3B2v8.06a/w (Dec 5 2003).51c XML:ver:5.0.1 CAG : 1575 Prod.Type:FTPFTP pp.1–8(col.fig.:1–3,5–10) ED:BibhuPrasadBibhuprasad PAGN:NMN SCAN:mouli

ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Graphics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Special Issue: Discrete Geometry for Computer Imagery

Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids

David Coeurjolly*, Loutfi Zerarga

Laboratoire LIRIS, Université Claude Bernard Lyon 1, 43 Bd du 11 novembre 1918, F-69622 Villeurbanne, France

Abstract

On the classical discrete grid, the analysis of digital straight lines (DSL for short) has been intensively studied for nearly half a century. In this article, we are interested in a discrete geometry on irregular grids. More precisely, our goal is to define geometrical properties on irregular isothetic grids that are tilings of the Euclidean plane with different sized axis parallel rectangles. On these irregular isothetic grids, we define digital straight lines with recognition algorithms and a process to reconstruct an invertible polygonal representation of an irregular discrete curve.

Keywords: ■; ■; ■

1. Introduction

When a straight line is digitized on a square grid, we obtain a sequence of grid points defining a digital straight-line segment. This computer representation of such a simple Euclidean object has drawn considerable attention in many applications (drawing [1], shape characterization [2–4], ...). The structure of DSL is now well known and links have been illustrated between DSL and objects from number theory or theory of words (see [5] for a survey on digital straightness). Beyond this characterization, an important task in computer vision consists in the recognition of DSL segments. More precisely, given a set of pixels, we have to decide if there exists a DSL segment that contains the given pixels. Many efficient algorithms exist to implement such a recognition process [6–9]. Based on a digital straight line recognition algorithm, we can also define a segmentation process that decomposes a discrete curve into maximal DSL segments. The next step of the segmentation process is to reconstruct a polygonal curve from the discrete data such that its digitization is equal to the original discrete curve. This process is called an *invertible reconstruction* of a discrete curve [10–12]. The invertible property is an important one in discrete geometry since it allows to convert discrete data to Euclidean ones such that no information is added nor lost.

In this article, we are interested in defining a geometry on irregular isothetic grids. More precisely, we consider grids defined by a tiling of the plane using axis parallel rectangles. Such a grid model includes, for example, the classical discrete grid, the elongated grids [13] and the quadtree based grids [14]. In [15], a general framework has been proposed that defines elementary objects and a digitization framework, the *supercover model*. An important aspect of this general framework is the consistency with classical definitions if the discrete space is considered.

Many applications may benefit from these developments. For example, we can cite the analysis of quadtree

*Corresponding author. Tel.: +33 4 7243 2662; fax: +33 4 7244 1312.
E-mail address: dcoeurjo@liris.cnrs.fr (D. Coeurjolly).

0097-8493/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2005.10.009

compressed shapes, or the use of geometrical properties in objects represented by interval or affine arithmetics (see discussion in [15]). Based on this irregular model, we define digital straight lines with recognition algorithms and a process to reconstruct an invertible polygonal representation of an irregular discrete curve.

Section 2 presents more formal definitions in the irregular grids: adjacency relations, objects, arcs, curves and the supercover model. Based on a definition of the irregular isothetic digital straight lines, we present algorithms to recognize maximal irregular discrete straight segments and to reconstruct invertible polygonal arcs and curves (Section 3). Experiments and results are shown in Section 4.

2. Preliminary definitions

2.1. The irregular isothetic model

First of all, we define an *irregular isothetic grid*, denoted $\mathbb{I}$, as a tiling of the plane with isothetic rectangles. In this framework, the rectangles have not necessarily the same size but we can note that the classical digital space is a particular irregular isothetic grid. In that case, all squares are centered in $\mathbb{Z}^2$ points and have a border size equal to 1. Fig. 1 illustrates some examples of irregular isothetic grids. A rectangle of an isothetic grid is called a *pixel*. Each pixel P is defined by its center $(x_P, y_P) \in \mathbb{R}^2$ and a size $(l_P^x, l_P^y) \in \mathbb{R}^2$. Before we introduce objects and straight lines in such grids, we need adjacency relations between pixels.

Definition 1 (*ve-adjacency, e-adjacency*). Let P and Q be two pixels. P and Q are *ve-adjacent* if:

$$|x_P - x_Q| = \frac{l_P^x + l_Q^x}{2} \quad \text{and} \quad |y_P - y_Q| \leqslant \frac{l_P^y + l_Q^y}{2},$$

or

$$|y_P - y_Q| = \frac{l_P^y + l_Q^y}{2} \quad \text{and} \quad |x_P - x_Q| \leqslant \frac{l_P^x + l_Q^x}{2}.$$

P and Q are *e-adjacent* if we consider an exclusive "or" and strict inequalities in the above *ve-adjacent* definition.

In the following definitions, we use the notation *k-adjacency* in order to express either the *ve-adjacency* or the *e-adjacency*. Using these adjacency definitions, several basic objects can be defined:

Definition 2 (*k-path*). Let us consider a set of pixels $\mathscr{E} = \{P_i, i \in \{1, \dots, n\}\}$ and a relation of k-adjacency. $\mathscr{E}$ is a k-path if and only if for each element P_i of $\mathscr{E}$, P_i is k-adjacent to P_{i-1}.

Definition 3 (*k-object*). Let $\mathscr{E}$ be a set of pixels, $\mathscr{E}$ is a k-object if and only if for each couple of pixels (P, Q) belonging to $\mathscr{E} \times \mathscr{E}$, there exists a k-path between P and Q in $\mathscr{E}$.

Definition 4 (*k-arc*). Let $\mathscr{E}$ be a set of pixels, $\mathscr{E}$ is a k-arc if and only if for each the element of $\mathscr{E} = \{P_i, i \in \{1, \dots, n\}\}$, P_i has exactly two k-adjacent pixels, except P_1 and P_n which are called the extremities of the k-arc.

Definition 5 (*k-curve*). Let $\mathscr{E}$ be a set of pixels, $\mathscr{E}$ is a k-curve if and only if $\mathscr{E}$ is a k-arc and $P_1 = P_n$.

If we consider pixels such that $l_P^x = l_P^y = 1$ and $(x_P, y_P) \in \mathbb{Z}^2$ (i.e. a 2D digital space), all these definitions coincide with the classical ones [16,17]. More precisely, the *ve*-adjacency (resp. *e*-adjacency) is exactly the 8-adjacency (resp. the 4-adjacency). In the following, we only consider geometrical properties of such objects. A complete topological analysis of k-curves and k-objects is not addressed here.

2.2. Supercover model on the irregular isothetic grids

Before defining the digital straight lines on the irregular isothetic grids, we have to consider a digitization model. In the following, we choose to extend the supercover model. This model was first introduced by Cohen-Or and Kaufman in [18] on the classical discrete grid and then widely used since it provides an analytical characterization of basic supercover objects (e.g. lines, planes, 3D polygons,...) [19,20].

Definition 6 (*Supercover on irregular isothetic grids*). Let F be an Euclidean object in $\mathbb{R}^2$. The

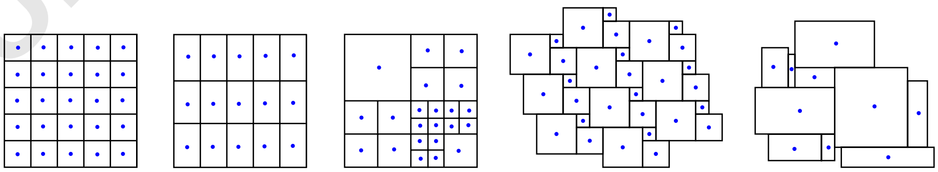

Fig. 1. Examples of irregular isothetic grids: (*from left to right*) the classical discrete grid ($(x_P, y_P) \in \mathbb{Z}^2$ and $l_P^x = l_P^y = 1$), an elongated grid ($l_P^x = \lambda$, $l_P^y = \mu$ and $(x_P, y_P) = (\lambda i, \mu j)$ with $(i,j) \in \mathbb{Z}^2$), a quadtree decomposition (for a cell of level k, $(x_P, y_P) = (m/2^k, n/2^k)$ and $l_P^x = l_P^y = 1/2^{k-1}$ for some $m, n \in \mathbb{Z}$); a unilateral and equitransitive tiling by squares: the size of the biggest square is equal to the sum of the two other square sizes; finally a general irregular isothetic grid.

supercover $\mathbb{S}(F)$ is defined on an irregular isothetic grid $\mathbb{I}$ by

$$\mathbb{S}(F) = \{P \in \mathbb{I} \mid B(P) \cap F \neq \emptyset\} \tag{1}$$

$$= \left\{ P \in \mathbb{I} \mid \exists (x,y) \in F, |x_P - x| \leqslant \frac{l_P^x}{2} \right.$$

$$\left. \text{and } |y_P - y| \leqslant \frac{l_P^y}{2} \right\}, \tag{2}$$

where $B(P)$ is the rectangle centered in (x_P, y_P) of size (l_P^x, l_P^y) (if $l_P^x = l_P^y$, $B(P)$ is the ball centered in (x_P, y_P) of size l_P^x for the L_∞ norm).

Properties of this model are discussed in [15].

Fig. 2 illustrates some examples of the supercover digitization of Euclidean objects. If $\mathbb{I}$ is the classical digital space (i.e. $(x_P, y_P) \in \mathbb{Z}^2$ and $l_P^x = l_P^y = 1$), many links exist between the supercover of an Euclidean straight line and classical digital straight line definitions [5,20]. Since we have not any assumption on the irregular grid, no strong topological property can be stated on the supercover of an Euclidean straight line.

Proposition 1 (*Coeurjolly [15]*). *Let l be an Euclidean straight line and a $\mathbb{I}$-grid, the $\mathbb{S}(l)$ is a single ve-object.*

3. Irregular isothetic digital straight line definition and recognition

3.1. Definitions and IDSL Recognition

Definition 7 (*Irregular isothetic digital straight line*). Let S be a set of pixels in $\mathbb{I}$, S is called a piece of *irregular digital straight line* (IDSL for short) iff there exists an Euclidean straight line l such that:

$$S \subseteq \mathbb{S}(l). \tag{3}$$

In other words, S is a piece of IDSL iff there exists l such that for all $P \in S$, $\mathbb{B}^\infty(P) \cap l \neq \emptyset$.

To detect if $\mathbb{B}^\infty(P) \cap l$ is empty or not, we use the notations presented in Fig. 3. Hence, $\mathbb{B}^\infty(P) \cap l$ is not

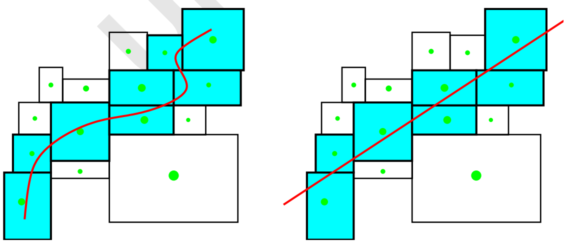

Fig. 2. Illustration of the supercover digitization of a curve (*left*) and of a straight line (*right*).

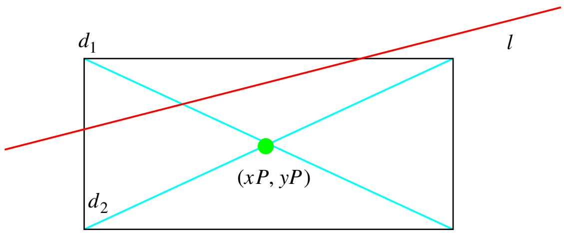

Fig. 3. Notations used to detect if the pixel of center (x_P, y_P) belongs to the supercover of a straight line l (d_1 and d_2 are the diagonals of the rectangle P).

empty iff l crosses either (or both) the diagonals d_1 or d_2 of P.

Without loss of generality, we suppose that l is given by $y = \alpha x + \beta$ with $(\alpha, \beta) \in \mathbb{R}^2$ (an appropriate treatment can be design to handle the straight lines $x = k$ with $k \in \mathbb{R}$). To solve the recognition problem, we use the following statement:

$$\mathbb{B}^\infty(P) \cap l \neq \emptyset \iff l \cap d_1 \neq \emptyset \quad \text{and} \quad \alpha \geqslant 0 \tag{4}$$

$$\text{or} \quad l \cap d_2 \neq \emptyset \quad \text{and} \quad \alpha < 0 \tag{5}$$

During a recognition process, it is convenient to consider the set of Euclidean straight lines whose digitization contains the set of pixels S: if such a set is empty, we can conclude that S is not a discrete straight line segment. In the literature, the set of Euclidean straight lines whose digitization contains S is called the *preimage* of S. Many works have been done concerning the preimage analysis in the classical discrete grid [7,21,22].

Given a pixel P, Eq. (4) can be represented by two inequalities in the (α, β)-parameter space:

$$\mathscr{E}^+(P) = \begin{cases} \alpha\left(x_P - \dfrac{l_P^x}{2}\right) + \beta - y_P - \dfrac{l_P^y}{2} \leqslant 0, \\ \alpha\left(x_P + \dfrac{l_P^x}{2}\right) + \beta - y_P + \dfrac{l_P^y}{2} \geqslant 0. \end{cases} \tag{6}$$

Details on the computation of these inequalities can be found in [15]. If we consider Eq. (5), we may obtain the following inequalities:

$$\mathscr{E}^-(P) = \begin{cases} \alpha\left(x_P - \dfrac{l_P^x}{2}\right) + \beta - y_P + \dfrac{l_P^y}{2} \geqslant 0, \\ \alpha\left(x_P + \dfrac{l_P^x}{2}\right) + \beta - y_P - \dfrac{l_P^y}{2} \leqslant 0. \end{cases} \tag{7}$$

$\mathscr{E}^+(P)$ is defined for $\alpha \geqslant 0$ and $\mathscr{E}^-(P)$ for $\alpha < 0$. We can now define the preimages of a piece of IDSL:

Definition 8 (*Preimages of an IDSL*). Let S be a piece of IDSL, the two preimages $\mathscr{P}^+$ and $\mathscr{P}^-$ of S are given by:

$$\mathscr{P}^+(S) = \bigcap_{P \in S} \mathscr{E}^+(P) \quad \text{and} \quad \mathscr{P}^-(S) = \bigcap_{P \in S} \mathscr{E}^-(P). \tag{8}$$

Hence, the recognition process can be described as follows:

Proposition 2. *Let S be a set of pixels in a $\mathbb{I}$-grid. S is a piece of IDSL iff $\mathscr{P}^+(S) \neq \emptyset$ or $\mathscr{P}^-(S) \neq \emptyset$.*

Using Proposition 2, the recognition of a piece IDSL leads to a linear programming problem: we have to decide whether a linear inequality system has a solution or not. To solve this problem, two different classes of algorithms exist: the IDSL *identification* algorithms which decide if S is an IDSL or not, and the IDSL *recognition* algorithms which return the complete pre-images (maybe empty) of the recognized IDSL. To solve the identification problem, incremental $O(n)$ solutions exist if n is the number of linear constraints (i.e. the number of irregular pixels in our case) [23,24]. To completely describe the preimages, the incremental Preparata and Shamos algorithm [25] may be used whose computational cost is optimal in $O(n \log n)$. In [15], an algorithm based on a linear programming procedure is proposed to recognize IDSL given a set of pixels. This algorithm can also be used to segment an irregular arc, i.e. to decompose the arc into maximal piece of IDSL (see Fig. 4).

The segmentation of a curve gives information concerning the geometry of the curve. In the next section, we detail an algorithm to obtain an invertible polyline from the irregular set of pixels.

3.2. Invertible reconstruction of irregular arcs and curves

In the following, we propose an algorithm to construct an Euclidean polyline from a discrete curve such that its digitization is equal to the original discrete curve. If we consider the supercover digitization model, a polyline $\mathscr{L}$ is an invertible reconstruction of a discrete curve S if it lies inside the discrete curve. More precisely, for each Euclidean point p on $\mathscr{L}$, there exists a pixel P in S such that p belongs to P.

Usually, the reconstruction task is a post-treatment of a DSL segmentation algorithm: first we decompose the discrete curve into maximal DSL, then, for each piece of DSL, we compute a representative Euclidean segment. The main drawback of this approach is that it is difficult to ensure the reversibility of the polyline vertices [10,12]. In the classical discrete grid, Sivignon et al. [11] propose an invertible reconstruction algorithm in which both the recognition and the Euclidean segment extraction are performed at the same time. More precisely, the authors reduce the problem forcing the first extremity of the segments to be inside the discrete curve. Then, they perform an analysis on the preimage of the segment to compute the second extremity.

In the following, we propose a similar algorithm without the computation of the preimages that would have required complex linear programming procedures. The main idea is to use a visibility test technique commonly used in computational geometry to solve shortest path extraction problems [26].

3.2.1. Visibility cone based approach

First, we define the predicate TurnPositive(a, b, c) which is true if the points $\{a, b, c\}$ in the plane are sorted counterclockwise. Note that such a predicate can be computed according to the sign of the determinant $det(\vec{ab}, \vec{bc})$.

Let $S = \{P_i\}_{i=0..n}$ be a k-arc, we first fix the first extremity p_0 of the first segment such that $p_0 \in P_0$. Given the pixel P_1 k-adjacent to P_0, we denote e_0 the Euclidean segment shared by the two pixels P_0 and P_1. We consider the first *cone* $C_0(p_0, s, t)$ with center p_0 and defined by the two points s and t such that $\{p_0, t, s\}$ is sorted counterclockwise (i.e. TurnPositive(p_0, t, s) is true) and such that s and t coincide with the extremities of e_0 (see Fig. 5(*left*)). C_0 is a visibility cone since for

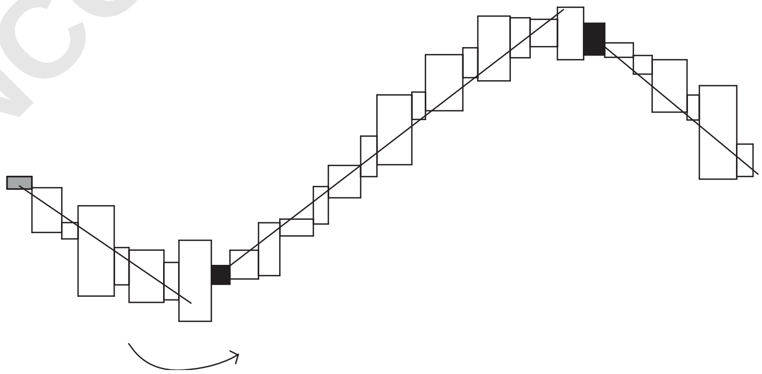

Fig. 4. Illustration of the segmentation algorithm on a general irregular curve. The Euclidean straight lines are manually extracted from the preimages associated to each IDSL segment [15].

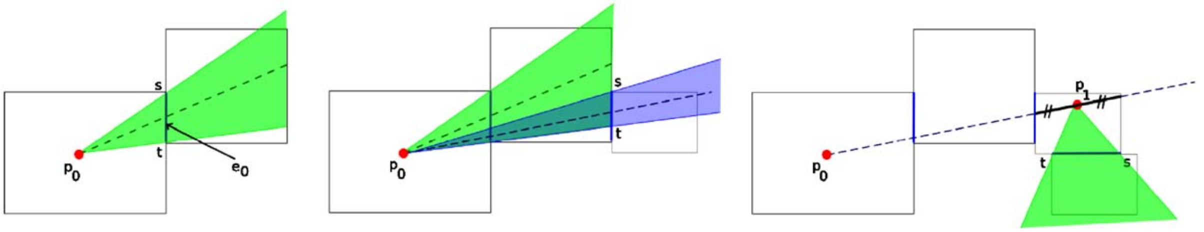

Fig. 5. Illustration of the visibility cone based algorithm: (*from left to right*) the first cone C_0, the update of the cone considering the pixel P_2 and an example when the visibility fails.

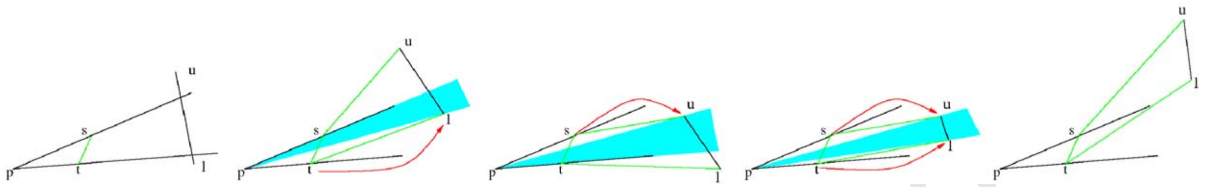

Fig. 6. Illustration of the different cases when we update a cone: (*from left to right*) the cone is not modified, only the point t is moved, only the point s is moved, both s and t are moved, and finally, the visibility fails.

each point p in the intersection between C_0 and the pixels of S, the segment $[p_0p]$ lies exactly in S. In other words, the supercover digitization of $[p_0p]$ is a subset of S.

According to the previous definitions, the cone C_0 describes a subset of the preimages $\mathscr{P}^+(\{P_0, P_1\})$ and $\mathscr{P}^-(\{P_0, P_1\})$ in the parameter space. Indeed, each straight line (p_0p) crosses the pixels P_0 and P_1. More precisely, the set of straight lines contained in the cone C_0 is the segment in the (α, β)-parameter space which corresponds to the intersection between the preimages $\mathscr{P}^+$ and $\mathscr{P}^-$ and the straight line defined by the point p_0. Hence, as proposed in [11], we could have performed all computations in the parameter space (α, β) but the analysis using visibility cones leads to a more efficient algorithm.

The algorithm can be sketched as follows: for each pixel P_i, we consider the shared segment e_i between P_{i-1} and P_i. Then, we have a simple procedure to update the current cone $C_j(p_j, s, t)$ according to $e_i(u, l)$ (such that TurnPositive(p_0, l, u) is true). The different cases are presented in Fig. 6. Note that using the predicate TurnPositive, Algorithm 1 is valid whatever the orientation of the curve and the segment $[ul]$ is not necessarily vertical nor horizontal.

From the different cases presented in Fig. 6, we can design a simple algorithm (Algorithm 1) with three possible outputs: the visibility fails, the cone is updated or the cone remains unchanged.

When the update procedure fails, it means that there is no euclidean straight line going through p_j and crossing the pixel P_i. In that case, we need to start a new recognition process. Hence, we set up a new cone $C_{j+1}(p_{j+1}, s, t)$ where s and t are given by the edge e_i. To compute the new center of the cone p_{j+1} we use a similar strategy as in [11]: we consider the bisector of the cone

Algorithm 1. Visibility cone update procedure

Let $C_j(p_j, s, t)$ be the current cone and $e_i(u, l)$ the shared segment between P_i and P_{i-1}
if (**not** TurnPositive(p, t, u)) **or** TurnPositive(p, s, l) **then**
  return $\emptyset$ {the visibility test fails}
else
  if TurnPositive(p, t, l) **then**
    $t \leftarrow l$
    return $C_j(p_j, s, t)$
  end if
  if not TurnPositive(p, s, u)
    $s \leftarrow u$
    return $C_j(p_j, s, t)$
  end if
end if

(dashed straight lines in Fig. 5) and we define p_{j+1} as the midpoint of the intersection between the bisector and the pixel P_{i-1} (this intersection is not empty since P_{i-1} has already been considered). The idea of this strategy is to obtain a polyline as centered as possible in the discrete curve. By definition of Algorithm 1, the segment $[p_j, p_{j+1}]$ lies inside the irregular discrete curve. Hence, if we repeat the above process for each pixel of the k-arc, the final polyline is an invertible reconstruction of the arc (see Figs. 5(*right*) and 7).

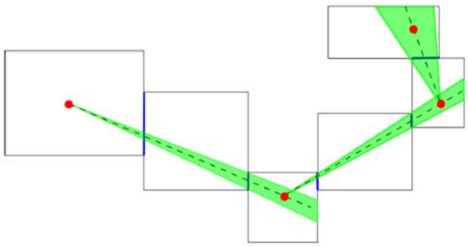

Fig. 7. Illustration of the reconstruction algorithm: (*left*) the sequence of cones during the visibility test and (*right*), the reconstructed polygonal curve.

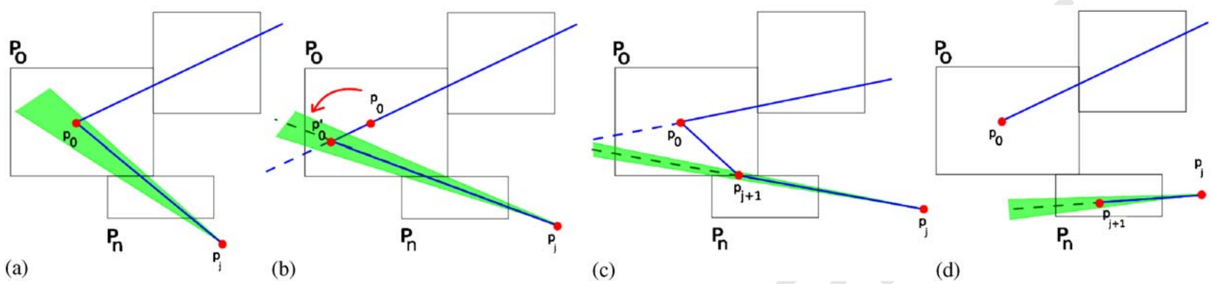

Fig. 8. Different cases to end the reconstruction of a k-curve: (a) and (b) we can close the curve using $[p_0 p_j]$ or $[p'_0, p_j]$, (c) a new vertex p_{j+1} must be inserted, and (d) we have to test cases (a), (b) or (c) using the cone C_{j+1} centered in p_{j+1}.

3.2.2. Overall algorithm

Algorithm 2. Invertible reconstruction of a k-arc

Let $S = \{P_i\}_{i=0..n}$ be a k-arc and p_0 the first point in P_0
Set j = 0
Initialization of the cone $C_j(p_j, s, t)$ using P_0 and P_1 such that TurnPositive(p_j, t, s) is true
p_j is the first vertex of the final polyline
for i from 2 to n **do**
  Compute the shared segment e_i between P_i and P_{i-1}
  $C' \leftarrow$ Update the visibility cone using Algorithm 1
  if $C' = \emptyset$ **then**
    Compute the point p_{j+1} using the bisector of C_j and the pixel P_{i-1}
    Initialization of a new cone C_{j+1} with p_{j+1} and e_i
    Mark p_j as a vertex of the final polyline
  else
    $C_j \leftarrow C'$
  end if
end for

Algorithm 2 presents the complete incremental reconstruction algorithm based on the visibility cone update procedure. Since Algorithm 1 updates the cone is $O(1)$, the overall computational cost of Algorithm 2 is $O(n)$ if n is the number of irregular pixels. Compared to the segmentation algorithm based on the complete preimages, the IDSL segments may be shorter since subsets of the preimages are considered. However, this restriction allows us to construct an invertible polyline.

3.2.3. Invertible reconstruction of k-curves

If we consider an irregular k-curve $S = \{P_i\}_{i=0..n}$, the reconstructed polyline must be closed and thus defines a simple polygon. Hence, we can use Algorithm 2 for the pixels P_0 to P_n and add a specific analysis to handle the adjacency between P_n and P_0 that creates as few as possible new vertices. Let $C_j(p_j, s, t)$ be the last visibility cone such that the intersection between P_n and this cone is not empty. Several cases occur (see Fig. 8): for example, if $p_0 \in C_j$, we close the polyline using the segment $[p_j p_0]$. Otherwise, we may move p_0 along $(p_0 p_1)$ if there exists an intersection between C_j and the straight line $(p_0 p_1)$ that lies inside P_0 (see Fig. 8(b)). In that case, we still close the curve using $[p_j p'_0]$ and the global reversibility of the polygonal curve can be easily proved. Other cases can be derived (for example using the visibility from p_0 to P_n) but additional vertices may be inserted to the polygonal curve (see Fig. 8).

4. Experiments

We have constructed a C++ library to handle elementary irregular objects (irregular pixels, k-arcs and k-curves) (Fig. 9). Using this library, we have

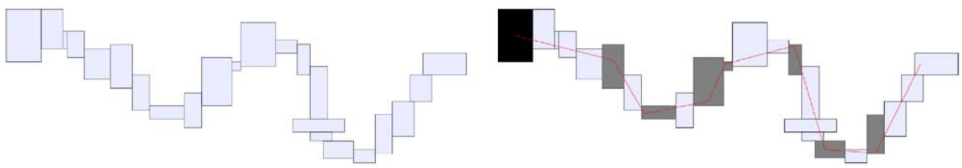

Fig. 9. Result of Algorithm 2 on an irregular *ve*-arc: the input *ve*-curve and the invertible reconstruction using Algorithm 2.

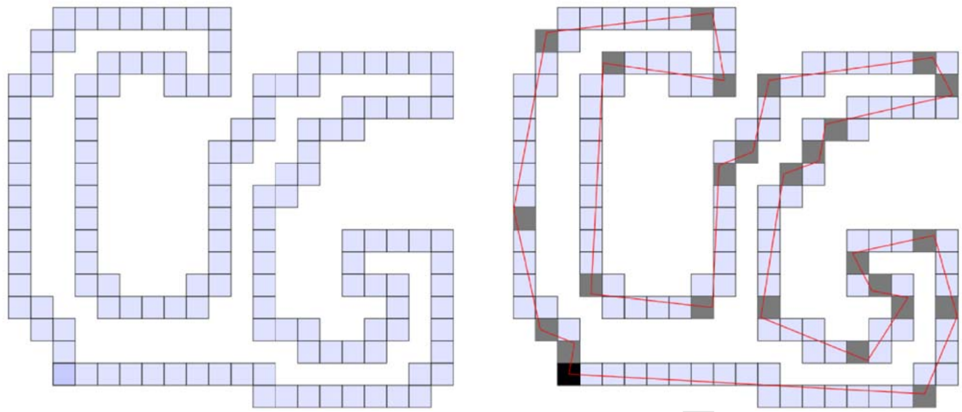

Fig. 10. Result of Algorithm 1 on a classical 4-connected curve: the input 4-connected curve and the invertible reconstruction using Algorithm 2.

implemented the reconstruction algorithm described in the previous section (the code is available on the following web page: <http://liris.cnrs.fr/~dcoeurjo/Code/Reconstruction>). Fig. 10 presents the result of Algorithm 2 on an irregular *k*-arc. Since the classical digital grid is a specific irregular isothetic grid, Algorithm 2 can also be used to reconstruct a polygonal curve from a classical 4-connected curve (see Fig. 10). In this case, results are similar to [11].

5. Conclusion

In this article, we have presented a global digitization framework on irregular isothetic grids: the supercover model. Based on this digitization scheme, we have defined the digital straight lines and briefly presented algorithmic solutions to solve the recognition and segmentation problem. We have also presented an $O(n)$ on-line algorithm to reconstruct a polygonal curve from a discrete irregular arc or curve. Since the classical regular digital grid can be seen as a particular irregular grid, all the presented framework is consistent with classical definitions and algorithms. To achieve the linear in time computational cost, we have only considered specific subsets of the preimages defined by the visibility cones. Thus, the reconstruction may not be optimal in the number of segments. Additional processes similar to [12] in the classical discrete case could be investigated.

Since adaptive grids or QuadTree based decompositions are specific irregular isothetic models, an important future work is to use the proposed framework to provide geometric tools to characterize object boundaries in such grids. Furthermore, topological definitions and data structure to handle irregular objects is an important on going research topic.

References

[1] Bresenham J. An incremental algorithm for digital plotting. In: Proceedings of ACM national conference; 1963.

[2] Dorst L, Smeulders AWM. Length estimators for digitized contours. Computer Vision, Graphics, and Image Processing 1987;40(3):311–33.

[3] Dorst L, Smeulders AWM. Discrete straight line segments: parameters, primitives and properties. In: Vision geometry, series contemporary mathematics. Providence, RI: American Mathematical Society.

[4] Coeurjolly D, Klette R. A comparative evaluation of length estimators of digital curves. IEEE Transactions on Pattern Analysis and Machine Intelligence 2004;26(2):252–8.

[5] Rosenfeld A, Klette R. Digital straightness. In: International workshop on combinatorial image analysis; 2001.

[6] Hübler A, Creutzburg E, Wedler V. Decomposition of digital arcs and contours into a minimal number of digital straight line segments. In: Proceedings of the international conference on pattern recognition; 1982. p. 1218.

[7] Lindenbaum M, Bruckstein AM. On recursive, $o(n)$ partitioning of a digitized curve into digital straight segments. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15(9):949–53.

[8] Debled I, Reveillès JP. A linear algorithm for segmentation of digital curves. In: Third international workshop on parallel image analysis; June 1994.

[9] Brimkov VE, Dantchev SS. Digital hyperplane recognition in arbitrary fixed dimension. Technical Report, CITR-TR-154, Center for Image Technology and Robotics, University of Auckland, New Zealand; 2004.

[10] Breton R, Sivignon I, Dupont F, Andrès E. Towards an invertible Euclidean reconstruction of a discrete object. In: DGCI: international workshop on discrete geometry for computer imagery; 2003.

[11] Sivignon I, Breton R, Dupont F, Andrès E. Discrete analytical curve reconstruction without patches. Image and Vision Computing 2005;23(2):191–202.

[12] Dexet M. Design of a topology based geometrical discrete modeler and reconstruction methods in 2D and 3D. PhD thesis, Laboratoire SIC, Université de Poitiers; December 2005 [in French].

[13] Sintorn IM, Borgefors G. Weighted distance transforms for images using elongated voxel grids. In: 10th international conference on discrete geometry for computer imagery. Lecture notes in computer science, vol. 2301. Berlin: Springer; 2002. p. 244–54.

[14] Finkel RA, Bentley JL. Quad trees: a dsta structure for retrieval on composite key. Acta Informatica 1974;4(1):1–9.

[15] Coeurjolly D. Supercover model and digital straight line recognition on irregular isothetic grids. In: 12th international conference on discrete geometry for computer imagery; 2005.

[16] Rosenfeld A. Connectivity in digital pictures. Journal of the ACM 1970;17(1):146–60.

[17] Rosenfeld A. Digital straight lines segments. IEEE Transactions on Computers 1974; 1264–369.

[18] Cohen-Or D, Kaufman A. Fundamentals of surface voxelization. Graphical Models and Image Processing: GMIP 1995;57(6):453–61.

[19] Andrès E, Nehlig P, Françon J. Tunnel-free supercover 3D polygons and polyhedra. Computer Graphics Forum 1997;16(3):C3–C13.

[20] Andrès E. Modélisation analytique discrète d'objets géométriques. Master's thesis, Laboratoire IRCOM-SIC, Université de Poitiers; 2000.

[21] Dorst L, Smeulders AWM. Discrete representation of straight lines. IEEE Transactions on Pattern Analysis and Machine Intelligence 1984;6:450–63.

[22] McIlroy MD. A note on discrete representation of lines. AT&T Technical Journal 1985;64(2):481–90.

[23] Meggido N. Linear programming in linear time when the dimension is fixed. Journal of the ACM 1984;31(1):114–27.

[24] Buzer L. An incremental linear algorithm for digital line and plane recognition using a linear incremental feasibility problem. In: 10th international conference on discrete geometry for computer imagery. Lecture notes in computer science, vol. 2301. Berlin: Springer; 2002. p. 72–81.

[25] Preparata FP, Shamos MI. Computational geometry: an introduction. Berlin: Springer; 1985.

[26] Coeurjolly D. Visibility in discrete geometry: an application to discrete geodesic paths. In: Discrete geometry for computer imagery; 2002, p. 326–27.