Dynamic Reconstruction of Complex Planar Objects on Irregular Isothetic Grids

Antoine Vacavant, David Coeurjolly, Laure Tougne

▶ To cite this version:

Dynamic Reconstruction of Complex Planar 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. The vectorization of discrete regular images has been widely developed in many image processing and synthesis applications, where images are considered as a regular static data. Regardless of final application, we have proposed in [14] a reconstruction algorithm of planar graphical elements on irregular isothetic grids. In this paper, we present a dynamic version of this algorithm to control the reconstruction. Indeed, we handle local refinements to update efficiently our complete shape representation. We also illustrate an application of our contribution for interactive approximation of implicit curves by lines, controlling the topology of the reconstruction.

1   Introduction

The representation by lines of graphical elements is an important task in many image processing and synthesis applications [4, 5, 8, 12, 13]. In general, images are digitized on regular grids: 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 [11], to represent a part of an image in a more compact and adapted manner. An other example: to approximate an implicit curve, the algorithms designed in interval arithmetic build an irregular tiling of the plane [6, 9, 13]. We have presented in [14] a system to represent the elements contained in an *irregular isothetic grid* ($\mathbb{I}$-grid for short) [1]: the pixels are defined by variable sizes and positions, and may be determined by subdivision rules. We describe the topology of the elements in a two-dimensional (2-D) image by their associated Reeb graph [10], then we geometrically represent them by a simple polygonal structure. By using an irregular image representation, we naturally produce less line segments than in the regular discrete applications.

In this article, we propose to efficiently modify this representation to handle the dynamic construction of an $\mathbb{I}-$grid. Moreover, we control the topology of the recognized irregular objects. Indeed, we may build an irregular grid by successively applying *local refinements by inclusion* that respects a grid model. If we consider an irregular grid $\mathbb{I}$ and a cell $P \in \mathbb{I}$, such a refinement performed on

P is a set of subcells $S(P) \subseteq P$ (inclusion rule), obtained by the process S. For example, in the quadtree decomposition, a cell $P \in \mathbb{I}$ is divided into four subcells if the region of the 2-D image bounded by P is not homogeneous. In this case, S represents the operation of division in four subcells, and we consider the 2-D image data to check the homogeneity criterion. To completely define such a grid model, the choice of the refined cell P can be guided by two main approaches: (1) an irregular grid can be successively refined by inclusion to respect a subdivision model, as in the quadtree decomposition where all the heterogeneous cells are subdivided; (2) we can manually choose the cells to refine. This interactive way of construction is interesting for implicit curve approximation by interval arithmetic, where an user may locally refine the approximation. In this kind of implicit curve approximation, the digitization algorithms consist in tiling the plane with isothetic rectangles so that the real function $f : \mathbb{R}^2 \to \mathbb{R}$ belongs to each rectangle. For now, we focus on the interactive refinement of those *data driven grids* [1]. From a coarse description of f, we choose one or more rectangles in $\mathbb{I}$, where a finer description of f is performed. We proove that we can handle local refinements by inclusion in our reconstruction of an irregular object. By an efficient and simple algorithm, we update its topological representation by the Reeb graph, then we deduce a new polygonal approximation of its shape.

We first introduce the concepts of $k-$arcs 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 [2]. In the third part, we give details about our contribution: we shortly describe the topological and geometrical reconstruction of a complex object based on the Reeb graph [10] that we have proposed in [14]. Then, we present our algorithm designed to update this structure considering one or several local refinements. We also present an application of our system to the interactive approximation of an implicit curve by lines. We finally discuss about possible extensions of our method.

2   Preliminaries

We first define an irregular isothetic grid, denoted $\mathbb{I}$, as a tiling of the plane with isothetic rectangles. 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 $(l_P^x, l_P^y) \in \mathbb{R}^2$. 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 :*

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

P and Q are e−adjacent (edge adjacent) if we consider an exclusive "or" and strict inequalities in the above ve−adjacency definition. k will now represent either e or a ve adjacency 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.*

We now consider the extension of the supercover model from [3] on irregular isothetic grids [1] to digitize Euclidean objects on $\mathbb{I}$.

Definition 3 (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 :*

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

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).

We now present the $k-$arc reconstruction algorithm we use in our complex object geometrical representation phase (Section 3.1). Moreover, this approach respects the supercover model we have just presented. The algorithm proposed in [2] to decompose a curve into segments is first based on the following definition of an irregular digital line.

Definition 4 (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 Figure 1 illustrates the progressive construction of cones in a $k-$arc, and the resulting segmentation into lines.
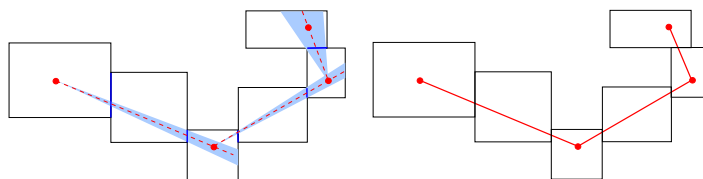

Fig. 1. An example of the progressive construction of cones in a $k-$arc (*left*), and the reconstruction into segments we obtain *(right)*

3 Dynamic Representation of Complex 2-D Objects on Irregular Isothetic Grids

3.1 Topological and Geometrical Reconstruction of a Complex 2-D Object

To represent the shape of a $k-$object $\mathcal{E}$, we have chosen in [14] an incremental directional approach to build its associated Reeb graph G, as in continuous space. It is an interesting structure introduced by G. Reeb [10] based on the Morse theory [7]. The Reeb graph $G = (V, E)$ is associated to a *height function f* defined on $\mathcal{E}$, and nodes of G represent the critical points of f. These critical nodes represent the important elements of the topology of $\mathcal{E}$. More exactly, we denote them *begin* (b), *merge* (m), *split* (s) and *end* (e) nodes (see Algorithm 1). To handle the orientation of f, we 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 along X axis, *i.e.* the height function f is defined along X axis. Moreover, we want to represent an edge between two nodes by a $k-$arc, to have a minimal information in the topological representation of $\mathcal{E}$. To respect this feature, we use a simple procedure to update and recode a $k-$arc with one or more adjacent cells (see [14] for more details). In Algorithm 1, we present the main stages of our reconstruction algorithm of complex irregular objects. As a $k-$arc A is associated to an edge $n_1 - n_2 \in E, n_1, n_2 \in \{b, e, m, s\}$, we note in this algorithm $inf(A) = n_1$ and $sup(A) = n_2$. 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. We finally note $\mathcal{E} = \{P_i\}_{i=0,\dots,n}$ a given 2-D set of cells.

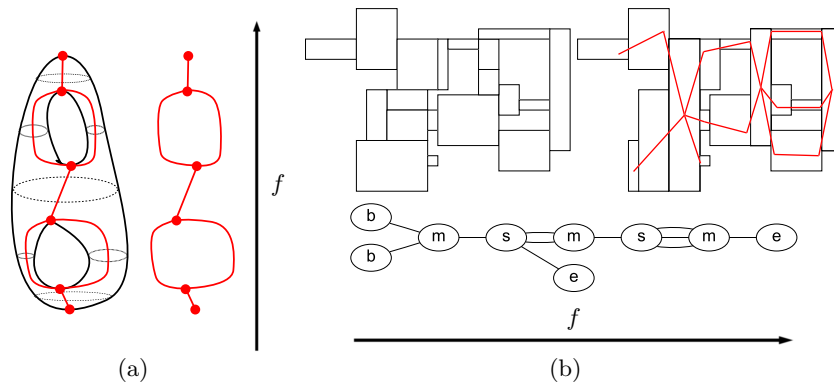(a)   (b)

Fig. 2. *(a)* : an example of the Reeb graph G of a continuous object $\mathcal{E}$. The node 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, the obtained polygonization *(right)* and the Reeb graph associated to the height function f defined on $\mathcal{E}$ *(bottom)*

Indeed, our method first builds a complete topological representation of $\mathcal{E}$ with the Reeb graph G by recognizing and linking critical nodes. At the end of our approach, we use the $k-$arc reconstruction algorithm (see section 2) to build a polygonal description of $\mathcal{E}$. We process the geometrical reconstruction of E from merge and split nodes (internal nodes of our structure) to the others. Thus, all the merge or split nodes in G are represented by a single point. 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. The complexity in the worst case of this algorithm is $\mathcal{O}(m \times n)$: for all P_i, we test all the m $k-$arcs A_j to find which cell P_i is adjacent with A_j. In Figure 2, we present the Reeb graph and the polygonization obtained by our approach on a simple example.

Algorithm 1 Reconstruction of an object $\mathcal{E} \in \mathbb{I}$

Input : $\mathcal{E} = \{P_i\}_{i=1,...,n}$, a set of n 2-D cells in $\mathbb{I}$.
Output : $G = (V, E)$, the Reeb graph associated to $\mathcal{E}$
         L, the reconstruction by lines of $\mathcal{E}$
         $A = \{A_i\}_{i=1,...,m}$, the set of m $k-$arcs recoding $\mathcal{E}$
1: Add an edge $b - e$ in G
2: Create the arc A_1 with $inf(A_1) = b$ and $sup(A_1) = e$ {m = 1}
3: **for all** cells P_i, $2 \leq i \leq n$ **do**
4:    **for all** arcs A_j, $1 \leq j \leq m$ **do**
5:       **if** one cell P_i is adjacent with one arc A_j **then**
6:          Update A_j with P_i
7:       **end if**
8:       **if** p cells $P_i, P_{i+1}, ..., P_{i+p}$ are adjacent with A_j **then**
9:          Update A_j with $P_i, P_{i+1}, ..., P_{i+p}$
10:         Add a split node s in G {$deg(s) = p + 1$}
11:         $sup(A_j) = s$
12:         Link p new arcs $A_{m+1} = P_i, ..., A_{m+p} = P_{i+p}$ with A_j
13:         $inf(A_{m+1}) = ... = inf(A_{m+p}) = s$
14:       **end if**
15:       **if** one cell P_i is adjacent with p $k-$arcs $A_j, A_{j+1}, ..., A_{j+p}$ **then**
16:         Update $A_j, A_{j+1}, ..., A_{j+p}$ with P_i
17:         Add a merge node m in G {$deg(m) = p + 1$}
18:         $sup(A_j) = ... = sup(A_{j+p}) = m$
19:         Link a new arc $A_{m+1} = P_i$ with $A_j, A_{j+1}, ..., A_{j+p}$
20:         $inf(A_{m+1}) = m$
21:       **end if**
22:       **if** a cell P_i is adjacent with any arc A_j **then**
23:         Add an edge $b - e$ in G
24:         Create an arc A_{m+1} with $inf(A_{m+1}) = b$ and $sup(A_{m+1}) = e$
25:       **end if**
26:    **end for**
27: **end for**
28: Perform the polygonization over the set of arcs $\{A_j\}_{j=1,...,m}$ considering G

3.2   Dynamic Reconstruction of Complex 2-D Objects

We now want to update our reconstruction of complex objects, considering one or more local refinements by inclusion. More precisely, we have to insert the reconstruction of the set of cells defined in the local refinement into the global reconstruction. Here, we propose an efficient algorithm to locally change the Reeb graph and the polygonal approximation we compute in Algorithm 1.

Let $S_t(\Omega)$ be a set of cells obtained by a process S_t at time t on the domain $\Omega \subset \mathbb{R}^2$, *e.g.* a quadtree decomposition of level t in the image plane. Let $P \in S_t(\Omega)$ be a cell where we want to have a more precise decomposition, *e.g.* a local quadtree refinement of level $t + 1$. We denote $S_{t+1}(P)$ the set of cells in the domain bounded by P obtained by the process S_{t+1}. G_t, L_t and A_t are the elements computed by Algorithm 1 (see Input of Algorithm 2). For each $k-$arc $A_i \in A_t$, its associated edge in G_t is denoted in a lower case a_i. In this section, we propose a simple algorithm to update G_t, L_t and A_t considering $S_{t+1}(P)$. First, we consider all the coarse $k-$arcs in $S_t(\Omega)$ that contain P. Then, we replace P with the smaller pixels $S_{t+1}(P)$, and perform Algorithm 1 over those pixels to have a local topological and geometrical representation of them. Then, we branch the local recognized arcs represented in $G_{t+1}(P)$, $L_{t+1}(P)$ and $A_{t+1}(P)$ (associated to $S_{t+1}(P)$) in G_t, L_t and A_t respectively.

In Figure 3, we illustrate the behaviour of our algorithm for the two cases $p = 1$, *i.e.* the cell P belongs to one $k-$arc (Algorithm 2, line 4) and $p > 1$, where P belongs to several $k-$arcs (Algorithm 2, line 2).
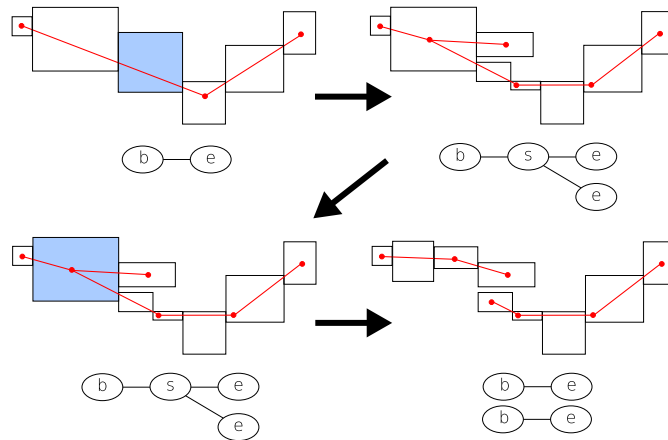

Fig. 3. Examples of results for Algorithm 2. The first reconstruction *(top-left)* contains one arc. Then, we choose the cell in dark and decompose it into two $k-$arcs *(top-right)*. This cell belongs to one arc, thus $p = 1$ and the edge $b - e$ has to be divided into two before adding the node s. The next update *(bottom)* considers three $k-$arcs, since we choose a split cell $(p > 1)$. This cell and its associated node (s) are removed before inserting the new edges

---

Algorithm 2 Update reconstruction of an object $\mathcal{E} \in \mathbb{I}$

Input : $S_t(\Omega)$, decomposition of $\Omega \subset \mathbb{R}^2$ by the process S_t
$\qquad$ $S_{t+1}(P)$, the local decomposition of $P \in S_t(\Omega)$ by S_{t+1}
$\qquad$ G_t, the Reeb graph associated to $S_t(\Omega)$
$\qquad$ L_t, the reconstruction by lines of $S_t(\Omega)$
$\qquad$ A_t, the set of $k-$arcs recoding $S_t(\Omega)$
Output : G_t, L_t and A_t are updated
1: Find the p $k-$arcs $A_k^t, ..., A_{k+p}^t \in A_t$ so that $P \in A_k^t, ..., A_{k+p}^t$
2: **if** $p > 1$ **then** $\{P$ represents a split or merge node in $G_t\}$
3: $\qquad$ Remove the node $n \in G_t$ associated to P
4: **else** $\{P$ is in one arc$\}$
5: $\qquad$ Split a_k^t into two edges a_k^t, a_{k+1}^t
6: **end if**
7: Remove the polylines of L_t associated to the $k-$arcs $A_k^t, ..., A_{k+p}^t$
8: Remove P from $A_k^t, ..., A_{k+p}^t$
9: Use Algorithm 1 to compute $G_{t+1}(P)$, $L_{t+1}(P)$ and $A_{t+1}(P)$ associated to $S_{t+1}(P)$
10: **for all** edges $a_i^{t+1} \in G_{t+1}(P)$ **do**
11: $\qquad$ **for all** edges a_j^t, $j = k, ..., k+p$ **do**
12: $\qquad\qquad$ **if** A_j^t and A_i^{t+1} are $k-$adjacent **then**
13: $\qquad\qquad\qquad$ Branch a_i^{t+1} to a_j^t
14: $\qquad\qquad\qquad$ Merge the $k-$arcs A_j^t and A_i^{t+1}
15: $\qquad\qquad$ **end if**
16: $\qquad$ **end for**
17: **end for**
18: Add all polylines from $L_{t+1}(P)$ to L_t
19: **for all** $k-$arcs A_j^t, $j = k, ..., k+p$ **do**
20: $\qquad$ Compute $k-$arc reconstruction on A_j^t
21: **end for**

To find the p $k-$arcs required by the first operation of Algorithm 2 (line 1), we use a pointer between the cell P and a $k-$arc A_k^t such that $P \in A_k^t$. If P is inside A_k^t, it is clear that $p = 1$, and so only one $k-$arc is considered. When P stands on one of the extremities of A_k^t, we have to consider all the $k-$arcs linked with A_k^t. Thanks to its associated edge a_k^t in G, and the direct access to the edges linked of G with a_k^t, this search operation is computed in constant time $\mathcal{O}(1)$. Then, the next instructions change the Reeb graph G_t in the set of p edges $\{a_j^t\}_{j=k,k+p}$ and impact the p $k-$arcs $\{A_j^t\}_{j=k,k+p}$ found in the first operation (lines 2-17). The topology of the irregular object $\mathcal{E}$ is locally updated. The rest of the graph does not change. In the last phase of the algorithm, the polygonal reconstruction is first computed in the $k-$arcs belonging to the local refinement $A_{t+1}(P)$. Then, this reconstruction may be propagated to the p global $k-$arcs A_t, as in Figure 3 *(top)*. In the worst case, the update of the polygonal reconstruction is locally performed on $A_{t+1}(P)$ and A_t, and does not modify the rest of the reconstruction. Finally, we can notice that our algorithm locally impact the topological and geometrical reconstruction we globally compute for $\mathcal{E}$.

To perform our algorithm on several refined cells, we first update the Reeb graph with the local graphs using the same approach (Algorithm 2, lines 1-17).

Then, we perform only once the final polygonalization over the concerned $k-$arcs (see next section for a complete example).
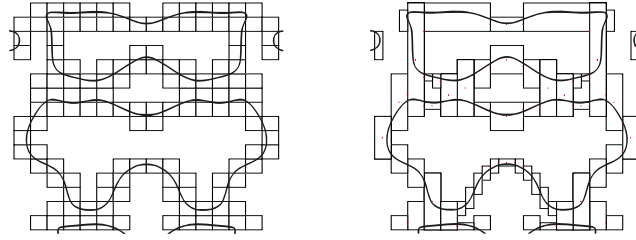
4 Application to Interactive Approximation of Implicit Curves by Lines

In this part, we propose to apply our interactive system to approximate an implicit curve by interval arithmetic analysis. The interval arithmetic is a widely used range based model for numerical computation where each quantity x is represented by an interval $\bar{x}$ of floating point numbers. Arithmetic operations are defined so that each resulting interval $\bar{x}$ is guaranteed to contain the unknown value corresponding to the real quantity x. Briefly, an interval $\bar{x}$ is represented by $[\bar{x}.inf, \bar{x}.sup]$ and we have, for example, $\bar{x}+\bar{y} = [\bar{x}.inf+\bar{y}.inf, \bar{x}.sup+\bar{y}.sup]$ or $\bar{x} - \bar{y} = [\bar{x}.inf - \bar{y}.inf, \bar{x}.sup - \bar{y}.sup]$. Thanks to this interval definition of operations, we can build an approximation $f(\bar{x}, \bar{y})$ of a function $f : \mathbb{R}^2 \to \mathbb{R}$ by tiling the plane with an irregular isothetic grid, where each rectangle is given by $[\bar{x}.inf, \bar{x}.sup] \times [\bar{y}.inf, \bar{y}.sup]$ (Figure 4-*up* for example). Then, the polygonal representation of the computed irregular object permits to have a first geometrical analysis of the unknown underlying Euclidean curve. For example, we can approximate geometrical features (*e.g.* length) of f.

In this section, we first use an interval computing algorithm described in [13] to discretize $f = x^2+y^2+\cos(2\pi x)+\sin(2\pi y)+\sin(2\pi x^2)\cos(2\pi y^2) = 1$ through a set of cells $S_0(\Omega)$, where $\Omega = [-1.1; 1.1] \times [-1.1; 1.1]$ (Figure 4 *(top-left)*). The width δ of the final computed ranges is fixed: $\delta = 0.1$. There exists many rules to build various irregular tilings of the plane. We have chosen this simple criterion to generate a regular grid, but our algorithm handles any irregular isothetic grid. Then, we successively update the elements computed by Algorithm 1 with local refinements by inclusion. We select several cells at each time, and use Algorithm 2 as we have presented it in the previous section. The new cells contained in the refinement are deduced from the same interval computation algorithm, with a smaller range width ($\delta = 0.05$). In the figure 4, we show four times of update. The first update procedure, with the input presented in $t = 0$, does not modify the Reeb graph ($G_0 = G_1$) while the geometrical reconstruction (A_1 and L_1) is finer ($t = 1, left$). At time $t = 1$, we choose two cells that modify the Reeb graph G_1 by splitting the edges $m - m$ and $s - s$ (*right*). Finally, selecting the four cells at time $t = 2$ permits to create two new edges in G_3 ($t = 3$).

5 Conclusion and Future Work

In this article, we have proposed a global system for dynamic approximation of irregular graphical elements by lines. When we consider a local refinement by inclusion, we are able to update the topology of an irregular 2-D object by locally modifying its associated Reeb graph. Then, the polygonal description of this object is updated by changing the reconstruction by lines, in respect to its

Fig. 4. 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 $[-1; 1] \times [-1; 1]$ *(top-left)* discretized in a set of cells by an algorithm described in [13]. We also present the result of our method on the right (A_3). For each update procedure, we show the recoded set of cells and the polygonalization *(left)*. The selected cells $P_1, P_2, ...P_n$ and their associated decomposition $S_{t+1}(P_1), S_{t+1}(P_2), ..., S_{t+1}(P_n)$ for the next iteration are presented *(center)*. The Reeb graph is presented in a circular format *(right)*. The considered arcs in the Reeb Graph are illustrated in a different color.

local geometry. We have also shown an application of our system for interactive approximation of implicit curves by lines. In future work, we would like to integrate our proposal into a complete software for 2-D interactive geometrical modelling.

References

[1] 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.

[2] 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.

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

[4] Cordella, L.P., 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.

[5] Debled, I., Feschet, F., 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.

[6] 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.

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

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

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

[10] 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.

[11] 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.

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

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

[14] Vacavant, A., Coeurjolly, D., Tougne, L.: Topological and geometrical reconstruction of complex objects on irregular isothetic grids. In *the 13th International Conference on Discrete Geometry for Computer Imagery (DGCI 2006)*. Szeged, Hungary, October 2006. *(to appear)*