}| + |\hat{\mathcal{T}}| < |\mathcal{S}| + |\mathcal{T}|$ and $\text{circ}(\hat{\mathcal{S}}, \hat{\mathcal{T}}) = \text{circ}(\mathcal{S}, \mathcal{T})$.

248 *4.1 Computation of $\hat{\mathcal{S}}$*

249 Let us consider a part C_{ij} of the boundary C . Since all circles are convex, no
 250 circle can enclose the vertices of the convex hull of C_{ij} without enclosing all
 251 its points. So $\hat{\mathcal{S}}$ is the set of the vertices of the convex hull of C_{ij} , denoted by
 252 $\mathcal{CH}(C_{ij})$. If $C_{ij} \neq C$, the first and last points of C_{ij} are put in $\hat{\mathcal{S}}$ even if they
 253 are not in $\mathcal{CH}(C_{ij})$ to make the extraction of the points of $\hat{\mathcal{T}}$ easier.

254 *4.2 Computation of $\hat{\mathcal{T}}$*

255 The extraction of the points of $\hat{\mathcal{T}}$ is independently performed for each part
 256 $C_{kl} \in C_{ij}$ that is lying between two consecutive points that belongs to $\hat{\mathcal{S}}$, the
 257 indices of which being respectively denoted by k and l . Let us denote by s_k
 258 and s_l the two end points of the part C_{kl} .

259 As the extraction algorithm depends on the convexity of C_{kl} , the following
 260 definition of convexity is required:

261 **Definition 1** *As C_{ij} is clockwise oriented, the right (resp. left) part of $\mathcal{CH}(C_{ij})$*
 262 *is the polygonal line that links C_i and C_j and that lies on the right (resp. left)*
 263 *of C_{ij} . C_{ij} is convex (resp. concave) if and only if there is no digital point*

264 between the polygonal line linking the digital points of C_{ij} and the right (resp.
 265 left) part of $\mathcal{CH}(C_{ij})$.

266 4.2.1 Case where C_{kl} is not convex

267 If C_{kl} is not convex, all the points of \bar{C} that are located between the digital
 268 points of C_{kl} and the segment $[s_k s_l]$ are put in $\hat{\mathcal{T}}$ (Figure 6).

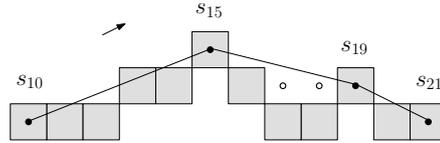


Fig. 6. Since C_{15-19} is not convex (according to the clockwise orientation), the background points that are added to $\hat{\mathcal{T}}$ are such that: (i) they are 4-neighbours of a point of C_{15-19} , (ii) they are located between C_{15-19} and the segment $[s_{15}s_{19}]$.

269 4.2.2 Case where C_{kl} is convex

270 Without loss of generality, let us consider the segment $[s_k s_l]$ in the first octant,
 271 so that the background points are located above $[s_k s_l]$. Let us consider the
 272 arithmetic description of $[s_k s_l]$ with a vector $\vec{u} = (a, b)^T$ with $a, b \in \mathbb{Z}$ and
 273 $\gcd(a, b) = 1$, such that $(s_l - s_k) = g \cdot \vec{u}$ with $g \in \mathbb{Z}$.

274 In order to have $\text{circ}(\hat{\mathcal{S}}, \hat{\mathcal{T}}) = \text{circ}(\mathcal{S}, \mathcal{T})$, we must keep the closest background
 275 points to the outer disk containing $[s_k s_l]$ but not containing any background
 276 point. If we assume first that the outer disk has a infinite radius, we show that
 277 we must keep the Bezout points of $[s_k s_l]$ whose definition is given below:

278 **Definition 2** A Bezout point b_q of a segment $[s_k s_l]$ is defined as a point above
 279 $[s_k s_l]$ such that $s_k \vec{b}_q = \vec{v} + q\vec{u}$ with $q \in [0, g]$, $\vec{v} = (c, d)^T$ and $\det(\vec{u}, \vec{v}) = 1$.

280 **Lemma 1** *A circle of infinite radius that encloses $[s_k s_l]$ but does not enclose*
 281 *any Bezout point b_q , does not enclose any other point above $[s_k s_l]$.*

282 This lemma and its proof may be found in other papers such as [20]. They are
 283 the basement of the arithmetic digital straight line recognition algorithm [27]
 284 because any lower leaning point of an 8-connected digital straight segment in
 285 the first octant that is vertically translated up by 1 is a Bezout point associated
 286 to this segment.

287 Lemma 1 shows that only Bezout points need to be taken into consideration
 288 as points of $\hat{\mathcal{T}}$. Furthermore, it seems that only a small part of them, located
 289 near the bisector of $[s_k s_l]$, are sufficient. In [20] (Definition 1), the closest point
 290 to the middle of $[s_k s_l]$ is arbitrarily chosen. Fig. 7 illustrates that only taking
 291 into account the closest point to the middle of $[s_k s_l]$ is not sufficient. In the
 292 following, we prove that at most two Bezout points have to be taken into
 account.

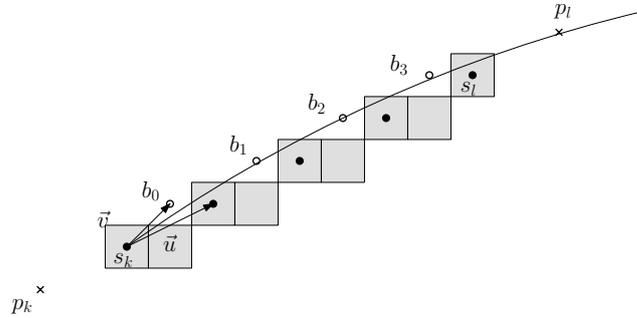


Fig. 7. The closest Bezout point to the middle of $[s_k s_l]$, denoted by b_1 , is not
 sufficient: there is a circle that separates b_1 from s_k and s_l but encloses b_2 , which is
 another Bezout point.

293

294 For each convex part C_{kl} , let us consider two extra points defined as the
 295 points p_k and p_l such that $p_k = s_k - \vec{u}$ and $p_l = s_l + \vec{u}$ (Fig. 7). p_k and p_l are
 296 background points, since $[s_k s_l]$ is an edge of a convex hull. The circles that

297 enclose $[s_k s_l]$ but do not enclose any background point cannot have an infinite
 298 radius because they must not enclose neither p_k nor p_l .

299 Let us introduce the following new definition:

300 **Definition 3** *The middle Bezout point(s) associated to the segment $[s_k s_l]$*
 301 *is(are) defined as:*

- 302 (1) *the unique Bezout point b_0 , if $g = 1$.*
 303 (2) *the two consecutive Bezout points b_{q-1} and b_q the closest to the middle*
 304 *such that q is the smallest integer for which the quantity $|2(\vec{u} \cdot \vec{v}) + (2q -$*
 305 *$g - 1)|\|\vec{u}\|^2|$ is minimized.*

306 Vector \vec{u} and integer g may be computed by applying Euclid's algorithm to
 307 the slope of $[s_k s_l]$. Vector \vec{v} is given by the Bezout's identity that is found
 308 thanks to the extended Euclid's algorithm. These computations are obviously
 309 made in $\mathcal{O}(\log(\max(|a|, |b|)))$. Once g , \vec{u} and \vec{v} are known, finding the middle
 310 Bezout points is sequentially performed in $\mathcal{O}(g)$.

311 Let us state the following proposition:

312 **Proposition 1** *A circle that encloses $[s_k s_l]$ but does not enclose neither the*
 313 *middle Bezout points associated to $[s_k s_l]$ nor the extra points p_k and p_l , does*
 314 *not enclose any other Bezout point.*

315 Because of its length, the proof is given in appendix, section A.

316 As a result, for each convex part C_{kl} , only two background points at most,
 317 which are the middle Bezout point(s), must be put in $\hat{\mathcal{T}}$. Notice that deciding if
 318 the extra point p_k (resp. p_l) also must be added to $\hat{\mathcal{T}}$ is done when considering,
 319 if it exists, the previous (resp. next) part of C_{ij} . As an exception, if C_{kl} is the

320 first (resp. last) convex part of C_{ij} , then the extra point p_k (resp. p_l) is also
 321 added to $\hat{\mathcal{T}}$.

322 4.3 Algorithm and Complexity

323 The algorithm that computes $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$ (Algorithm 2) is given below.

Algorithm 2 SnTComputation($C_{ij}, \hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$)

Input: C_{ij} , a part of a digital boundary

Output: $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$

- 1: $\hat{\mathcal{S}} = \hat{\mathcal{T}} = \emptyset$
 - 2: Add s_i to $\hat{\mathcal{S}}$
 - 3: Compute $\mathcal{CH}(C_{ij})$
 - 4: **for each** part C_{kl} of C_{ij} **do**
 - 5: Add s_l to $\hat{\mathcal{S}}$
 - 6: **if** C_{kl} is convex **then**
 - 7: Add the middle Bezout point(s) of $[s_k s_l]$ to $\hat{\mathcal{T}}$
 - 8: **else**
 - 9: Add to $\hat{\mathcal{T}}$ all the points of \bar{C} that are located between the digital
 points of C_{kl} and $[s_k s_l]$.
 - 10: **return** $\hat{\mathcal{S}}, \hat{\mathcal{T}}$
-

324 Computing $\mathcal{CH}(C_{ij})$ (1.3) is done in linear time (using Melkman's algorithm
 325 [42] for instance). The points of \bar{C} that are 4-neighbours of a point of C_{kl} are
 326 computed in linear time by contour tracking. Checking whether each part C_{kl}
 327 is convex or not (1.6) and performing the appropriate processing (1.7 and 1.9)
 328 is then straightforward and in $\mathcal{O}(l - k)$.

329 Fig. 8 illustrates that $|\hat{\mathcal{S}}|$ and $|\hat{\mathcal{T}}|$ are considerably smaller than $|\mathcal{S}|$ and $|\mathcal{T}|$,

330 if C_{ij} is convex.

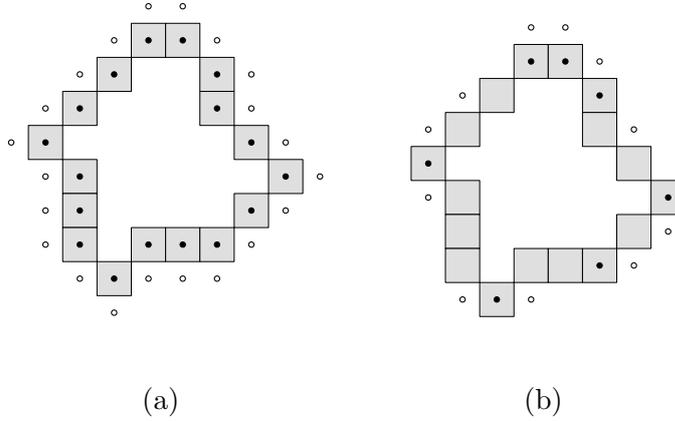


Fig. 8. \mathcal{S} in (a) and $\hat{\mathcal{S}}$ (b) (resp. \mathcal{T} in (a) and $\hat{\mathcal{T}}$ (b)) are depicted with black disks (resp. white disks).

331 Actually, $|\hat{\mathcal{S}}|$ is bounded by $\mathcal{O}(n^{2/3})$ according to known results [41]. If C_{ij} is
 332 convex, $|\hat{\mathcal{T}}|$ is at most twice bigger than $|\hat{\mathcal{S}}|$ according to Proposition 1 and $|\hat{\mathcal{T}}|$
 333 is bounded by $\mathcal{O}(n)$ otherwise. Therefore $m = |\hat{\mathcal{S}}| + |\hat{\mathcal{T}}|$ is bounded by $\mathcal{O}(n^{2/3})$
 334 in the case of convex parts and $\mathcal{O}(n)$ otherwise. As $\text{circ}(\mathcal{S}, \mathcal{T}) = \text{circ}(\hat{\mathcal{S}}, \hat{\mathcal{T}})$
 335 can be computed in $\mathcal{O}(m \log m)$, we can conclude that the circularity measure
 336 of C_{ij} can be computed in $\mathcal{O}(n)$ if C_{ij} is convex and $\mathcal{O}(n \log n)$ otherwise.

337 5 Experiments

338 It is clear that the proposed circularity measure fulfils the three properties of
 339 Section 2.2. In this section, the proposed circularity measure is probed with
 340 respect to its ability to deal with a part of a digital boundary.

342 Hundreds of noisy circles are generated. In order to study the impact of the
 343 amount of noise onto circularity, we implemented a degradation model very
 344 close to the one investigated in [43]. This model was proposed and validated
 345 in the context of document analysis and assume that: (i) the probability to
 346 flip a pixel (that is, label ‘foreground’ or ‘1’ a pixel previously labelled ‘back-
 347 ground’ or ‘0’, and conversely) depends of its distance to the nearest pixel of
 348 the complement set and (ii) blurring may be simulated with a morphological
 349 closing.

350 Figure 9 gives two examples of results of the degradation algorithm applied
 351 to a digital disk.

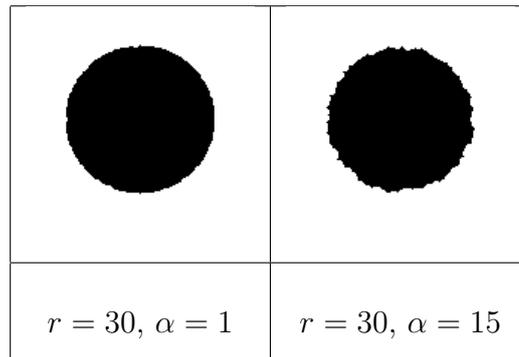


Fig. 9. Gauss digitization of two disks. The amount of noise that is added to the disks according to the degradation model of [43] depends of parameter alpha. The digital curves that we are called upon to measure are the 8-connected boundaries of these digital objects.

352 Figure 10 shows that the circularity decreases with the amount of noise, but
 353 with sawtooth because the pixels are flipped at random. The noisier the dig-
 354 ital circle, the more it looks different from a digital circle. Furthermore, even
 355 with rather noisy digital circles ($\alpha = 15$), the circularity is above 0.8, which

356 approximately corresponds to the circularity of a 7-gon. Hence, the measure
 357 is sufficiently robust to discriminate noisy circles given by the noise model of
 358 [43] at $\alpha = 15$, from k -gons where $k < 7$, such as triangles or squares, hav-
 359 ing a circularity around 0.3 and 0.4 respectively. Note that the comparison
 360 makes sens in spite of the difference of perimeter because the measure is size
 361 invariant.

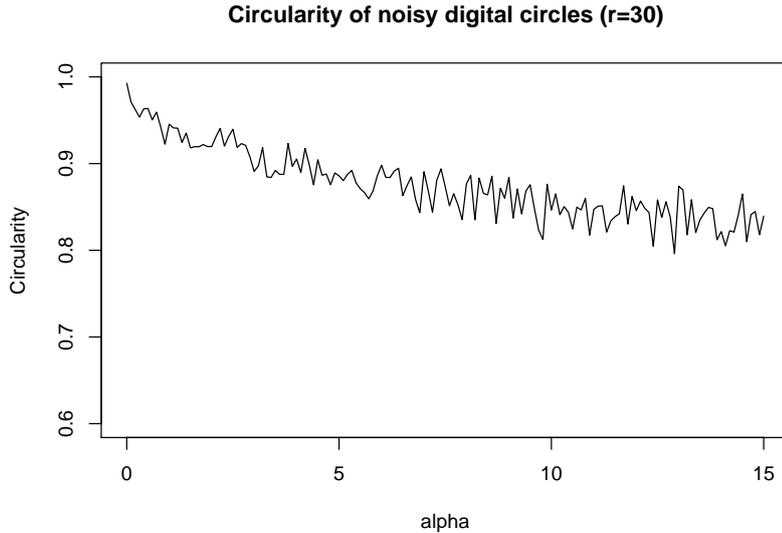


Fig. 10. One hundred digital circles of radius 30 are generated with more and more noise. Parameter alpha ranging from 1 to 15 controls the amount of noise (Fig. 9). Circularity is plotted against parameter alpha.

362 The accuracy of the measurements on digital arcs of various length is now
 363 investigated. Fifty noisy circles are generated ($r = 30$, $\alpha = 15$) (Fig. 9).
 364 For each circle and for each length from 20 to approximately 180 pixels, one
 365 digital arc is randomly extracted. The circularity measure is computed from
 366 these approximately 7500 digital arcs. Fig. 11 shows that from 20 to 45 pixels
 367 of length (90 degrees), measurements are not accurate, because the confidence
 368 range at 95% is wide (until more than 0.1). Though, the confidence range
 369 shrinks while the arc length increases and the measurements done on digital

370 arcs of more than 45 pixels of length (90 degrees) may be consider accurate.
 371 Obviously, the smallest angle for which measurements are accurate depends
 372 on the noise and the size of the digital circles. The smaller α is, the smaller
 373 the angle is. In the special case where $\alpha = 0$, measurements are perfect for
 374 all digital arcs. Moreover, the higher the radius is, the less the noise added by
 375 the model at a given α affects the shape, the smaller the angle is.

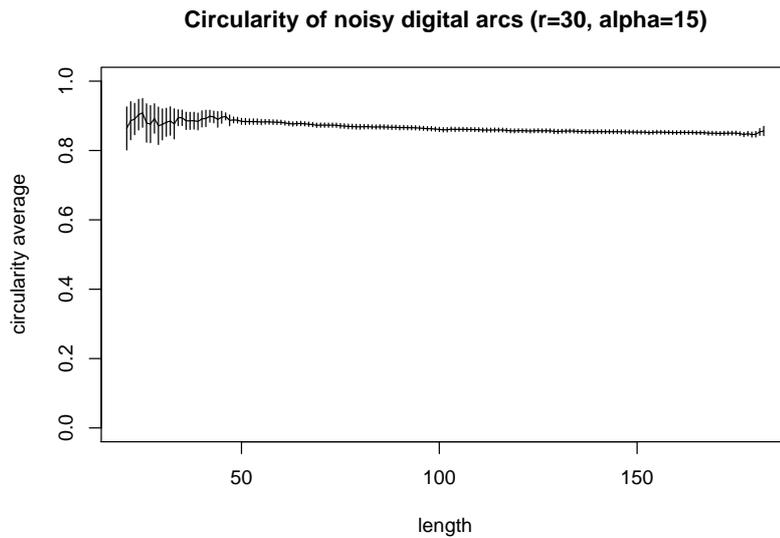


Fig. 11. Fifty noisy circles were generated ($r = 30$, $\alpha = 15$) (Fig. 9). For each circle, for each length from 20 to approximately 180 pixels, one digital arc is randomly extracted. The average of the circularity measure of the digital arcs (solid line) is plotted against the length with error bars at 95%.

376 *5.2 Real-world images*

377 We are currently working in collaboration with geographers. They want to
 378 perform a set of measurements that describes the shape of pebbles sedimented
 379 in river beds. The underlying assumption is that pebbles size and shape are
 380 determined by lithology, distance of transport, abrasion, etc. The objective is

381 to reduce the subjectivity and the time spent in the field thanks to digital
382 image analysis.

383 The circularity measure proposed in this paper is used in order to study the
384 shape of pebbles from digital images, collected in the bed of the Progo, an
385 Indonesian river located on Java Island near Yogyakarta. Approximately 1300
386 pebbles were randomly sampled in the bed, with 2 photos being taken on 12
387 stations located at various distances from the source. Fig. 12 shows two photos
taken near the source.



Fig. 12. Zoom in photos taken on the first (left) and second (right) stations.

388

389 First, we detected pebbles with clustering methods in the HSV color-space
390 and we extracted their digital boundary. Next, the circularity measure was
391 computed for all the digital boundaries.

392 In Fig. 13, the average of the circularity measure of the pebbles is plotted
393 against the distance from the source of the stations where the pebbles have
394 been collected. Circularity is valuable for geographers because experiments
395 show that it increases in the first 20 kilometres, while the pebbles get rounder,
396 but has a complex pattern after, with no clear trend, which raises the possibil-
397 ity of a substitution of macro-scale to micro-scale shape changes downstream.

398 Notice that Fig. 12 shows photos taken on two stations that have statisti-
 399 cally significant difference of circularity: the first station (Fig. 12, left) and
 400 the second one (Fig. 12, right). Obviously, other size, form and shape param-
 401 eters, like diameter, elongation, convexity and roundness [1], are computed in
 402 addition to circularity to provide multidimensional data of great interest for
 geographers.

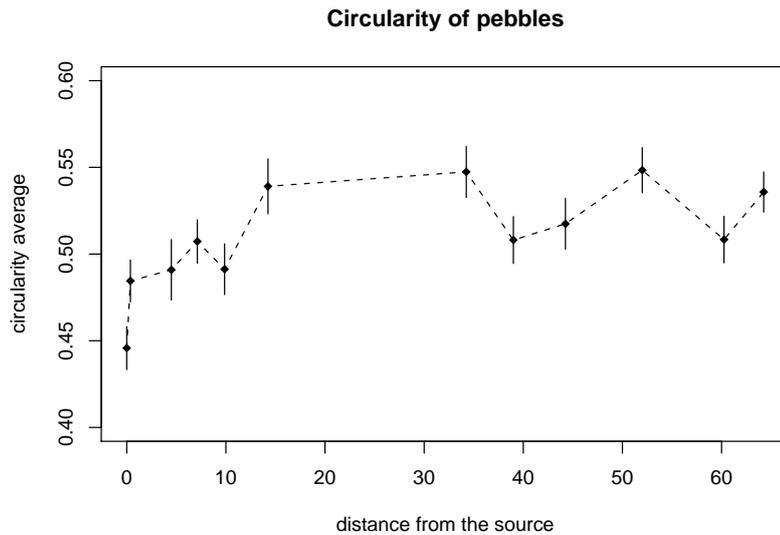


Fig. 13. The average of the circularity measure of the pebbles is plotted against the distance from the source of the 12 stations where the 1300 pebbles have been collected.

403

404 In the left photo of Fig. 12, two pebbles are badly detected because they touch
 405 each other. Another example is presented on the left of Fig. 14.

406 In such cases, it is possible to cut the digital boundary in two and independ-
 407 ently deal with the two parts of the digital boundary. We used an algorithm
 408 that robustly decomposes a digital curve into convex and concave parts [45].
 409 Each part may be viewed as a part of a pebble outline that has not been wholly
 410 retrieved. As the missing part of each outline is small enough, the circularity



Fig. 14. Corrupted outlines of pebbles.

411 measure of the retrieved part is assumed to be very close to the one that could
412 have been computed on the whole digital boundary. In the example presented
413 in the left photo of Fig. 14, the circularity measure of the whole digital bound-
414 ary is 0.005, whereas the circularity measure of the two parts corresponding
415 to the two pebbles is 0.488 and 0.598 respectively, from left to right.

416 In the right photo of Fig. 14, is presented a pebble outline that is corrupted
417 with a spike. Using [45], the digital boundary is coarsely cut before and after
418 the spike. The digital boundary with and without the spike has the same
419 circularity measure that is equal to 0.511 because the spike does not affect the
420 fitting of the minimum area annulus.

421 Generally speaking, the proposed method is able to infer the circularity mea-
422 sure of a digital boundary from a part of it, provided that bumps are uniformly
423 spread around the boundary and that the part is long enough with respect
424 to the amplitude of the bumps. For instance, in Section 5.1, it was shown
425 that for circles of radius 30 that are corrupted by the noise model of [43] at
426 $\alpha = 15$, the measurements done on parts of more than 90 degrees may be
427 consider accurate. We took profit of this property in our application to cope
428 with occlusions and spikes.

429 6 Conclusion and perspectives

430 In this paper, a circularity measure has been defined for parts of digital bound-
431 aries. An existing circularity measure of a set of discrete points, which is
432 sometimes used in computational metrology, is extended to the case of parts
433 of digital boundaries (Section 2.2). Once the minimum area annulus, such that
434 the outer disk contains all the points of the part of a digital boundary and the
435 inner disk does not contain any background point is computed, the circular-
436 ity measure is defined as the squared ratio between the inner and outer radii
437 (Section 2.2).

438 Because we consider two sets of points, the problem we deal with is more
439 general than the usual problem of finding a minimum area annulus enclosing
440 one set of points [5,22,6,7,9,10,12]. The circularity measure of these two sets
441 of points is computed thanks to an algorithm in $\mathcal{O}(n \log n)$ that only uses
442 classical tools of computational geometry (Section 3). Moreover, the two sets
443 of points may be computed so that the algorithm is linear in time in the case
444 of convex digital boundaries (Section 4). The method is exact contrary to
445 many methods that use *ad hoc* heuristics [7] or meta-heuristics like simulated
446 annealing [10,12]. Even if it is shown that a sophisticated machinery coming
447 from linear programming can provide a linear time algorithm (Section 3.1),
448 its time complexity is better than many quadratic methods based on Voronoi
449 diagrams [15,16,5,22] (Section 3.3).

450 Contrary to the famous measure introduced in [21], the measure proposed in
451 this paper fulfils the following properties:

- 452 • it may be applied on any part of digital boundaries.

- 453 • it is robust to rigid transformations.
- 454 • it ranges from 0 to 1 and is equal to 1 for any digital circle or arc, which
- 455 means that the measurements are accurate even at low resolution.
- 456 • it provides the parameters of a circle whose digitization is the measured
- 457 part of digital boundary if the circularity measure is 1 and the parameters
- 458 of an approximating circle otherwise.

459 The kind of measure and algorithm proposed in this paper is general enough to
 460 be applied in order to recognize or measure the deviation with other quadratic
 461 shapes like parabolas. In the case of parabolas, the extension is straightfor-
 462 ward: it is enough to modify function f , so that $f(x, y)$ equals x^2 (or y^2),
 463 instead of $x^2 + y^2$. The points of the xy -plane are merely vertically projected
 464 onto a parabolic cylinder instead of an elliptic paraboloid and algorithm 1
 465 does not change.

466 To end, it would be quite valuable to make the algorithm on-line (without
 467 increasing its complexity as far as possible). The on-line property would be
 468 of great interest to efficiently and robustly decompose a digital boundary into
 469 primitives like noisy digital arcs or pieces of noisy digital parabolas.

470 **Acknowledgements.** The authors thank the reviewers for their comments
 471 that significantly improved the paper.

472 **A Proof of Proposition 1**

473 In the sequel, we only consider the case of a circle that encloses $[s_k, s_l]$ but nei-
 474 ther p_l nor the closest middle Bezout point to p_l . The other case is symmetric
 475 and the two cases will be put together to conclude the proof.

476 Let us consider a circle passing through s_k and p_l . If such a circle encloses s_l
 477 but does not enclose any Bezout point, then any circle passing through s_k and
 478 intersecting $[s_l p_l]$ (of whatever radius) separates s_l from any Bezout point too.

479 The first point b that is touched by a circle passing through s_k and p_l of
 480 decreasing radius is such that the angle between \vec{bs}_k and \vec{bp}_l is maximized. To
 481 maximize such an angle in the range $[\pi/2, \pi]$ is equivalent to maximize the
 482 tangent of the angle that equals:

$$\frac{\det(\vec{bs}_k, \vec{bp}_l)}{\vec{bs}_k \cdot \vec{bp}_l}$$

483 However, $\det(\vec{bs}_k, \vec{bp}_l)$ is constant and equal to $g + 1 = h$. Then, only taking
 484 into account the denominator, we look for the integer q that minimizes:

$$f : \mathbb{Z} \mapsto \mathbb{Z}$$

$$f(q) = (-\vec{v} - q\vec{u}) \cdot (-\vec{v} + (h - q)\vec{u})$$

485 Developing, we finally get:

$$f(q) = q^2(\|\vec{u}\|^2) + q(2(\vec{u} \cdot \vec{v}) - h(\|\vec{u}\|^2)) + (\|\vec{v}\|^2 - h(\vec{u} \cdot \vec{v}))$$

486 The derivative is:

$$f'(q) = (2\|\vec{u}\|^2)q + 2(\vec{u} \cdot \vec{v}) - h(\|\vec{u}\|^2)$$

487 Since $2\|\vec{u}\|^2 \geq 0$, f is convex and has a global minimum at the value of q

488 for which $f'(q)$ is closer to 0 than for the other values of q . The minimum is
489 reached around $q = h/2$ because $f'(h/2) = 2(\vec{u} \cdot \vec{v}) \geq 0$ and that's why we call
490 the Bezout point b_q such that q is the smallest integer for which the quantity
491 $|2(\vec{u} \cdot \vec{v}) + (2q - h)|\|\vec{u}\|^2|$ is minimized (Def. 3) the middle Bezout point.

492 To end, the first point b that is touched by the circle of decreasing radius and
493 passing through s_k and p_l is the closest middle Bezout point to p_l according
494 to Def. 3. Similarly, we can show that the first point b that is touched by the
495 circle of decreasing radius and passing through s_l and p_k is the closest middle
496 Bezout point to p_k according to Def. 3, which concludes the proof. \square

References

- [1] H. Wadell, Volume, shape, and roundness of rock particles, *Journal of Geology* 40 (1932) 443–451.
- [2] A. Thom, A statistical examination of megalithic sites in britain, *Journal of Royal Statistical Society* 118 (1955) 275–295.
- [3] X. Hilaire, K. Tombre, Robust and accurate vectorization of line drawings, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (6) (2006) 890–904.
- [4] I. Frosio, N. A. Borghese, Real-time accurate circle fitting with occlusions, *Pattern Recognition* 41 (2008) 1045–1055.
- [5] V.-B. Le, D. T. Lee, Out-of-roundness problem revisited, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (3) (1991) 217–223.
- [6] K. Swanson, D. T. Lee, V. L. Wu, An optimal algorithm for roundness determination on convex polygons, *Computational Geometry* 5 (1995) 225–235.

- [7] J. Pegna, C. Guo, Computational metrology of the circle, in: *Computer graphics international*, 1998, pp. 350–363.
- [8] M. de Berg, P. Bose, D. Bremner, S. Ramaswami, G. Wilfong, Computing constrained minimum-width annuli of points sets, *Computers-Aided Design* 30 (4) (1998) 267–275.
- [9] P. K. Agarwal, B. Aronov, S. Har-Peled, M. Sharir, Approximation and exact algorithms for minimum-width annuli and shells, *Discrete & Computational Geometry* 24 (4) (2000) 687–705.
- [10] M.-C. Chen, Roundness measurements for discontinuous perimeters via machine visions, *Computers in Industry* 47 (2002) 185–197.
- [11] P. Bose, P. Morin, Testing the quality of manufactured disks and balls, *Algorithmica* 38 (2004) 161–177.
- [12] C. M. Shakarji, A. Clement, Reference algorithms for chebyshev and one-sided data fitting for coordinate metrology, *CIRP Annals - Manufacturing Technology* 53 (1) (2004) 439–442.
- [13] C. E. Kim, T. A. Anderson, Digital disks and a digital compactness measure, in: *Annual ACM Symposium on Theory of Computing*, 1984, pp. 117–124.
- [14] J. O'Rourke, S. R. Kosaraju, N. Meggido, Computing circular separability, *Discrete and Computational Geometry* 1 (1986) 105–113.
- [15] S. Fisk, Separating points sets by circles, and the recognition of digital disks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8 (1986) 554–556.
- [16] V. A. Kovalevsky, New definition and fast recognition of digital straight segments and arcs, in: *International Conference on Pattern Analysis and Machine Intelligence*, 1990, pp. 31–34.

- [17] S. Pham, Digital circles with non-lattice point centres, *The Visual Computer* 9 (1992) 1–24.
- [18] M. Worring, A. W. M. Smeulders, Digitized circular arcs: characterization and parameter estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (6) (1995) 554–556.
- [19] P. Damaschke, The linear time recognition of digital arcs, *Pattern Recognition Letters* 16 (1995) 543–548.
- [20] D. Coeurjolly, Y. Gérard, J.-P. Reveillès, L. Tougne, An elementary algorithm for digital arc segmentation, *Discrete Applied Mathematics* 139 (1-3) (2004) 31–50.
- [21] R. M. Haralick, A measure for circularity of digital figures, *IEEE Transactions on Systems, Man and Cybernetics* 4 (1974) 394–396.
- [22] U. Roy, X. Zhang, Establishment of a pair of concentric circles with the minimum radial separation for assessing roundness error, *Computer Aided Design* 24 (3) (1992) 161–168.
- [23] S. I. Gass, C. Witzgall, H. H. Harary, Fitting circles and spheres to coordinate measuring machine data, *The International Journal in Flexible Manufacturing Systems* 10 (1998) 5–25.
- [24] T. J. Rivlin, Approximation by circles, *Computing* 21 (1979) 93–104.
- [25] M. J. Bottema, Circularity of objects in images, in: *International Conference on Acoustics, Speech, and Signal Processing*, 2000, pp. 2247–2250.
- [26] D. Coeurjolly, R. Klette, A comparative evaluation of length estimators of digital curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004) 252–257.

- [27] I. Debled-Renesson, J.-P. Reveillès, A linear algorithm for segmentation of digital curves, *International Journal of Pattern Recognition and Artificial Intelligence* 9 (1995) 635–662.
- [28] R. O. Duda, P. E. Hart, Use of hough transformation to detect lines and curves in pictures, *Communications of the ACM* 15 (1) (1972) 11–15.
- [29] D. Luo, P. Smart, J. E. S. Macleod, Circular hough transform for roundness measurement of objects, *Pattern Recognition* 28 (11) (1995) 1745–1749.
- [30] S.-C. Pei, J.-H. Horng, Circular arc detection based on hough transform, *Pattern recognition letters* 16 (1995) 615–625.
- [31] U. M. Landau, Estimation of a circular arc centre and its radius, *Computer Vision, Graphics and Image Processing* 38 (1987) 317–326.
- [32] S. M. Thomas, Y. T. Chan, A simple approach to the estimation of circular arc centre and its radius, *Computer Vision, Graphics and Image Processing* 45 (1989) 362–370.
- [33] M. Berman, Large sample bias in least squares estimators of a circular arc centre and its radius, *Computer Vision, Graphics and Image Processing* 45 (1989) 126–128.
- [34] Z. Drezner, S. Steiner, G. O. Wesolowsky, On the circle closest to a set of point, *Computers and operations research* 29 (2002) 637–650.
- [35] T. Roussillon, I. Sivignon, L. Tougne, Test of circularity and measure of circularity for digital curves, in: *International Conference on Image Processing, Computer Vision and Pattern Recognition*, 2008, pp. 518–524.
- [36] N. Megiddo, Linear-time algorithms for linear programming in \mathbb{R}^3 and related problems, *SIAM Journal on Computing* 12 (4) (1984) 759–776.

- [37] N. Megiddo, Linear programming in linear time when the dimension is fixed, *SIAM Journal on Computing* 31 (1984) 114–127.
- [38] K. Q. Brown, Voronoi Diagrams from Convex Hulls, *Information Processing Letters* 9 (1979) 223–228.
- [39] F. P. Preparata, M. I. Shamos, *Computational geometry : an introduction*, Springer, 1985.
- [40] M. de Berg, M. van Kreveld, M. Overmars, O. Scharzkopf, *Computation geometry, algorithms and applications*, Springer, 2000.
- [41] D. Acketa, J. Zunić, On the maximal number of edges of convex digital polygons included into a $m \times m$ -grid, *Journal of Combinatorial Theory, Series A* 69 (2) (1995) 358–368.
- [42] A. A. Melkman, On-line construction of the convex hull of simple polygon, *Information Processing Letters* 25 (1987) 11–12.
- [43] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuezle, D. Madigan, A statistical, nonparametric methodology for document degradation model validation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000) 1209–1223.
- [44] T. Hirata, A unified linear-time algorithm for computing distance maps, *Information Processing Letters* 58 (3) (1996) 129–133.
- [45] T. Roussillon, I. Sivignon, L. Tougne, Robust decomposition of a digital curve into convex and concave parts, in: *The 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.