



Digital Straight Segment Filter for Geometric Description

Rémi Decelle¹(✉), Phuc Ngo¹, Isabelle Debled-Rennesson¹, Frédéric Mothe²,
and Fleur Longuetaud²

¹ CNRS, LORIA, UMR 7503, Université de Lorraine, 54506 Nancy,
Vandœuvre-lès-Nancy, France
remi.decelle@loria.fr

² AgroParisTech, INRAE, SILVA, Université de Lorraine, 54000 Nancy, France

Abstract. In this paper, an algorithmic scheme is proposed to estimate different local characteristics of image structures using discrete geometry tools. The arithmetic properties of Digital Straight Lines and their link with the Farey sequences allow the introduction of a new directional filter. In an incremental process, it provides local geometric information at each point in an image, such as the length, orientation and thickness of the longest Digital Straight Segment passing through that point. Experiments on binary and grayscale images are proposed and show the interest of this tool. Comparisons to a well-known morphological filter for grayscale images are also presented.

Keywords: Farey sequences · Stern-Brocot tree · Digital straight segment · Directional filter · Orientation field · Thickness

1 Introduction

Image processing applications often require computing, analyzing and studying the characteristics of objects contained in digital images. In the field of digital geometry [4], new mathematical definitions of basic geometric objects are introduced to better fit these discrete data, in particular, the notion of digital straight segments (DSS) [4, 12]. DSS has been used in many different contexts to study and analyze the geometrical characteristics of digital curves extracted from contour of objects in digital image [2, 6, 7, 10, 11], such as detection of the convex and concave parts of the shape, the discrete geometric estimators: length, tangent, curvature estimators, dominant point detection, ...

In this paper we are interested in computing the local geometric characteristics at every pixels in image, not only object contours, using the discrete tools. More precisely, based on the arithmetic properties of DSS and their link with the Farey sequences, we develop a new directional filter from which different

This research was made possible by support from the French National Research Agency, in the framework of the project TreeTrace, ANR-17-CE10-0016.

geometric features at each image point can be estimated, such as the length and directional features, the thickness map and the distance to object boundary.

In the context of directional filters using discrete geometry and mathematical morphology tools, several works have been proposed [8, 9, 13, 14, 17]. Van Herk has presented in [17] an efficient algorithm for local min/max filters using linear structuring element (SE). Soille *et al.* [13] proposed later a generalized implementation of this method for erosion and dilation along discrete line at arbitrary angles. The method was promising, however the obtained results are not translation invariant (TI) to the image frame. Indeed, depending on the position of SE along the line, the obtained results can slightly vary from one position to another. To tackle this issue, Soille and Talbot [14] proposed a TI method allowing to efficiently compute the different morphological operations of grayscale images. However, no implementation is given to reproduce the proposed method. Still in [14], an application of orientation fields is presented. For this, a segment of fixed length is used for the local orientation estimation. The obtained result depends strongly on this length parameter. Using the path operator from mathematical morphology, Merveille *et al.* have introduced in [8, 9] a non-linear operator, called Ranking the Orientation Responses of Path Operators (RORPO), for 2D and 3D curvilinear structure analysis. RORPO is well adapted to extract the characteristic of objects with intrinsic anisotropy, and has been applied as a curvilinear filter in segmentation framework of blood vessels in medical imaging.

Inspired by Soille and Talbot work [14] and to overcome the limitations of their method, we introduce a new directional filter using DSS. More precisely, we consider the DSS of adaptive lengths to exploit all possible directional spaces around a pixel and report the longest segments passing through it. Moreover the thickness of the selected DSS is also considered in our proposed method. These length and thickness of the segments are used to extract the local geometric features of pixels as they contain relevant information for the pixel description. This study leads to the proposal of an exact numerical and incremental algorithm allowing to efficiently compute these local features. Applications to the grayscale images and comparisons to a morphological filter are also detailed in the paper.

In the next section we recall definitions and results useful in our study. The Sect. 3 presents the proposed method on binary images and the deduced local geometric features are illustrated. The approach for grayscale images is presented in Sect. 4. Applications and results are shown in Sect. 5.

2 Digital Straight Lines and Stern-Brocot Tree

2.1 Digital Straight Lines [12]

Definition 1. A digital straight line (DSL), denoted by $\mathcal{D}(a, b, \mu, \omega)$, with a, b, μ and ω integer numbers and $\gcd(a, b) = 1$, is the set of points $(x, y) \in \mathbb{Z}^2$ verifying $\mu \leq ax - by < \mu + \omega$.

ω is called the *thickness* of the DSL. If $\omega = \max(|a|, |b|)$, the DSL is 8-connected and is named *naive DSL*, and denoted $\mathcal{D}(a, b, \mu)$. If $\omega > |a| + |b|$, the

DSL is named *thick DSL*. A (naive) *digital straight segment* (DSS) is a finite connected part of a (naive) DSL.

A naive DSS is called *symmetric* for a point (x, y) iff (x, y) belongs to it and if the DSS contains the same number of elements on both sides of (x, y) .

It can be noticed that for given values of a, b and μ , the set of naive DSL $\mathcal{D}(a, b, \mu + i \max(|a|, |b|))$, with $i \in \mathbb{Z}$, tessellates \mathbb{Z}^2 (see Fig. 1).

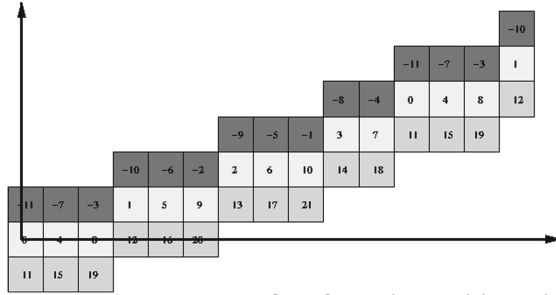


Fig. 1. Naive segments of 15 pixels for $x \in [0, 14]$ of $\mathcal{D}(4, 11, 0)$ (white), $\mathcal{D}(4, 11, 11)$ (light gray), $\mathcal{D}(4, 11, -11)$ (dark gray). The thick DSS $TDSS_{1,1}(4, 11)$ contains all the pixels of the figure for $x \in [0, 14]$. The value $4x - 11y$ labels each pixel (x, y) .

In the rest of this paper, we study the neighborhood of naive segments and we extend a DSS by stacking its closest naive DSS to obtain a thick DSS. In this way, we define on $[0, l] \times [0, L]$ in \mathbb{Z}^2 a *thick DSS of indices k, j and characteristics (a, b)* , noted $TDSS_{k,j}(a, b)$, as the set of points $(x, y) \in [0, l] \times [0, L]$ belonging to $\mathcal{D}(a, b, -k \max(|a|, |b|), (1 + k + j) \max(|a|, |b|))$ with k and j in \mathbb{N} . k and j are the number of naive DSS of characteristics (a, b) that could be stack on one side or on the opposite side (according to current octant) of the naive DSS of $\mathcal{D}(a, b, 0)$. This naive DSS is named the *seed* of the $TDSS$.

2.2 Stern-Brocot Tree and Farey Sequence [3, 15]

Constructing the set of reduced positives fractions $\frac{a}{b}$ can be done iteratively. From two successive fractions $\frac{a}{b}$ and $\frac{a'}{b'}$, we insert a new fraction $\frac{a+a'}{b+b'}$. The *Stern-Brocot tree* is created by starting with $0 = \frac{0}{1}$ and $1 = \frac{1}{1}$ (see Fig. 2).

Definition 2. *The Farey sequence of order $n \in \mathbb{N}^*$, denoted by \mathcal{F}_n is the set of reduced fractions between 0 and 1 for which denominators are lower or equal to n .*

Some examples of Farey sequences: $\mathcal{F}_2 = \{\frac{0}{1}, \frac{1}{2}, \frac{1}{1}\}$, $\mathcal{F}_3 = \{\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}\}$. The number of DSS in the first octant in a grid of size $N \times N$ is exactly the number of elements of Farey sequence of order N (see Berenstein et al. [1] and Koplowitz et al. [5]). The Farey sequence of order N is a subtree of Stern-Brocot tree.

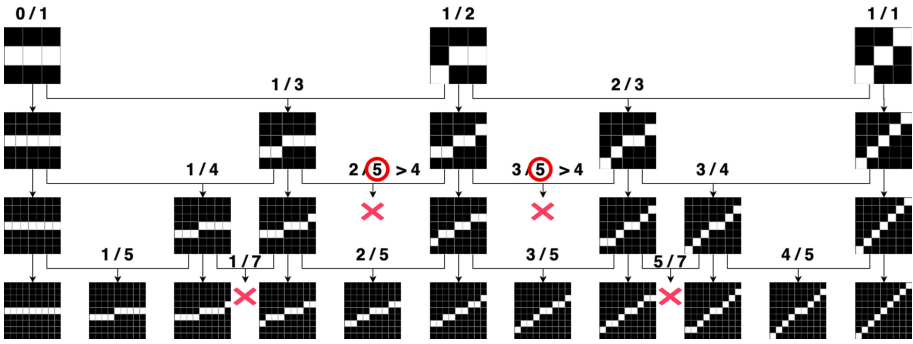


Fig. 2. Construction of DSS according to the first four levels Stern-Brocot tree.

3 DSS Filter on Binary Images

As described in the introduction, we are interested in structures close to DSS but with different lengths and widths. By using Stern-Brocot tree, we can take the advantage of constructing DSS of variable lengths and widths to retrieve more precise features. We will first describe our approach in binary images, where objects under consideration are white and the background is black.

3.1 Proposed Method

The method is described in Algorithm 1 and hereafter we summarize it. Let be p the pixel under consideration of a binary image I , the aim of the presented method is to obtain the longest naive DSS, centered in p , with its pixels in the same color than p . We can, without loss of generality, consider the first octant, i.e. the Stern-Brocot tree goes from $0/1$ up to $1/1$.

The method is as follows from the level n to the level $n + 1$. The considered set of fractions issued from Stern-Brocot tree at the level n is denoted by T_n (T in Algorithm 1). The segments which can be extended are stored in \mathcal{S} . At the start of each iteration, \mathcal{S} is set to T_n (all segments may be extended). For each a/b in T_n , we create the DSS of $\mathcal{D}(a, b, \mu)$ with $2n + 1$ pixels (all μ are considered). If the segment cannot be extended, i.e. one of its points belongs to the background, then (a, b, μ) is removed from \mathcal{S} . Once all elements of T_n have been checked, if $\mathcal{S} = \emptyset$, meaning that no segment can be extended, the algorithm stops. Otherwise, T_{n+1} is built according to T_n and \mathcal{S} , and then we proceed to the iteration $n + 1$.

At each iteration, for a given n , the set T_n contains the fractions of the Farey sequence of order $n + 1$ for which the corresponding DSS could be extended. The incremental process to construct the considered fractions level $n + 1$ from the level n of Stern-Brocot tree is described in Algorithm 2. The slopes of DSS, kept in \mathcal{S} at the previous iteration, permit to add in T_{n+1} only the useful fractions for which DSS could be extended at iteration $n + 1$.

Algorithm 1: Computation of the longest naive DSS at a given point (first octant)

```

Input: An image  $I$  and a pixel  $p(x, y)$  of  $I$ 
Output: The set  $\mathcal{S}$  of all longest naive DSS centered at  $p$ 
1  $T \leftarrow \{0/1, 1/1\}$  // Stern-Brocot sequence represented by fractions  $a/b$ 
2  $n \leftarrow 0$ 
3 repeat
4    $\mathcal{S} \leftarrow \emptyset$ 
5    $T' \leftarrow \text{increment}(T, \mathcal{S})$  // See Algo. 2
6    $n \leftarrow n + 1$ 
7    $cut \leftarrow 0$ 
8   foreach  $a/b \in T'$  do // Compute for each node of  $T'$ 
9     for  $\mu \leftarrow -b + 1$  to 0 do // This loop is applied for small  $n$ 
10       $\mathcal{S}_n \leftarrow (a, b, \mu, n)$  // Characteristics of a DSS of  $\mathcal{D}(a, b, \mu)$ 
11      if  $b < n + 2$  then // Case  $(a, b, \mu, n - 1) \in \mathcal{S}$  at the level  $n - 1$ 
12         $L_n \leftarrow \{(n, \frac{a*n-\mu}{b}), (-n, -\frac{a*n-\mu}{b})\}$  // End-points of the DSS
13      else // Case  $b = n - 2$   $(a, b, \mu, n - 1) \notin \mathcal{S}$  at the level  $n - 1$ 
14         $L_n \leftarrow$  all the  $2n + 1$  pixels of the DSS centered in  $p$ 
15       $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_n$ 
16      foreach  $(i, j) \in L_n$  do
17         $q \leftarrow (x + i, y + j)$ 
18        if  $q \notin I$  or  $I(p) \neq I(q)$  then
19           $cut \leftarrow cut + 1$ 
20           $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{S}_n$ 
21          break
22    $T \leftarrow T'$ 
until  $cut \neq \text{size}(T)$ 
24 Return  $\mathcal{S}$ 

```

Algorithm 2: Stern-Brocot incrementation (first octant)

```

Input: A Stern-Brocot sequence  $T$  at level  $n$  and the set  $\mathcal{S}$  of extensible DSS
Output: The Stern-Brocot sequence  $T'$  at level  $n + 1$ 
1  $T' \leftarrow \emptyset$ 
2 for  $i \leftarrow 0$  to  $\text{size}(T) - 1$  do
3    $node \leftarrow a_i/b_i$  // The current node in  $T$ 
4    $prev \leftarrow a_{i-1}/b_{i-1}[\text{size}(T)]$  // The previous node of  $node$  in  $T$ 
5    $next \leftarrow a_{i+1}/b_{i+1}[\text{size}(T)]$  // the next node of  $node$  in  $T$ 
6   if ( $prev$  or  $next$  or  $node$  is the slope of a DSS in  $\mathcal{S}$ ) then
7      $\lfloor$  Insert  $a_i/b_i$  in  $T'$  // Keep the current node in the next level  $T'$ 
8      $p \leftarrow a_i + a_{i+1}$ 
9      $q \leftarrow b_i + b_{i+1}$ 
10    if ( $q \leq n + 2$ ) then
11       $\lfloor$  Insert  $p/q$  in  $T'$  // Insert the node to the next level  $T'$ 
12 Return  $T'$ 

```

Figure 2 shows the constructed DSS at the first four levels of the tree (with $\mu = 0$) according to the incremental Algorithm 2. At the level three, we only consider the fraction of the Farey sequence of order four then both $2/5$ and $3/5$ are not inserted at this level but at the level four.

Figure 3 illustrates some steps of the method for one pixel. On the left side of the four subfigures, the red color shows the covered area by DSS at different levels. At each step, extensible DSS are kept. On the right side, we show on a circle the fractions of Stern-Brocot tree. Green lines indicate extendable DSS

with the corresponding fractions. Red lines indicate new DSS which cannot be extended. And blue lines are DSS required for building new DSS at the next level. The tree starts at the level 1, with 8 fractions: $0/1, 1/2, \dots, -1/1$. All DSS can be extended. We move to the level 2. At this level, some DSS are not extendable, we removed them and move to the level 3. The algorithm stops at the level 6 (meaning DSS of length 13 pixels) where only the DSS $\mathcal{D}(5, 4, -4)$ is still extensible. This last DSS is shown in Fig. 4.

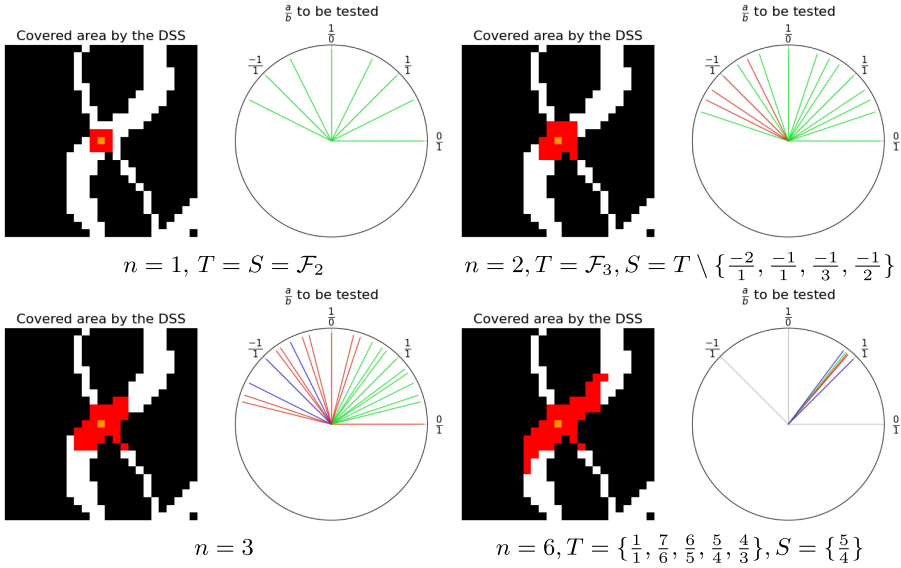


Fig. 3. Step-by-step method for binary image. In green the extendable fractions. In red the rejected ones. In blue those that are required to build new fractions. (Color figure online)

3.2 Geometric Description

At the end of the Algorithm 1, for a given point p in I , we get a value of n and the set \mathcal{S}_n of all the DSS with $2n + 1$ pixels which fit the object. From this set, the following features can be retrieved.

Local Elongation. The elongation L at p is the length of the longest final DSS. We use the simple estimator, $length(DSS) = N_e + \sqrt{2}N_o$, with N_e et N_o the number of even and odd codes in the freeman code of the DSS. This feature indicates if locally the object is elongated. A high value indicates a straight and long shape.

Local Orientation and Direction. Another feature is the local orientation θ of the shape at p and the local vector direction Δ . It is computed by the average of the local orientations obtained in \mathcal{S}_n at the final step of the method.

Orthogonal Span. By constructing, pixel by pixel in both directions, the orthogonal DSS of $\mathcal{D}(-b, a, \mu)$, we can extract the number of pixels ND or the length D of this DSS which reach object's border.

Directional Thickness. The directional thickness of the local shape at the point p . To obtain this feature, we thicken each DSS in \mathcal{S}_n according to our definition of $TDSS_{k,j}(a, b)$ (see Sect. 2). The directional thickness DT is then defined as the sum of the maximum of j and k for which the $TDSS$ is included in I .

Those geometric features are visible in Fig. 4 for DSS of $\mathcal{D}(5, 4, -4)$ with 13 pixels previously extracted (see Fig. 3). The length is $L = 15.7$, the orientation is $\theta = \tan^{-1}(\frac{5}{4}) \approx 51^\circ$, the orthogonal span is $ND = 6$, $D = 6.65$ and finally the directional thickness is $DT = 0 + 1 = 1$.

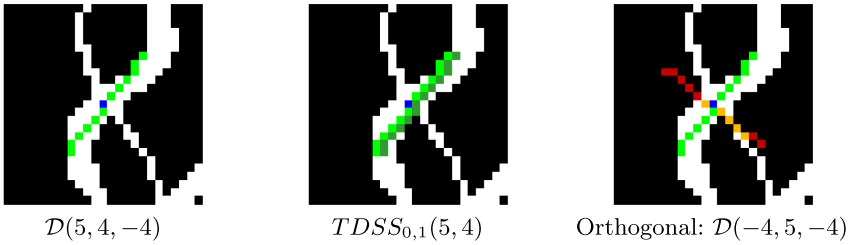
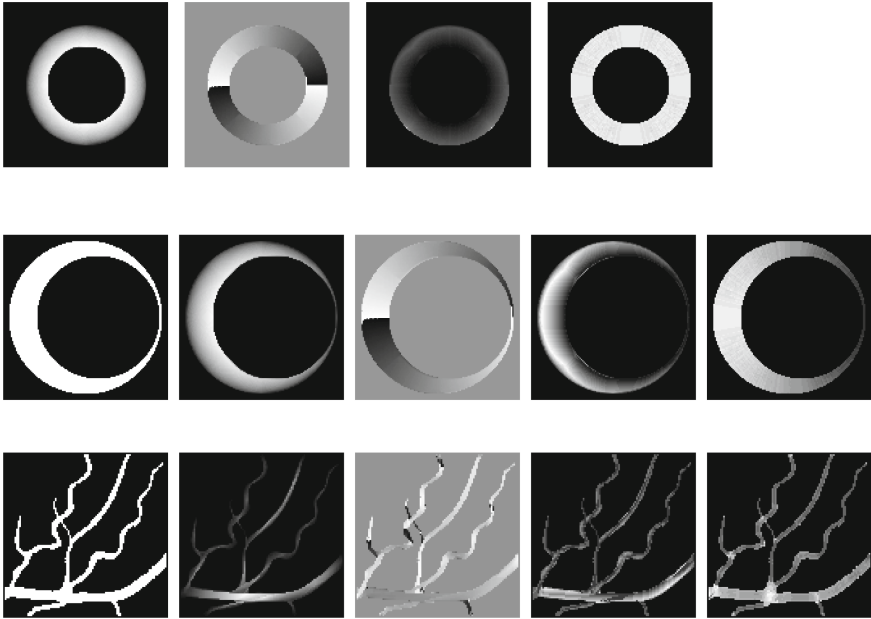


Fig. 4. On the left, $\mathcal{D}(5, 4, -4)$ from last step of the algorithm (see Fig. 3). The blue pixel is the one under consideration. Green pixels are those of the DSS which are valid. On the right, red pixels are outside of the object, orange are inside of it. (Color figure online)

The Table 1 shows these geometric features for some images. The first image is an annulus, the second one is two circles with different center points and the last one represents vessels. At first, it can be observed that the length map is uniform for the annulus as expected. The length is higher close to the centre. It is the same for the two circles. For vessel image, elongated vessels have high value. For orientation map, the black/white transition corresponds to the change between π and 0. Note that the angle θ is considered for the orientation; e.g., the naive DSL $\mathcal{D}(1, 3, 0)$ and $\mathcal{D}(-1, -3, 0)$ have the same orientation (which is around 18 degree). For this, we use arctan of a/b . This explains the discontinuity between 0 (black pixel) and π (white pixel).

There are some artefacts due to the mean of orientation. The obtained orientation is close to the theoretical one. For the directional thickness we show that it is higher outside the annulus. It is due to small length of DSS at the end, which lead to being able to thicken easily. Small number of pixels reduces the probability to be outside the object when thickening. At last, the orthogonal span has to be high when the object is width or when it is in an intersection as it can be shown in vessel image.

Table 1. Extracted geometric features from DSS filter



4 DSS Filter on Grayscale Images

In this section, we focus on a similar approach as the one developed by Soille and Talbot [14] for orientation field. Therefore, we are interested in local orientation of image structure with different lengths and widths. Similar approach to the binary case to take account different lengths and widths is used.

As the binary case, we start with first level of the Stern-Brocot tree and with small segments. We compute the erosion by it of the image. We thicken the segment if possible, and compute again the erosion by it. Then, we move to the next level of the tree. We stop until we reach a certain level n . We sum up the results of each erosion. Algorithm 3 sums up the method. Erosons should be used for brighter image structure than background. For darker image structure than the background, the invert image $J = 255 - I$ combined with erosions should be used. For each a/b , we denoted $B_{i,j,k}$ the $TDSS_{k,j}(a, b)$ for which the seed DSS contains $2i + 1$ points, we compute the image $Y_{a,b}$ equal to:

$$Y_{a,b} = \frac{1}{n - \alpha + 1} \sum_{i=\alpha}^n \left[\sum_{k=0}^{i-1} \varepsilon_{B_{i,k,0}}(I) + \sum_{j=1}^{i-1} \varepsilon_{B_{i,0,j}}(I) \right] \tag{1}$$

where ε is the morphological erosion and $\alpha = \max(\max(|a|, |b|) - 1, 1)$. We introduce a coefficient of normalization in order to take account that the number

Algorithm 3: Grayscale method (first octant)

```

Input: An image  $I$ ,  $n$  the maximum number of points of the DSS
Output: An array of images  $Y$ 
1  $Y \leftarrow$  matrix of size  $(2n+1, 2n+1)$  // The element at position  $(a, b)$  is the image  $Y_{a,b}$ 
2  $A \leftarrow (0, 1, 1)^T$ 
3 for  $i \leftarrow 0$  to  $n$  do // Iterate over points number up to  $n$ 
4     foreach  $a/b \in \mathcal{F}_{i+2}$  do // Iterate over Farey sequence of order  $i + 2$ 
5          $B \leftarrow TDSS_{0,0}(a, b)$  with  $2i+1$  points
6          $E \leftarrow \varepsilon_B(I)$  // Compute erosion of  $I$  by the DSS  $B_{i,0}$ 
7          $E^+ \leftarrow E$ 
8          $E^- \leftarrow E$ 
9          $Y_{a,b} \leftarrow Y_{a,b} + E$ 
10        for  $k \leftarrow 1$  to  $i$  do // Iterate over the segment width
11             $E^+ \leftarrow \varepsilon_A(E^+)$  // Erosion by  $TDSS_{0,k}(a, b)$ 
12             $E^- \leftarrow \varepsilon_{\bar{A}}(E^-)$  // Erosion by  $TDSS_{k,0}(a, b)$ 
13             $Y_{a,b} \leftarrow Y_{a,b} + E^+$ 
14             $Y_{a,b} \leftarrow Y_{a,b} + E^-$ 
15 Return  $Y$ 

```

of DSS depends on a/b . Indeed, if a/b is defined at a level p , it will still be there for all the next levels. For instances, $1/2$ will be considered n times (DSS with $3, 5, 7, \dots, 2n + 1$ points), $1/3$ will be considered $n - 1$ times (DSS with $5, 7, \dots, 2n + 1$ points), and so on.

Figure 5 presents, for a pixel in an image of tree rings (last image, bottom right, in the figure), different steps of the proposed process if we only consider length variations. The pixel under consideration is the one located at the center of the image (green cross).

The ten first images of the Fig. 5 highlight the results of each erosion, i.e. erosion of I by $B_{i,0,0}$ with i varying from 1 to 10. Axes are the parameters a and b from DSS. For each couple (a, b) from Farey sequences, we plot the gray value resulting from the erosion of I by $B_{i,0,0}$. In each image, the pixel with the highest value is framed with red color.

The image before the last one shows for each couple (a, b) the sum of each erosion for the different lengths. The orientation is assumed to be the one with the highest gray value (red pixel).

We can see that the a/b for which the maximum is reached changes several times depending on the value of i , i.e. the number of points in the segment. By choosing $i = 9$ (i.e. 19 points), the orientation seems correct, but by choosing $i = 10$ (i.e. 21 points) points, the direction is not the expected one (the corresponding segment is represented in red in the last image). However, the direction resulting from the sum is correct (the corresponding segment is represented in green in the last image).

4.1 Fast Implementation

For the first octant, with $A = (1\ 1\ 0)^T$, we have the relation $B_{i,j,k+1} = B_{i,j,k} \oplus A$, where \oplus is the morphological dilatation. Knowing the property $\varepsilon_{A \oplus B}(I) =$

$\varepsilon_A(\varepsilon_B(I))$, the relation becomes $\varepsilon_{B_{i,j,k+1}}(I) = \varepsilon_A(\varepsilon_{B_{i,j,k}}(I))$. The relation holds for $B_{i,j,k} \subset B_{i,j+1,k}$ by taking the symmetric of A , i.e. $\check{A} = (0 \ 1 \ 1)^T$. This relation is shown by the loop line 10 in Algorithm 3. A is adjusted according to the octant. Moreover, for elements $B_{i,0,0}$ we can take the advantage of the algorithm developed by [14] (line 6 in Algorithm 3).

4.2 Orientation Feature

We can consider three variants of the proposed method. First, we consider all possible lengths and thicknesses (see Eq. 1). Second variant, we consider all thicknesses but the length is fix to n (in Algorithm 2 line 3 is removed and $i = n$). Third variant, we consider all lengths but the thickness is fix to 0 (loop line 10 is removed). The Eq. 1 is then respectively for the second and third variants:

$$Y_{a,b} = \sum_{k=0}^{n-1} \varepsilon_{B_{n,k,0}}(I) + \sum_{j=1}^{n-1} \varepsilon_{B_{n,0,j}}(I) \quad Y_{a,b} = \frac{1}{n - \alpha + 1} \sum_{i=\alpha}^n \varepsilon_{B_{i,0,0}}(I)$$

In all three cases, we are back to estimate only the dominant orientation for each point x since we have images rely only on a/b . Inspired by [14], we define at each point x the bright orientation as:

$$Dir^+(x) = \left\{ \tan^{-1}\left(\frac{a_i}{b_i}\right), Y_{a_i,b_i}(x) \geq Y_{a_j,b_j}(x) \forall (a_i, b_i) \neq (a_j, b_j) \right\}$$

We introduce an additional quantity to allow us to determine which orientations to choose.

$$G^+(x) = \max_{a,b} (Y_{a,b}(x)) - \min_{a,b} (Y_{a,b}(x))$$

$G^+(x)$ can be interpreted as the strength of the bright structures at the point x . The orientation Dir^- for dark structures and their strength G^- are computed on the same principle as G^+ but using $J = 255 - I$ instead of I . Finally the orientation is computed as follows:

$$\theta(x) = \begin{cases} Dir^+(x) , & \text{if } G^+(x) \geq G^-(x) \\ Dir^-(x) , & \text{otherwise} \end{cases}$$

It can be noticed that during the process of orientation, we can extract filtered images. These are images resulting from the calculation of $\max_{a,b}(Y_{a,b}(x))$ for both I and J (see Fig. 7).

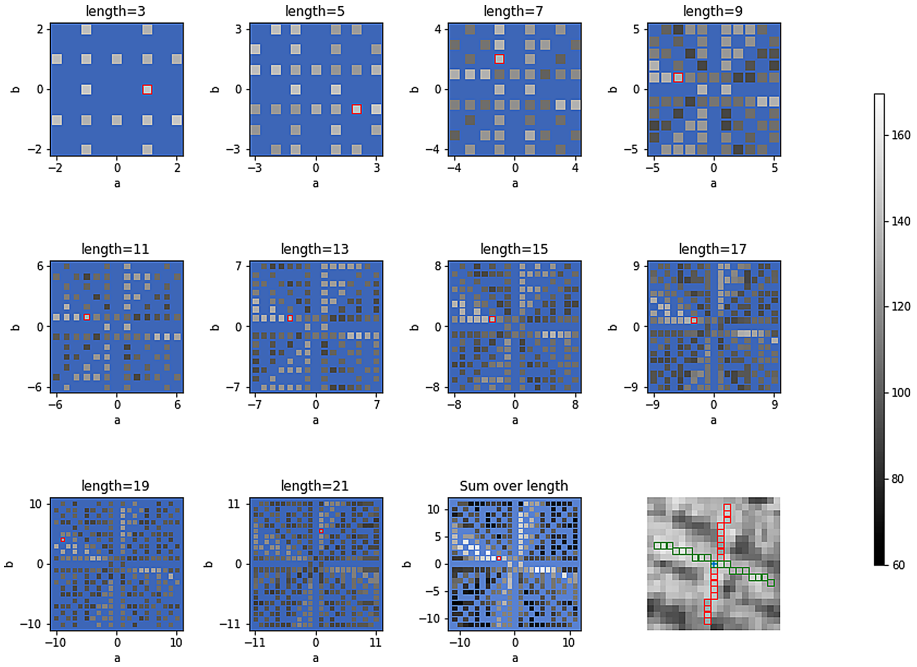


Fig. 5. Illustration of the approach with length variations. (Color figure online)

5 Results and Discussion

A deep analysis of grayscale method is out of range of this article. Nonetheless, we provide a comparison with the proposed method by Soille and Talbot [14]. To do this, we first consider circular objects (see Fig. 6) since the ground truth can easily be determined. This information is useful to demonstrate the efficiency of the proposed method. For the comparison, we computed the deviation between a ground truth and output using the formulae proposed by Turrone et al. [16]:

$$RMSD = \sqrt{\frac{\sum_x d^2(\theta_x, \theta_x^G)}{\text{Number of pixels}}}, \quad d(\theta_1, \theta_2) = \begin{cases} \theta_1 - \theta_2 & \text{if } -\frac{\pi}{2} \leq \theta_1 - \theta_2 < \frac{\pi}{2} \\ \pi + \theta_1 - \theta_2 & \text{if } \theta_1 - \theta_2 < -\frac{\pi}{2} \\ \pi - \theta_1 + \theta_2 & \text{if } \theta_1 - \theta_2 \geq \frac{\pi}{2} \end{cases}$$

with θ_x^G theoretical orientation at x , θ_x its estimated orientation. The proposed method by [14] provides a RMSD of 0.2809. For our methods, we have respectively for thick variations a value of 0.3111, length variations a value of 0.2436 and both variations a value of 0.2317. For each methods, we set $n = 10$ (i.e. $\lambda = 21$ for [14]). Including thickness does not improve orientation, but including length (with or without thickness) improved the orientation.

Figure 7(a) shows the circles in Fig. 6 corrupted by the additive Gaussian noise of mean 0 and standard deviation 0.1, on top right a zoom on the central part of the image). Figure 7(b) and (c) are results filtered by both [14] and our

approach. Since Soille and Talbot fix the length parameter, their approach fails to filter areas where this length is not suitable. As the orientation, we fix $n = 10$ (i.e. $\lambda = 21$ for [14]).

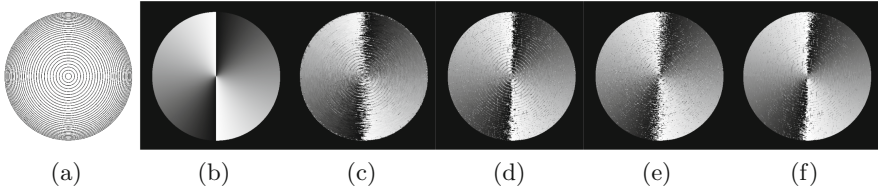


Fig. 6. Results for our method and [14]. (a) Input image of concentric circles. (b) Ground truth. (c) [14] method. (d) Thick variations. (e) Length variations. (f) Both variations.

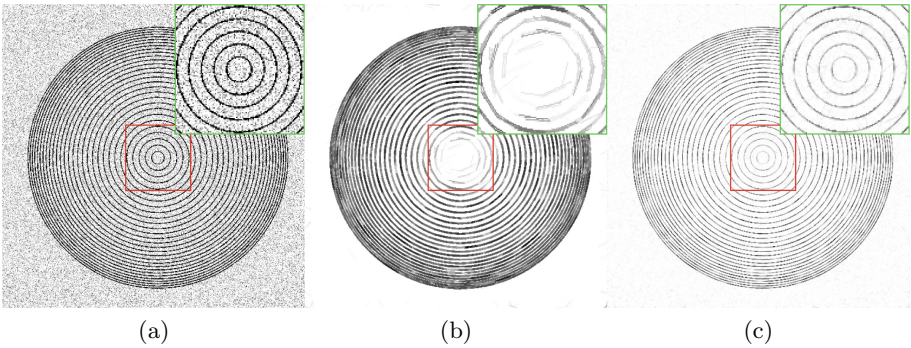


Fig. 7. (a) Noisy image, (b) filtering with [14] of image (a), (c) filtering with our method (length variations) of image (a). On the top right a zoom on the central part of the image.

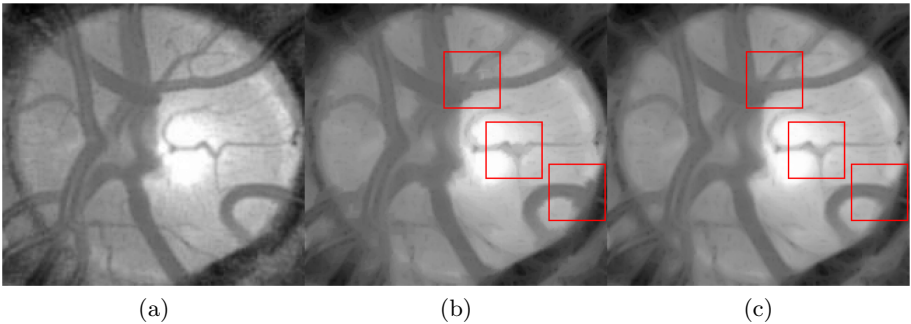


Fig. 8. (a) Original image, (b) filtering with [14], (c) filtering with our method (length variations).

Figure 8 shows an other example. From left to right, we can see the original image, the resulting images filtered by [14] and our approach. We can see that our method preserves better the circular structures and is smoother at intersections.

6 Conclusion

In this paper, we firstly presented a method able to estimate geometric features on binary images such as the local elongation, local orientation, directional thickness and orthogonal distance. These features are based on digital straight segments (DSS) according to Farey sequences. By taking the advantage of widths and lengths of DSS, we secondly introduced an adaptation of the previous method to grayscale images. Comparisons to the approach proposed in [14] shows its interest. In particular, using the orientation estimation as a directional filtering, our method allows to reduce noise along line segments. It should be noted that such filter is well adapted for images with structures of different lengths such as tree ring and fingerprint images.

Applications of the extracted features for image analysis framework are currently under study. As short-term perspective, we would like to use the method for wood-log-section image analysis. In this context, several problems could be addressed such as pith detection, tree-ring delineation, . . .

In other perspectives, we plan to investigate improvements for selecting optimal orientation among line segments. We are also looking to retrieve pertinent thickness information in grayscale images.

Acknowledgment. This research was made possible by support from the French National Research Agency, in the framework of the project TreeTrace, ANR-17-CE10-0016.

References

1. Berenstein, C.A., Lavine, D.: On the number of digital straight line segments. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(6), 880–887 (1988)
2. Dorksen-Reiter, H., Debled-Rennesson, I.: Convex and concave parts of digital curves. In: Klette, R., Kozera, R., Noakes, L., Weickert, J. (eds.) *Geometric Properties for Incomplete data*. Springer, Dordrecht (2006). https://doi.org/10.1007/1-4020-3858-8_8
3. Hardy, E.W.G.: *An Introduction to the Theory of Numbers* (1989)
4. Klette, R., Rosenfeld, A.: *Digital Geometry - Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, Burlington (2004)
5. Koplowitz, J., Lindenbaum, M., Bruckstein, A.: The number of digital straight lines on an $n \times n$ grid. *IEEE Trans. Inf. Theory* **36**(1), 192–197 (1990)
6. Lachaud, J.-O.: Digital shape analysis with maximal segments. In: Köthe, U., Montanvert, A., Soille, P. (eds.) *WADGMM 2010*. LNCS, vol. 7346, pp. 14–27. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32313-3_2
7. Lachaud, J.-O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. *Image Vis. Comput.* **25**(10), 1572–1587 (2007)

8. Merveille, O., Naegel, B., Talbot, H., Najman, L., Passat, N.: 2D filtering of curvilinear structures by ranking the orientation responses of path operators (RORPO). *Image Process. On Line* **7**, 246–261 (2017)
9. Merveille, O., Talbot, H., Najman, L., Passat, N.: Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 304–317 (2018)
10. Nasser, H., Ngo, P., Debled-Rennesson, I.: Dominant point detection based on discrete curve structure and applications. *J. Comput. Syst. Sci.* **95**, 177–192 (2018)
11. Provençal, X., Lachaud, J.-O.: Two linear-time algorithms for computing the minimum length polygon of a digital contour. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) *DGCI 2009. LNCS*, vol. 5810, pp. 104–117. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04397-0_10
12. Reveillès, J.-P.: *Géométrie discrète, calcul en nombres entiers et algorithmique* (1991)
13. Soille, P., Breen, E.J., Jones, R.: Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(5), 562–567 (1996)
14. Soille, P., Talbot, H.: Directional morphological filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1313–29 (2001)
15. Stern, M.: *Über eine verallgemeinerung der kreistheilung. J. für die reine und angewandte Mathematik* **55**, 193–220 (1858)
16. Turrone, F., Maltoni, D., Cappelli, R., Maio, D.: Improving fingerprint orientation extraction. *IEEE Trans. Inf. Forensics Secur.* **6**(3), 1002–1013 (2011)
17. van Herk, M.: A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognit. Lett.* **13**(7), 517–521 (1992)