Topological and Geometrical Reconstruction of Complex Objects on Irregular Isothetic Grids

Antoine Vacavant[1], David Coeurjolly[2], and Laure Tougne[1]

[1] LIRIS - UMR 5205, Université Lumière Lyon 2
5, avenue Pierre Mendès-France
69676 Bron cedex, France
[2] LIRIS - UMR 5205, Université Claude Bernard Lyon 1
43, boulevard du 11 novembre 1918
69622 Villeurbanne cedex, France
{antoine.vacavant,david.coeurjolly,laure.tougne}@liris.cnrs.fr

Abstract. In this paper, we address the problem of vectorization of binary images on irregular isothetic grids. The representation of graphical elements by lines is common in document analysis, where images are digitized on (sometimes very-large scale) regular grids. Regardless of final application, we propose to first describe the topology of an irregular two-dimensional object with its associated Reeb graph, and we recode it with simple irregular discrete arcs. The second phase of our algorithm consists of a polygonal reconstruction of this object, with discrete lines through the elementary arcs computed in the previous stage. We also illustrate the robustness of our method, and discuss applications and improvements.

1 Introduction

The character and symbol representation, description and classification are necessary tasks in many current applications, and concern both research and industrial challenges. Those tasks are applied on images generally designed within a regular grid, *i.e.* all the pixels have the same size, and their position can be easily indexed. However, it is now common to successively divide an image into subimages, as in *quadtree* decomposition [21, 22], to represent a part of an image in a more compact and adapted structure. These techniques describe interesting parts of an image, from different points of view, through a set of irregular pixels. In this paper, we introduce the concept of shape representation within an *irregular isothetic grid* ($\mathbb{I}$-grid for short) [3]. The pixels are defined by variable sizes and positions, and may be determined by subdivision rules. We propose to represent the topology of the elements contained in the irregular two-dimensional (2-D) image by constructing their associated Reeb graph [20], then we represent them by a simple polygonal structure that respects the extended supercover digitization model defined in [3]. This structure also preserves the topology that we reveal in the previous stage. We clearly address the problem of *vectorization* (or *raster-to-vector*) on irregular isothetic grids, and not only in the scope of document analysis. In our framework, we are interested in binary images containing irregular objects, *i.e.* $k-$objects in respect to the definition given in [3], where k represents the considered relation of adjacency (see Section 2 for further details). Those complex objects may contain holes, and could represent characters, symbols, lines, etc. An application of such binary image processing is clearly document and line drawings analysis, but we can also consider a discrete subdivision of a part of $\mathbb{R}^2$ representing the solutions of a given function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. The algorithms designed in interval arithmetic are interesting approaches to address those problems [10, 16, 24].

The techniques of vectorization developed until now on the discrete regular domain can be divided into several classes, up to the final application of the method [6, 15, 19, 29]. We will only focus on a few kinds of raster-to-vector methodologies, largely developped for document analysis applications. To our knowledge, there exists no generic extension of those approaches on irregular isothetic grids. The *run length encoding* (RLE) *based methods* first build a decomposition into elongated cells along an axis of the image where we can build a line adjacency graph (LAG) [2, 11]. Those methods aim to describe the topology of the encountered objects in the image, but the geometrical structure deduced from it has to be improved by many post-treatment processings. The *skeletonization* and *thinning methods* are surely the most widely employed methods in vectorization. We can notice that tools designed in mathematical morphology [25] are a frequent choice to prepare the images before processing the skeletonization. A survey of vectorization methods based on skeleton can be found in [18], and another one about such techniques not using it in [28]. The aim is to compute a medial axis of the object that minimally represents its shape [17]. However, those techniques modify the original geometry of the object to obtain a minimal representation of it. Besides, they need filtering or smoothing pre-treatment processings to reduce the noise that could pertubate the final medial axis. The $k-$object can contain holes, and so may be composed by *thick arcs*. In the work of Debled et al. [7–9], the definitions of *discrete lines* and *blurred segments* join the concept of thick regular arcs. But, beyond this geometrical representation of arcs, the global structure is not aborded, and thus there are no description of the topology of the recognized objects.

In this article, we first introduce the concepts of $k-$arcs and $k-$objects by recalling some definitions, then we present the extended supercover model on an $\mathbb{I}-$grid. We also recall the invertible reconstruction of $k-$arcs described in [4]. In the third part, we give details about the two main phases of our system: the description of the topology of a complex object based on the Reeb graph [20], and its polygonal reconstruction. Then, we present some experiments and revealing results to illustrate the two phases of our algorithm. We also proove the robustness of the polygonal reconstruction by a test on a large image of technical drawing. We finally discuss the applications of our contribution, and the improvement on its global performance.

2 Preliminaries

We first define an irregular isothetic grid, denoted $\mathbb{I}$, as a tiling of the plane with isothetic rectangles. We shortly recall that each rectangle P (also called *cell*) of $\mathbb{I}$ is defined by its center $(x_P, y_P) \in \mathbb{R}^2$ and a size $(l_P^x, l_P^y) \in \mathbb{R}^2$. The position and the size of P may be controlled by different level of constraints; *e.g.* in the case of quadtree decomposition [21, 22], for a cell of level k, $(x_P, y_P) = (\frac{m}{2^k}, \frac{n}{2^k})$ and $l_P^x = l_P^y = \frac{1}{2^{k-1}}$ for some $m, n \in \mathbb{Z}$ [3, 4].

In our framework, adjacency relation is an important feature that we depict through the following definitions.

Definition 1 ($ve-$adjacency and $e-$adjacency). *Let P and Q be two cells. P and Q are $ve-$adjacent (vertex and edge adjacent) if :*

$$or \begin{cases} |x_P - x_Q| = \frac{l_P^x + l_Q^x}{2} \text{ and } |y_P - y_Q| \leq \frac{l_P^y + l_Q^y}{2} \\ |y_P - y_Q| = \frac{l_P^y + l_Q^y}{2} \text{ and } |x_P - x_Q| \leq \frac{l_P^x + l_Q^x}{2} \end{cases}$$

P and Q are $e-$adjacent (edge adjacent) if we consider an exclusive "or" and strict inequalities in the above $ve-$adjacency definition. k may be interpreted as e or ve in the following definitions.

Definition 2 ($k-$arc). *Let $\mathcal{E}$ be a set of cells, $\mathcal{E}$ is a $k-$arc if and only if for each element of $\mathcal{E} = \{P_i, i \in \{1, ..., n\}\})$, P_i has exactly two $k-$adjacent cells, except P_1 and P_n which are called extremities of the $k-$arc.*

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

We now consider the extension of the supercover model from [5] on irregular isothetic grids [3] to digitize Euclidean objects on $\mathbb{I}$.

Definition 4 (Supercover on irregular isothetic grids). *Let F be an Euclidean object in $\mathbb{R}^2$. The supercover $\mathbb{S}(F)$ is defined on an irregular isothetic grid $\mathbb{I}$ by:*

$$\begin{aligned} \mathbb{S}(F) &= \{P \in \mathbb{I} \mid \mathbb{B}^\infty(P) \cap F \neq \emptyset\} \\ &= \{P \in \mathbb{I} \mid \exists (x, y) \in F, |x_P - x| \leq \frac{l_P^x}{2} \text{ and } |y_P - y| \leq \frac{l_P^y}{2}\} \end{aligned}$$

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

This model has several interesting properties, *e.g.* for F, G two Euclidean objects in $\mathbb{R}^2$, we have $\mathbb{S}(F \cup G) = \mathbb{S}(F) \cup \mathbb{S}(G)$ or $\mathbb{S}(F \cap G) \subseteq \mathbb{S}(F) \cap \mathbb{S}(G)$ (see proposition 2 in [3] for more details).

We now present the $k-$arc reconstruction algorithm we use in our complex object geometrical representation phase (Section 3.2). Moreover, this approach respects the supercover model we have just presented. The algorithm proposed in [4] to decompose a curve into segments is first based on the following definition of an irregular digital line.

Definition 5 (Irregular isothetic digital straight line). *Let S be a set of cells 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:*

$$S \subseteq \mathbb{S}(l)$$

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

The algorithm inspired from [23] principally uses the construction and update procedures of a *visibility cone*, and can be sketched as follows. We first fix the extremity p_0 of the first segment such that $p_0 \in P_0$. We note e_0 the Euclidean segment shared by P_0 and P_1, and we consider the first cone $C_0(p_0, s, t)$ such that s and t coincide with the extremities of e_0 and $\{p_0, s, t\}$ is sorted counterclockwise. Then, for each cell P_i, we consider the shared segment e_i between P_{i-1} and P_i, and the current cone $C_j(p_j, s, t)$ is updated. When the update procedure fails, a new cone $C_{j+1}(p_{j+1}, s, t)$ is set up, and we add the point p_{j+1} to the reconstruction: to compute the new cone, authors of [4] consider the bisector of the cone and define p_{j+1} as the midpoint of the intersection between the bisector and the pixel P_{i-1}. The Figure 1 illustrates the progressive construction of cones in a $k-$arc, and the resulting segmentation into lines.

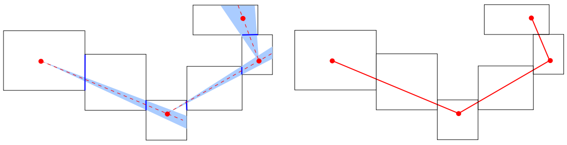

Fig. 1. An example of the progressive construction of cones in a $k-$arc *(left)*, and the reconstruction into segments we obtain *(right)*

3 Complex Objects Definition and Representation on Irregular Isothetic Grids

In this section, we present the two main phases of our system for object representation on irregular isothetic grids.

3.1 Representation of a Complex Topology

To represent the shape of a $k-$object $\mathcal{E}$, we have chosen an incremental directional approach to build its associated Reeb graph G, as in continuous space (see Figure 2). It is an interesting structure introduced by G. Reeb [20] based on the Morse theory [12, 13]. This graph is also used in many applications for surface and curve description [14, 26, 30]. The Reeb graph G is associated to a *height function* f defined on $\mathcal{E}$, and nodes of G represent the critical points of f. Moreover, to have a minimal representation of the topological information of $\mathcal{E}$, each edge of the Reeb graph corresponds to a $k-$arc. Those $k-$arcs will be segmented in the stage of polygonal description of $\mathcal{E}$ (Section 3.2).

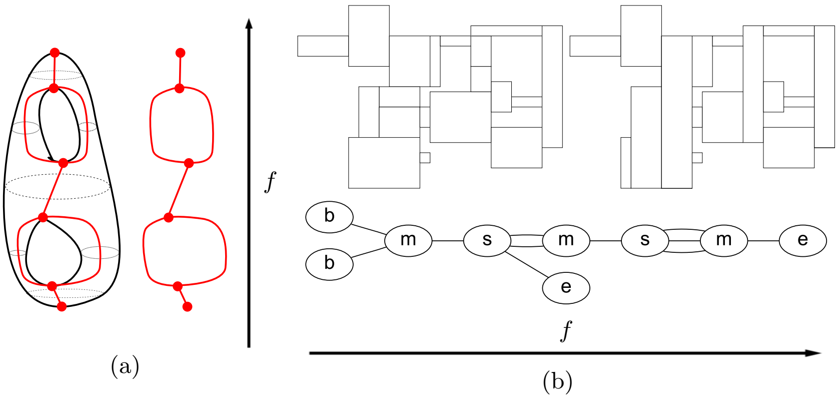

Fig. 2. *(a)*: an example of the Reeb graph G of a continuous object $\mathcal{E}$. The nodes of G represent the critical points of f (maxima, minima, inflection points), and an edge is a connected component of $\mathcal{E}$ between two critical points. *(b)*: an example of an irregular object $\mathcal{E}$ *(left)*, the final recoded structure with $k-$arcs *(right)* and the Reeb graph associated to the height function f defined on $\mathcal{E}$ *(bottom)*. The notations b, e, m and s are given at the end of this section

We denote the left, right, top and bottom borders of a cell P respectively P^L, P^R, P^T and P^B. We have, for example, the abscissa of P^L equal to $x_P - (l_P^X/2)$ (that we denote $P^L = x_P - (l_P^X/2)$). We also abusively say that a $k-$arc A and a cell P are $k-$adjacent if there exists a cell Q in A such that P and Q are $k-$adjacent. Let $\mathcal{E} = \{P_i\}_{i=1,...,n}$ be a given 2-D set of cells. We first choose a direction to treat the cells of $\mathcal{E}$. Without loss of generality, we can suppose that we choose the left-to-right orientation above X axis, *i.e.* the height function f is defined along X axis. At time $t = 0$, we merge together all the $k-$adjacent cells P of $\mathcal{E}$ with the smallest left border $x_{t=0} = x_0$, *e.g.* $P^L = x_0 = 0$. This merging task is processed by the update procedure described below. Those m collections of cells define the *begin cells* of the initial recognized $k-$arcs $A_1, A_2, ..., A_m$.

Update Procedure. Let A be a $k-$arc, and P_1 and P_2 two adjacent cells of $\mathcal{E}$ such that $P_1 \in A$, $P_1^L < P_2^L$, and P_2 should be added to A. If $P_2^L = P_1^R$, we just add P_2 to A, else the procedure *updates* the $k-$arc A with P_2, and may recode A. For that, we first build the *greatest common rectangle* F_2 of P_1 and P_2.

Definition 6 (Greatest common rectangle). *Let P_1 and P_2 be two adjacent rectangles. F_2 is the greatest common rectangle (or GCR) of P_1 and P_2 iff*

i) $F_2 \subseteq P_1 \cup P_2$,
ii) $F_2 \cap P_1 \neq \emptyset$,
iii) $F_2 \cap P_2 \neq \emptyset$,
iv) there is no rectangle greater than F_2 by inclusion respecting i), ii) and iii).

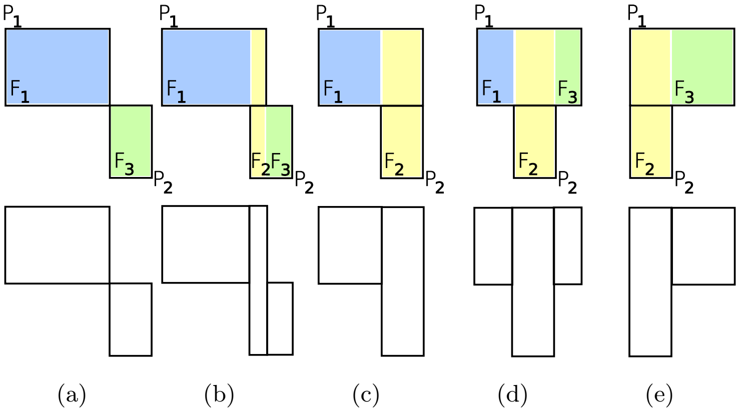

Fig. 3. Description of rectangles F_1, F_2 and F_3 in the update procedure *(top)*, and the associated cells as result *(bottom)*. When $P_1^R < P_2^R$ (a and b), $P_1 - F_2 = F_1$ and $P_2 - F_2 = F_3$, else $P_1 - F_2 = \{F_1, F_3\}$ (d and e). If $P_1^R = P_2^L$, $F_2 = \emptyset$, when $P_1^R = P_2^R$, $F_3 = \emptyset$ and finally $F_1 = \emptyset$ in the case $P_1^L = P_2^L$

Then, we consider the rectangles $P_1 - F_2$ and $P_2 - F_2$. If $P_1^R < P_2^R$, we denote $P_1 - F_2 = F_1$ and $P_2 - F_2 = F_3$, else we prefer $P_1 - F_2 = \{F_1, F_3\}$. We can notice that those rectangles may be empty, *e.g.* $F_3 = \emptyset$ if $P_1^R = P_2^R$, since in that case $F_3^L = F_3^R$. Figure 3 presents the five general configurations of update procedure (there are also five other configurations, obtained by symetry when $P_2^T > P_1^T$), and the $k-$arc recoding that we have to consider. Besides, we propose to reduce the number of cells in A by joining the two rectangles F_1 and F_3 if $F_1^T = F_3^T$, $F_1^B = F_3^B$ and $F_2 = \emptyset$. This junction is processed by replacing F_1 and F_3 by the rectangle $F_1 \cup F_3$. Finally, the procedure ends by removing P_1 from A, and by adding the cells corresponding to the rectangles F_1 and F_2 to A. F_3 is also pushed in $\mathcal{E}$, and will be treated later; more exactely at time t such that $x_t = F_3^L$.

At time $t+1$, our algorithm consists first in merging the adjacent cells with the same left border x_{t+1} in k cells $C_1, C_2, ..., C_k$ (see update procedure for details). Those candidate cells may be added to one or more $k-$arcs among A_i, $i \in \{1, ..., m\}$ if they are adjacent to A_i. It is clear that only a cell Q built at time t and having its right border Q^R equal to x_{t+1} may be adjacent with a cell C_j, $j \in \{1, ..., k\}$. A cell C_j can be treated by several manners:

- C_j is not adjacent with any $k-$arc A_i. We initialize a new $k-$arc A_{m+1} with the cell C_j. C_j represents the *begin cell* of A_{m+1}.
- If C_j is adjacent with one $k-$arc A_i, then we just update A_i with C_j.
- When C_j is $k-$adjacent with p $k-$arcs $A_i, A_{i+1}, ..., Ai + p$, it is a *merge phase*. First, we update each $k-$arc with C_j. The cell C_j is marked as a *merge cell* and indicates that each $k-$arc $A_i, ..., A_{i+p}$ has a $k-$arc $A_{m+1} = \{C_j\}$ linked as a *next arc*.
- The case where p cells $C_j, C_{j+1}, ..., C_{j+p}$ are $k-$adjacent with an $k-$arc A_i is called a *split phase*. We first update A_i with C_j by the update procedure. Then we denote Q the cell in A_i such that $Q^R = x_{t+1}$. We also define p new next $k-$arcs $A_{m+1}, ..., A_{m+p}$ of A_i such that $A_{m+1} = \{Q, C_j\}, ..., A_{m+p} = \{Q, C_{j+p}\}$. In those p $k-$arcs and in A_i, Q is marked as a *split cell.*

When the algorithm ends, at time t such that x_t is the greatest left border in $\mathcal{E}$, we define the last added cell in every $k-$arc A_i as an *end cell.* In this stage of our algorithm, there may also appear a split phase and a merge phase for a cell C_j. We do not detail this specific case but it can be easily handled.

We depict in Figure 4 the progressive construction of the graph and the recoding of the $k-$object presented in Figure 2 *(b)* in five stages of the algorithm.

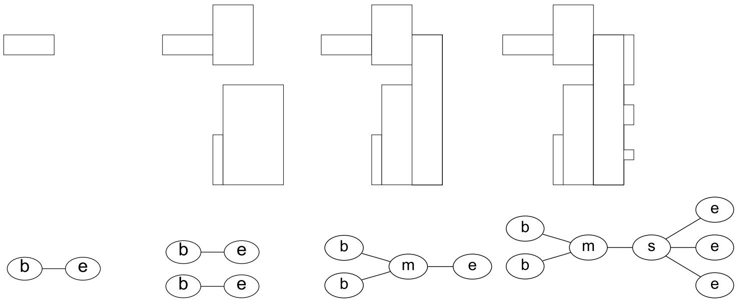

Fig. 4. The recognized $k-$arcs and the associated Reeb graph for some iterations of our algorithm on the object presented in Figure 2 *(b)*. First, we initialize a $k-$arc with the cell with the smallest left border. Then, we progressively update and recode $k-$arcs. The third and fourth images present merge and split phases. We can notice that in one hand the recoding stage is not detailed in this figure, and in the other hand the edges $m - s$ represent a $k-$arc with one cell in this example

Our algorithm finally builds a complete topological representation of $\mathcal{E}$ with the Reeb graph G by recognizing and linking *begin* (b), *merge* (m), *split* (s) and *end* (e) cells in it. There are nine possible configurations of edges in G: $b - s$, $b - m$, $b - e$, $s - s$, $s - m$, $s - e$, $m - s$, $m - m$ and $m - e$. The number of critical points in f can be linked to the Euler number χ of $\mathcal{E}$ [20]. We consider the following equation, where G is denoted as the couple of sets of vertices and edges (V, E):

$$\chi = \sum_{n \in V, (n=b) \vee (n=e)} (deg(n)) - \sum_{n \in V, (n=s) \vee (n=m)} (deg(n) - 2)$$

where $deg(n)$ is the *degree* of the node n in G, so $deg(n) = 1$ if n is a *begin* or *end* node. The Euler number permits to describe the topology of an object by an unique value. For example, for a torus, $\chi = 0$, for a disc, $\chi = 2$, and the object described in Figure 2 *(b)* has a Euler number $\chi = -4$; we can also say that this shape is homeomorpheous to a torus with 3 holes where $\chi = 2 - 2 \times \#(holes) = -4$. Beside the topological invariants obtained by critical points, the structure of the graph clearly depends on the direction we choose for the height function f. A part of the nodes and the edges may change, but the information on the topology of $\mathcal{E}$, *i.e.* internal nodes of G, is not modified. The Euler number is an example of the use of the Reeb graph for shape description. Let us consider now $\mathcal{E}'$ as the object drawn in the fourth image of Figure 4. The three cells added during the last iteration could be noise modifying the contour of $\mathcal{E}'$. The Reeb graph is modified by a split phase, three nodes are created, whereas these cells are maybe noise. Actually, the problem of the perturbation of the contour of $\mathcal{E}'$ could be certainly reduced if the object was first filtered or smoothed. This kind of pre-treatment processings is often adopted, whatever the approach we may choose for shape representation, *e.g.* skeletonization. Finally, with the update procedure, we recode the cells in $\mathcal{E}$ so that a $k-$arc is always represented between two nodes of G. This geometrical rearrangement clearly depends on the direction of f, but does not change neither the topology nor the contour of the recognized $k-$arcs. The topological structure so described is simple, and prepares the next phase of our complex objects reconstruction system.

3.2 Polygonal Reconstruction of Thick Objects

Since the reconstruction into polylines always affects the first point p_0 as the center of the first treated cell, we propose to start the reconstruction of every $k-$arcs computed in the previous stage by the *merge* and *split* nodes detected in the Reeb graph G. This insures that each of those particular nodes of G will be represented by an unique point in the final polygonalization. The segments are recognized from intersections between several parts of the object $\mathcal{E}$ to its extremities, *i.e.* we consider the edges $m - e$, $s - e$, $m - b$ and $s - b$ of G. Moreover, since the recognition algorithm is greedy, the possible error induced by the visibility cone approach is propagated to the extremities of $\mathcal{E}$, instead of those intersections that represent the shape of the object. For the $m - s$, $m - m$, $s - s$ and $s - m$ configurations of edges in G we propose to process a bidirectional reconstruction that begins from each node of the edge, and ends in its center. Thus, the error may be concentrated in the midpoint of those edges. This approach confirms that the nodes m and s of G represent the places of an object where the description of its geometry must be precise. Finally, we choose to treat edges $b-e$ by the same bidirectional reconstruction, that seems to be the more efficient way to insure a robust reconstruction. We do not deal about the problem of linking the two reconstructions on the $k-$arc (reconstruction with *patch*), because an efficient and general joint technique between two discrete lines implies that our algorithm would not be linear anymore [1]. Hence we just add a segment between the two polylines. This phase of our system can not be handled without patch, since we use the internal points of the shape of $\mathcal{E}$ to guide the geometrical reconstruction.

In Figure 5 *b*, we illustrate the behaviour of our algorithm in the case of the object $\mathcal{E}$ presented in the previous section. We also show the interest of our approach for a symetrical complex object.

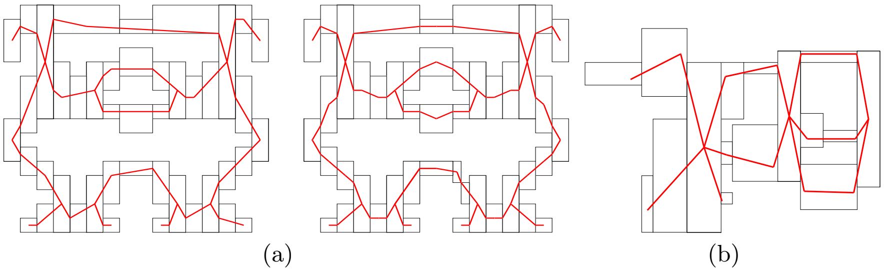

Fig. 5. If we consider the original orientation of the $k-$arcs, the shape of the $k-$object presented in the next section *(a left)* is not well defined since the symetry is not preserved. So, we propose to start the reconstruction by the nodes s and m *(a right)*. This structure respects the supercover model, and the symetrical shape of this object. We also show the result of our algorithm on the $k-$object presented in the previous section *(b)*

Contrary to conventional vectorization methods, we propose a technique that respects the supercover model on an $\mathbb{I}-$grid. We do not address the quality of the global polygonal structure deduced from this second phase of our system. To introduce the concept of quality in the framework of document analysis, we may refer to [27].

4 Experimentation and Results

In Figure 6, we present the polygonal structure obtained on an image first rearranged by a quadtree-based approach. The reconstruction of $k-$arcs stands inside the object, and the *split* and *merge* nodes are represented with one point in the reconstruction. The polygonal representation also permits to measure geometrical features (*e.g.* length) of a complex function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ (Figure 7). f is first discretized by an interval computing algorithm through a set of cells $\mathcal{E}$, then we use our system to minimally describe the curves of $\mathcal{E}$. Finally, to show the robustness of our system, we present in Figure 8 the polygonal and topological reconstructions of a large image of technical drawing.

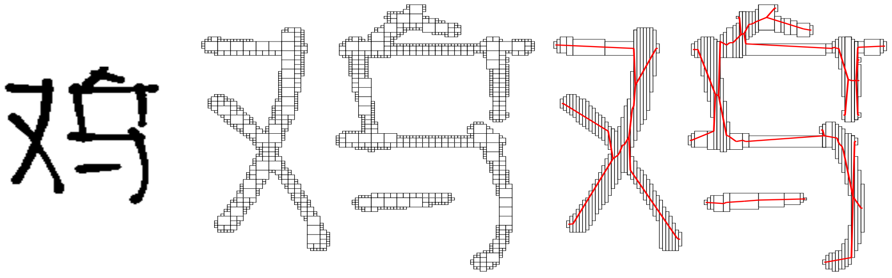

Fig. 6. An image of a chinese character *(left)*, compressed by a quadtree-based approach *(center)*. We show the final $k-$arcs recoding and the polygonalization *(right)*

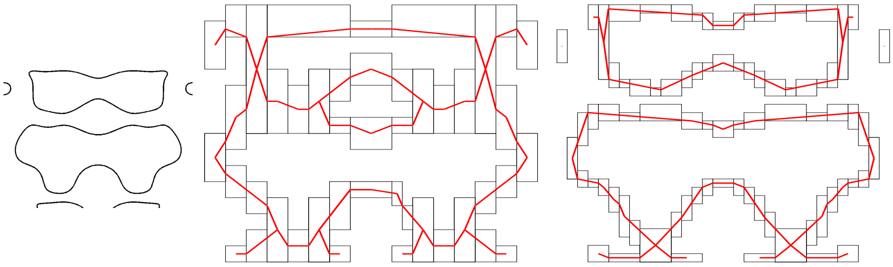

Fig. 7. The function $x^2 + y^2 + \cos(2\pi x) + \sin(2\pi y) + \sin(2\pi x^2)\cos(2\pi y^2) = 1$ on $[-1.1; 1.1] \times [-1.1; 1.1]$ *(left)* discretized by an algorithm described in [24] with two different resolutions, then recoded and polygonalized *(center* and *right)*

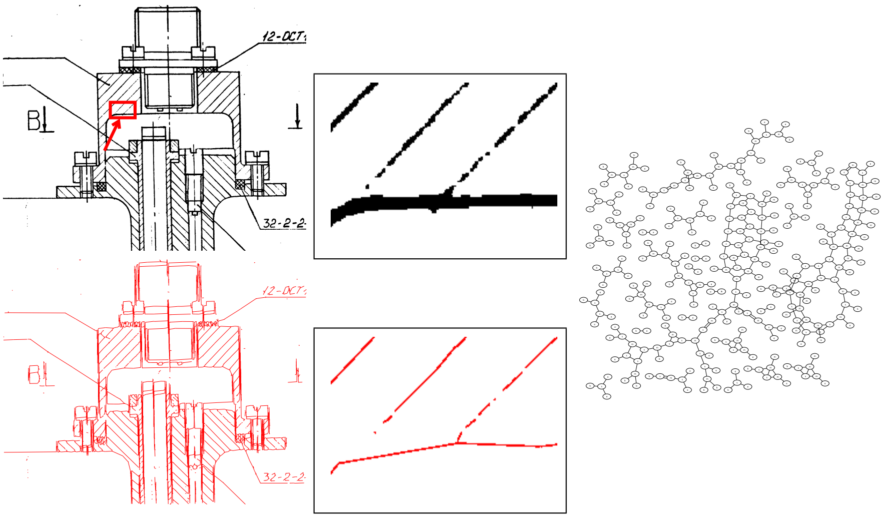

Fig. 8. An image of technical drawing of size 1765 x 1437 pixels we submit to our system, and a zoomed part of it, indicated by the arrow *(up)*. The polygonalization we obtain and the associated zoom are presented *(bottom)*. The complete Reeb graph (about 300 nodes) is also illustrated in a circular format *(right)*

5 Conclusion and Future Work

The representation by lines of an object described on a binary image is a classical problem often considered in the framework of document analysis. We have proposed to enlarge the scope of vectorization methodologies to irregular isothetic representation of binary data. Depending on the final application of our system, we can treat the initial image with pre-treatment processings, reorganize the Reeb graph (edge contraction, etc.), or rearrange the segments finally processed in the second phase. The geometrical reconstruction stands inside the object, *i.e.* it respects the irregular digitization supercover model. Moreover, this reconstruction preserves the topology described by the Reeb graph. Thus, our system is robust, and topologically and geometrically correct. The Reeb graph can be extended to three-dimensional (3-D) object description, with a similar incremental approach. However, visibity cone reconstruction is hardly adaptable to such irregular objects. Our system should be modified to provide a 3-D polygonalization based on the Reeb graph. Such technique would be convenient especially for medical imaging, *e.g.* organ representation in an irregular 3-D CT-scan image.

References

[1] Breton, R: Reconstruction inversible d'objets discrets 2D. PhD thesis, *Université de Poitiers*, Poitiers, France, December 2003.

[2] Burge, M., Kropatsch, W.G.: A minimal line property preserving representation of lines images. In *Computing*, 62:355–368, published by Springer-Verlag, 1999.

[3] Coeurjolly, D.: Supercover model and digital straight line recognition on irregular Isothetic Grids. In *12th International Conference on Discrete Geometry for Computer Imagery*, LCNS 3429, pages 311–322, 2005.

[4] Coeurjolly, D., Zerarga, L.: Supercover model, digital straight line recognition and curve reconstruction on the irregular isothetic grids. In *Computer and Graphics*, 30(1):46–53, 2006.

[5] Cohen-Or, D., Kaufman, A.: Fundamentals of surface voxelization. In *Graphical models and image processing : GIMP*, 57(6):453–461, November 1995.

[6] Cordella, L.P. and Vento, M.: Symbol recognition in documents: a collection of techniques ? In *International Journal on Document Analysis and Recognition*, 3(2):73–88, published by Springer-Verlag, 2000.

[7] Debled, I., Reveillès, J.P.: A linear algorithm for segmentation of digital curves. In *Third International Workshop on Parallel Image Analysis*, June 1994.

[8] Debled, I., Tabbone, S., Wendling, L.: Fast polygonal approximation of digital curves. In *International Conference on Pattern Recognition*, Volume 1 pages 465–468, Cambridge, United Kingdom, August 2004.

[9] Debled, I., Feschet, F and Rouyer-Degli, J.: Optimal blurred segments decomposition in linear time. In *12th International Conference on Discrete Geometry for Computer Imagery*, LCNS 3429, pages 311–322, 2005.

[10] de Figueiredo, L.H., Van Iwaarden, R., Stolfi, J.: Fast interval branch-and-bound methods for unconstrained global optimization with affine arithmetic. Technical Report IC-9708, *Institute of Computing, Univ. of Campinas*, June 1997.

[11] Elgammal, A., Ismail, M.A.: Graph-based segmentation and feature-extraction framework for arabic text recognition. In *6th International Conference on Document Analysis and Recognition (ICDAR 01)*, Seattle, Washington, USA, September 10–13, 2001.

[12] Gramain, A.: Topologie des surfaces. *Presses Universitaires Françaises*, 1971.

[13] Hart, J.C.: Computational topology for shape modeling. In *Shape Modeling International SMI'99*, pages 36–43, published by IEEE Computer Society, Aizu, Japan, March 1–4 1999.

[14] Hétroy, F.: Méthodes de partitionnement de surfaces. PhD thesis, *Institut National Polytechnique de Grenoble*, Grenoble, France, September 2003.

[15] Hilaire, X., Tombre, K.: Robust and accurate vectorization of line drawings. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[16] Kearfott, B.: Interval computations: introduction, uses, and resources. In *Euromath Bulletin* 2(1):95–112, 1996.

[17] Klette, R., Rosenfeld, A.: Digital geometry. Published by Elsevier, San Fransisco, USA, 2004.

[18] Lam, L., Lee, S.W., Suen C.Y.: Thinning methodologies - a comprehensive survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9): 869–885, 1992.

[19] Mertzios, B.G., Karras, D.A.: On applying fast and efficient methods in pattern recognition. In *Signal Processing for Multimedia*, published by IOS Press, J.S. Byrnes Ed., 1999.

[20] Reeb, G.: Sur les points singuliers d'une forme de Pfaff complément intégrable ou d'une fonction numérique. In *Comptes Rendus de L'Académie ses Séances*, Paris 222, pages 847–849, 1946.

[21] Samet, H.: The quadtree and related hierarchical data structures. In *ACM Computer Survey*, 16(2):187–260, 1984.

[22] Samet, H.: Hierarchical spatial data structures. In *Design and Implementation of Large Spatial Databases, First Symposium SSD'89*, 409:193–212, published by Springer, Santa Barbara, California, July 17–18 1989.

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

[24] Snyder, J.M.: Interval analysis for computer graphics. In *Computer Graphics*, 26(2):121–130, July 1992.

[25] Soille, P.: Morphological image analysis, 2nd ed. Published by Springer, Berlin, Germany, 2003.

[26] Tung T.: Indexation 3D de bases de données d'objets 3D par graphes de Reeb améliorés. PhD thesis, *Telecom Paris, ENST/TIC*, Paris, France, June 2005.

[27] Wenyin, L., Dori, D.: A protocol for performance evaluation of line detection algorithms. In *Machine Vision and Applications*, 9(5/6):240–250, 1997.

[28] Wenyin, L., Dori, D.: A Survey of non-thinning based vectorization methods. In *A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, Advances in Pattern Recognition (Proceedings of Joint IAPR Workshops SSPR'98 and SPR'98)*, LNCS 1451, pages 230–241, Sydney, Australia, August 1998.

[29] Wenyin, L., Dori, D.: From raster to vectors: extracting visual information from line drawings. In *Pattern Analysis and Application*, 2(1):10–21, 1999.

[30] Xiao, Y., Siebert, P., Werghi, N.: A discrete Reeb graph approach for the segmentation of human body scans. In *4th International Conference on 3D Digital Imaging and Modeling*, pages 378–385, Banff, Canada, October 6–10 2003.