ISSN 2007-9737

Boundary Tracing for Digital Objects of Triangular Pixels

Petra Wiederhold*

Instituto Politécnico Nacional,
Centro de Investigación y de Estudios Avanzados,
Departamento de Control Automático,
Mexico

petra.wiederhold@cinvestav.mx

Abstract. This paper presents a boundary tracing algorithm for digital objects made of triangular tiles, using two connectivity types based on edge- and vertex-adjacencies. The article introduces to the mathematical foundations on oriented adjacency graphs in triangular tilings and studies boundaries and contours of such objects. The proposed algorithm is illustrated using examples and compared with previously known algorithms of boundary determination for abstract structures containing triangular tiles.

Keywords. Boundary tracing, contour following, triangular tiling, connected objects of triangular pixels, triangular mosaic.

1 Introduction

Triangular and hexagonal tilings of the plane have received attention for several decades for 2D digital image modeling, as alternative to the tiling of square tiles which may be identified with the set of pixels or grid points in the discrete plane $(c\mathbb{Z})^2$ ($c \in \mathbb{R}, c > 0$). Triangular pixels have been used, for example, to study convexity properties of digital objects [13], to develop geometrical transformations [1], to design digital topological spaces and thinning algorithms [3, 7, 10, 11], and for data visualization [9]. The present article considers binary digital images defined on triangular tilings, where the pixels are identified with triangular tiles being all congruent each other. The digital objects of interest are represented as connected sets of tiles, using edge- and vertex-adjacency based connectivity types.

Boundary tracing, also called *contour tracing* or *contour following*, is a standard digital image processing segmentation method which determines the frontier of an object. Well-known boundary tracing algorithms for 4- and 8-connected objects in $(c\mathbb{Z})^2$ are described, for example, in the widely used textbooks [4, 5, 8], they can be equally used for sets of square tiles modelled by 4- or 8-adjacency graphs. Boundary tracing was generalized for edge-adjacency-connected subsets of rectangular tilings in [17].

The previous conference paper [18] presented an algorithm of boundary tracing for edge-adjacency-connected subsets of the tiling of equilateral triangles, equivalently for sets of triangular pixels. The resulting so-called canonical boundary path was used in [18] to determine the minimal perimeter polygon (MPP) for objects whose boundaries are digital Jordan curves. In [16], the properties of the canonical boundary path were further studied and exploited in an MPP algorithm for more general digital objects, which are certain edge-adjacency-connected sets of triangular tiles called regular complexes.

The present article continues studying boundaries in triangular tilings, it proposes a boundary tracing algorithm for connected digital objects of triangular tiles or pixels, permitting both edge-adjacency and vertex-adjacency based connectivity types. The algorithm is illustrated by examples and compared with previously known algorithms for triangular pixels. The resulting boundary sets and paths are studied under both connectivity types, where it arises as necessary to distinguish between boundaries and contours.

In the remaining, Section 2 introduces connectivity, objects and boundaries in triangular tilings.

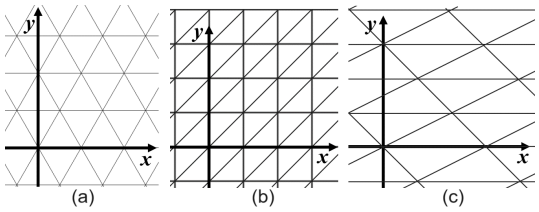

Fig. 1. Examples of triangular tilings, positioned as aligned to the Cartesian coordinate system, (a) shows the standard case of the tiling of equilateral triangles

Section 3 presents the boundary tracing algorithm, Section 4 studies properties of boundaries and contours. The proposed algorithm is compared to two algorithms previously known from the literature. Some conclusions complete the paper.

2 Connectivity, Digital Objects and Boundaries in Triangular Tilings

By a ***triangular tiling*** $\mathcal{T}$ we mean a family of congruent non-degenerated triangles named ***tiles***, whose union covers the plane, and where the intersection of any two tiles, either is empty, or is a common side of both tiles, called an ***edge*** of the tiling, or is a point which is a common vertex of six tiles. Hence $\mathcal{T}$ is an edge-to-edge polygonal tiling of the plane due to [6]. Let the tiling be positioned as shown in Figure 1, where the tiles form rows parallel to the x-axis. This can always be achieved since the union of any two triangular tiles which share an edge, forms a parallelogram. The tiles of $\mathcal{T}$ are of two types which occur alternating in each row. In the ***standard triangular tiling***, see Figure 1(a), the triangles are equilateral, and each row has alternating upright and inverted triangles.

Two distinct tiles of $\mathcal{T}$ which intersect each other, share a side, or, share a vertex. For each $T \in \mathcal{T}$, there are three other tiles sharing a side with T, and twelve tiles $T' \neq T$ which intersect T by sharing at least a vertex. Similarly to the well-known 4- and 8-adjacencies for square pixels, this gives rise to the following two adjacency relations on the set $\mathcal{T}$:

Definition 1 *Let T_1, T_2 be distinct tiles of a triangular tiling $\mathcal{T}$.*

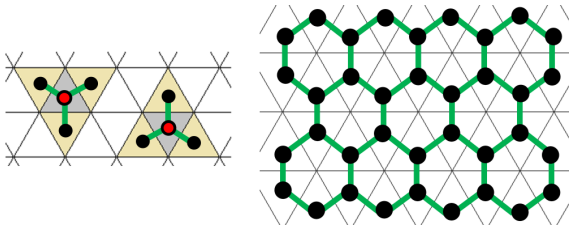

Fig. 2. 3-neighbors and a portion of the 3-adjacency graph for the standard triangular tiling

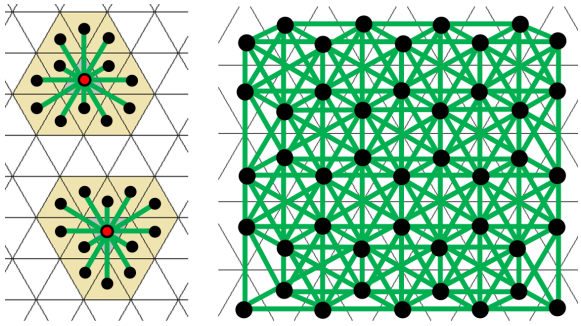

Fig. 3. 12-neighbors and a portion of the 12-adjacency graph for the standard triangular tiling

*– T_1, T_2 are called **edge- adjacent** or **3-adjacent** or **3-neighbors** if they share a common side.*

*– T_1, T_2 are called **vertex- adjacent** or **12-adjacent** or **12-neighbors** if they intersect each other, that is, they have a common vertex.*

*For $k \in \{3, 12\}$, if T_1, T_2 are **k-neighbors**, we say that T_1 is a k-neighbor of T_2.*

The k-adjacency, $k \in \{3, 12\}$, is a non-reflexive symmetric binary relation on $\mathcal{T}$. 3-neighbors also are 12-neighbors, but generally not vice versa. The k-adjacency generates a graph $G(\mathcal{T})$ called **k-*adjacency graph*** with node set $\mathcal{T}$, its edges are given by the pairs of k-neighbors, see Figures 2 and 3 where each tile is presented by its centre point drawn as a black dot, and the graph edges are depicted as green straight line segments. These centre points form a discrete set in the plane which is distinct from $\mathbb{Z}^2$ but could be the support (i.e., the set of pixels) of a digital image. The 3-adjacency graph is planar, which means that it can be presented in $\mathbb{R}^2$ drawing its nodes as points and its edges by line or curve segments such

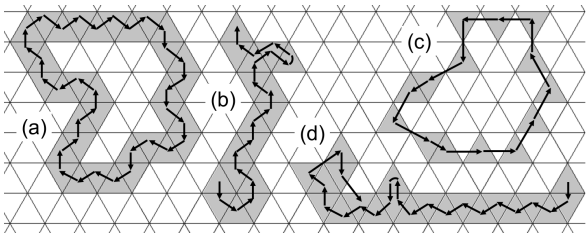

Fig. 4. In the standard triangular tiling, (a) shows a Jordan 3-curve, (b) a non-simple 3-path, (c) a Jordan 12-curve and (d) a non-simple 12-path

that these intersect each other only in points that represent nodes. The 12-adjacency graph is not planar since it cannot be presented in such a way.

Graph theory provides paths and connectivity: a **k-*path*** is a sequence of tiles $(T_1, T_2, \cdots, T_n)$ such that T_i is a k-neighbor of T_{i+1}, $i \in \{1, \cdots, n-1\}$. It is called a ***closed k-path*** if also T_k is a k-neighbor of T_1. A set $A \subset \mathcal{T}$ is **k-*connected*** if any two tiles of A are connected via a k-path in A. If A has at least two tiles, $T \in A$ is an ***end tile*** if T is k-adjacent to exactly one other tile of A.

For any set of tiles $\mathcal{C} \subset \mathcal{T}$ and $k \in \{3, 12\}$, a subset $A \subset \mathcal{C}$ is called a ***k-component*** of $\mathcal{C}$ if it is a maximal k-connected subset of $\mathcal{C}$, that is, A is k-connected, and for any tile $T \in \mathcal{C} \setminus A$, $A \cup \{T\}$ is not k-connected. Clearly, if $\mathcal{C}$ is k-connected, $\mathcal{C}$ coincides with its unique k-component.

Recall that the tiles of a triangular tiling are of two types. Note that all 3-neighbors of a triangular tile of a given type are tiles of the other type, hence the tiles in any 3-path alternately are of both types. In particular, a 3-path in the standard triangular tiling has alternately upright and inverted triangles.

Analogous to the well-known simple curves in the 4- and 8-neighborhood graphs of $\mathbb{Z}^2$ [8], a ***simple k-path*** or ***Jordan k-curve*** is a closed k-path whose each element has exactly two k-neighbors in this path, see Figure 4.

For any set of tiles $\mathcal{C} \subset \mathcal{T}$, let us denote its point set union by $|\mathcal{C}| = \bigcup \mathcal{C} = \{p \in \mathbb{R}^2 : p \in T \text{ for a tile } T \in \mathcal{C}\} \subset \mathbb{R}^2$. We define a ***(digital) object*** as any finite non-empty subset $\mathcal{C}$ of $\mathcal{T}$.

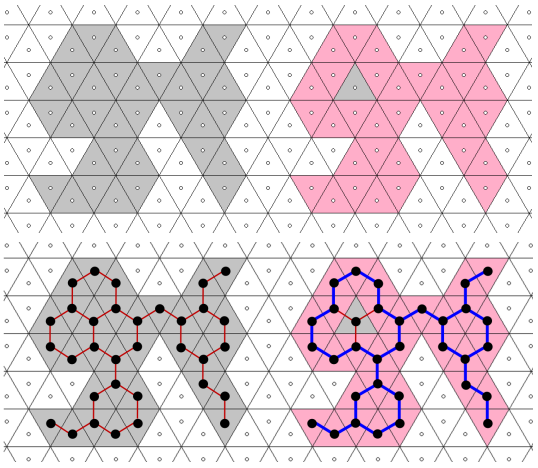

Fig. 5. A 3-connected object (shaded grey) with its boundary highlighted in pink. The subgraph induced by the object in the 3-adjacency graph is shown presenting its tiles as black dots (being the graph nodes), the object graph edges as brown lines, and the graph edges between boundary tiles as thick blue lines

Definition 2 *(from [12, 13]) Let $\mathcal{T}$ be a triangular tiling and $\mathcal{C} \subset \mathcal{T}$ an object. The set of tiles of $\mathcal{C}$ that intersects the topological frontier $fr(|\mathcal{C}|)$, is called the **boundary** $\mathcal{B}(\mathcal{C})$ of $\mathcal{C}$. Any closed k-path, for $k \in \{3, 12\}$, that consists of all tiles of $\mathcal{B}(\mathcal{C})$, is named a **boundary k-path** of $\mathcal{C}$.*

Whereas $\mathcal{B}(\mathcal{C})$ is a uniquely defined set, $\mathcal{C}$ may have several boundary k-paths with distinct properties. If $\mathcal{B}(\mathcal{C})$ is not k-connected then $\mathcal{C}$ has no boundary k-path at all. Figure 5 presents a 3-connected object with its boundary due to Definition 2. This object has several boundary 3-paths and 12-paths but none of them is simple since the object (as well as its boundary) has end tiles. Under the 3-connectivity, the object also has thin parts where any boundary 3-path must pass twice. Since any 3-connected object also is 12-connected, it can be considered as a subgraph of the 12-adjacency graph as shown in Figure 6.

Figures 7 and 8 show a 12-connected object with its boundary, also presented as a subgraph of the 12-adjacency graph. This object is not 3-connected, it has thirteen 3-components.

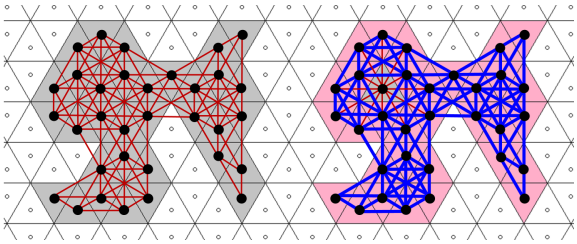

Fig. 6. The object from Figure 5 and its boundary are shown as induced subgraphs of the 12-adjacency graph, the object graph edges are drawn as brown lines, the graph edges between boundary tiles as bold blue lines

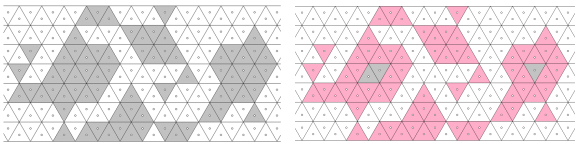

Fig. 7. A 12-connected object (shaded grey) with its boundary tiles highlighted in pink

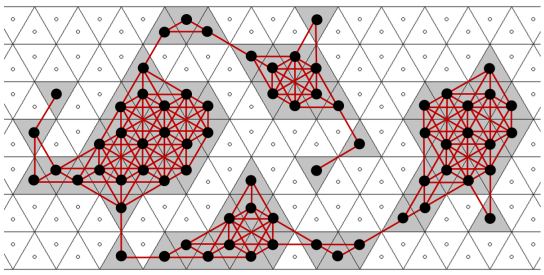

Fig. 8. The object from Figure 7 is presented as induced subgraph of the 12-adjacency graph

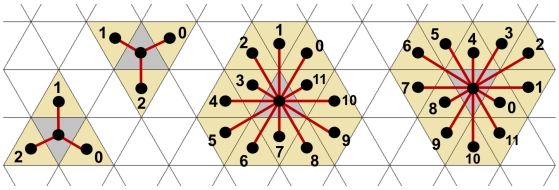

Fig. 9. Freeman codes for 3-neighbors and 12-neighbors in the standard triangular tiling

3 Boundary Tracing and Contours for 3- and 12-Connected Objects

The previous articles [16, 18] proposed a boundary tracing algorithm for 3-connected objects in the standard triangular tiling.

The more general Algorithm 1 presented in this section is suitable for any k-connected object $\mathcal{C}$ with $k \in \{3, 12\}$ in a triangular tiling $\mathcal{T}$, supposing that $\mathcal{C}$ has at least two tiles and $|\mathcal{C}| \subset \mathbb{R}^2$ has no hole. We also assume that there exists a finite set $\mathcal{N}$ with $\mathcal{C} \subset \mathcal{N} \subset \mathcal{T}$ such that the *background tiles* from $(\mathcal{N} \setminus \mathcal{C})$ fully surround the object $\mathcal{C}$.

Similar as known for adjacency graphs for square pixels, for any tile $T \in G(\mathcal{N})$, we employ the *Freeman code* to represent the movement direction from T to each of its k-neighbors T', which is coded by a number $f(T, T') \in \{0, 1, 2, \cdots, k-1\}$. As result, any k-path in $G(\mathcal{N})$ may be described by its *Freeman chain code*, see Figure 9.

In the standard triangular tiling, a 3-path has alternately upright and inverted triangles. This is not in general true for a 12-path which uses more possible movements. So, a 3-path can be reconstructed from its Freeman chain code whenever the type of the starting tile is known. For reconstructing a 12-path, it would be necessary to store also the types of tiles, together with the Freeman codes.

Boundary tracing starts with finding a first boundary tile. This can be achieved, for example, by detecting the leftmost tile T_0 of the top row of $\mathcal{C}$, as included in Algorithm 1, by scanning $\mathcal{N}$ on horizontal rows from the left to the right. Clearly then $T_0 \in \mathcal{B}(\mathcal{C})$, and T_0 has a left 3-neighbor q in the same horizontal row (independently of the connectivity type of $\mathcal{C}$), where q belongs to the background. In particular, using the Freeman code as in Figure 9 for the standard triangular tiling, if T_0 is an upright triangle and $f(T_0, q) = 1$, or, if T_0 is an inverted triangle and $f(T_0, q) = 2$, then $q \notin \mathcal{C}$.

For each last boundary tile T_n found, Algorithm 1 determines the Freeman code $b = f(T_n, T_{n-1})$, that is, of the direction from T_n back to T_{n-1}. Then it inspects the tile T' which lies in direction $(b+1) \mod k$, if T' belongs to $\mathcal{C}$ then it is the next boundary tile found. Otherwise, the tile T' lying in direction $(b+2) \mod k$ is inspected, it is the next boundary tile if it lies in $\mathcal{C}$. If not, the process is continued, inspecting the tiles T' in directions $(b+3) \mod k$, $(b+4) \mod k$, $\cdots$, $(b+(k-1)) \mod k$, until for the first time, T' is found to belong to $\mathcal{C}$. If all these inspected tiles T' are in the

background, then the tile in direction $(b+k) \mod k$ belongs to $\mathcal{C}$ and is the next boundary tile. The last case only occurs if T_n is an end tile of $\mathcal{C}$, we have then $T_{n+1} = T_{n-1}$.

Algorithm 1 determines a uniquely defined k-path *Cont*$(\mathcal{C}) = (T_0, T_1, T_2, \cdots, T_n)$ such that, when boundary tracing would be continued, the next tiles found would be, again, $T_{n+1} = T_0, T_{n+2} = T_1$. Step 2 will found T_0 again, when boundary tracing is finished, but this may occur also if *Cont*$(\mathcal{C})$ touches itself at T_0 but continues differently, the End Condition Test distinguishes between both situations. Since the Freeman codes from Figure 9 generate counterclockwise orders of the k-neigbors, the resulting k-path goes through the boundary in counterclockwise sense.

For the standard triangular tiling and $k = 3$, with Freeman codes fixed as in Figure 9, Step 1 of Algorithm 1 must set $b := 1$ if T_0 is an upright triangle, and $b := 2$ if T_0 is an inverted triangle.

In the previous works [16, 18] where exclusively 3-connected objects were considered, the resulting 3-path of Algorithm 1 was called *canonical boundary path* of $\mathcal{C}$. We will see in Section 4 that for 12-connectivity, it is necessary to distinguish between a boundary k-path due to Definition 2 and the list constructed by Algorithm 1. This justifies the following definition.

Definition 3 *Let $\mathcal{C} \subset \mathcal{T}$ be a k-connected object, $k \in \{3, 12\}$ in a triangular tiling $\mathcal{T}$, such that $\mathcal{C}$ has at least two tiles and $|\mathcal{C}| \subset \mathbb{R}^2$ has no hole. Any k-path $b = (T_0, T_1, \cdots, T_n)$ of tiles of $\mathcal{C}$ which, up to appropriate shifting of the cyclic sequence b, coincides with the sequence Cont$(\mathcal{C})$ determined by the Boundary Tracing Algorithm (Algorithm 1), is called the* ***k-contour*** *of $\mathcal{C}$.*

Figure 10 presents the 3-contour obtained by Algorithm 1 of a 3-connected object, comparison with Figure 5 reveals that the set of tiles belonging to the 3-contour coincides with the boundary due to Definition 2, the 3-contour is a boundary 3-path. The situation is distinct for the 12-connected object shown in Figure 11, where not all boundary tiles participate in the 12-contour.

Algorithm 1 has linear time complexity depending on the number of boundary tiles that belong to the k-contour, $k \in \{3, 12\}$. Step 0 needs constant time to find T_0, and Step 1 performs at most k tests to determine T_1. For each last found tile T_n, Step 2 uses at most k tests to determine T_{n+1}. Note that Algorithm 1 does not inspect all object tiles if the object is distinct from its boundary.

Algorithm 1 Boundary tracing for a k-connected object $\mathcal{C}$ in a triangular tiling, where $\mathcal{C}$ has at least two tiles, and $|\mathcal{C}| \subset \mathbb{R}^2$ has no hole.

Input: A finite subset $\mathcal{N}$ of a triangular tiling with Freeman code of the k neighbors fixed for both types of tiles, an object $\mathcal{C} \subset \mathcal{N}$ which is completely surrounded by tiles from the background $(\mathcal{N} \setminus \mathcal{C})$.

Output: List *Cont*$(\mathcal{C})$, a k-path of boundary tiles of the object $\mathcal{C}$.

Step 0: Find the leftmost tile T_0 of the top horizontal row of $\mathcal{C}$, then go to Step 1.

Step 1: Initialize the list *Cont*$(\mathcal{C}) := (T_0)$. Set b as the Freeman code $f(T_0, q)$ where the background tile q is the left 3-neighbor of T_0 lying in the same horizontal row as T_0. Then perform iteratively $b := (b+1) \mod k$. For each such b, determine whether the tile T' that satisfies $f(T_0, T') = b$, belongs to $\mathcal{C}$. While this is false, b is augmented to continue searching for a tile of $\mathcal{C}$. (In last instance, such T' is found when b was increased k times.) If $T' \in \mathcal{C}$ for the first time, add $T_1 = T'$ to *Cont*$(\mathcal{C})$ that becomes *Cont*$(\mathcal{C}) = (T_0, T_1)$. Then go to Step 2.

Step 2: Determine the direction b as the Freeman code $f(T_n, T_{n-1})$ from the current list *Cont*$(\mathcal{C}) = (T_0, T_1, T_2, \cdots, T_n)$. Then perform iteratively $b := (b+1) \mod k$. For each b analyze whether the tile T' that satisfies $f(T_n, T') = b$, belongs to $\mathcal{C}$. While $T' \notin \mathcal{C}$, b is increased to continue searching for a tile of $\mathcal{C}$. (Such T' is found as latest when b was increased k times.) When for the first time $T' \in \mathcal{C}$, proceed to the End Condition Test.

End Condition Test:

if $T' \neq T_0$ **then** add T' to *Cont*$(\mathcal{C})$, then go to Step 2.

else (now $T' = T_0$) add T' to *Cont*$(\mathcal{C})$. Then run Step 2 with the current list *Cont*$(\mathcal{C})$ only to obtain a new next boundary tile $T' \in \mathcal{C}$. Then,

    if $T' = T_1$ **then** remove the last tile from *Cont*$(\mathcal{C})$, then STOP.

    else add T' to *Cont*$(\mathcal{C})$, then go to Step 2.

    end if

end if

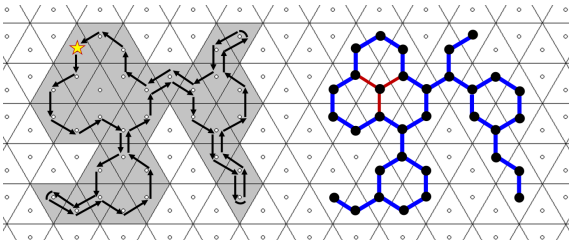

Fig. 10. The 3-contour of the object from Figure 5 is illustrated by black arrows, it starts at the leftmost tile of the top row which is marked by a star in the figure, its Freeman chain code begins as $(2, 2, 2, 0, 0, 0, 2, 2, 2, 1, 0, 0, 0, 1, \cdots)$. The path has alternating upright and inverted triangles. The right figure shows the 3-contour as induced 3-adjacency subgraph, each tile is presented by its centre point, the graph edges between boundary tiles are drawn bold in blue and the other object graph edges as brown lines

4 Boundaries and Contours

By the construction performed by Algorithm 1, any k-contour is a closed k-path which lies within the boundary due to Definition 2. For 3-connected objects, Algorithm 1 coincides with the boundary tracing algorithm from [16, 18] where only 3-connectivity was considered and the resulting 3-path was called the canonical boundary path of $\mathcal{C}$. This is justified by the following property.

Lemma 1 *Let $\mathcal{C}$ be a 3-connected object with at least two tiles and such that $|\mathcal{C}|$ has no hole, in a triangular tiling. Then the following is true:*

(1) The boundary $\mathcal{B}(\mathcal{C})$ is 3-connected.

(2) The 3-contour Cont$(\mathcal{C})$ is a boundary 3-path.

(3) $\mathcal{B}(\mathcal{C})$ coincides with the set of tiles that belong to Cont$(\mathcal{C})$.

(4) Any boundary 3-path that traces the boundary counterclockwise, that is, which corresponds to visit $fr(|\mathcal{C}|)$ counterclockwise, coincides with Cont$(\mathcal{C})$ or a shifted version of Cont$(\mathcal{C})$.

Proof: Let $\mathcal{C}$ be a 3-connected object satisfying the hypothesis. Then the boundary $\mathcal{B}(\mathcal{C}) \subset \mathcal{C}$ has at least two tiles. By the hypothesis, $fr(|\mathcal{C}|)$ is a Jordan curve[1] which is touched from its interior by any boundary tile T, where $T \cap fr(|\mathcal{C}|)$ is a side or a vertex of T.

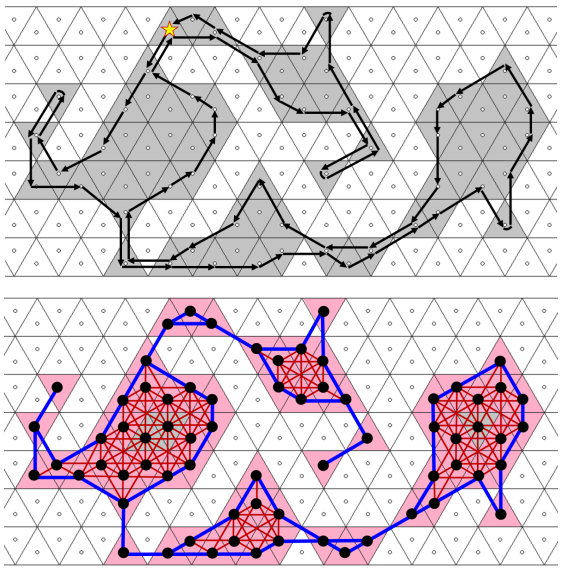

Fig. 11. The 12-contour of the object from Figure 7 is illustrated by black arrows. Figure below: the graph edges between tiles belonging to the 12-contour *Cont*$(\mathcal{C})$ are drawn as thick blue lines, the other object graph edges as brown lines. In the underlying tiling, the boundary tiles are highlighted in pink

(1) The curve $fr(|\mathcal{C}|)$ is made of sides of triangles which are boundary tiles. Let T_1, T_2 tiles in $\mathcal{B}(\mathcal{C})$ that intersect only in a vertex v (then T_1, T_2 are not edge-neighbors). Assume that s_1 is a side of T_1 and s_2 a side of T_2 such that $\{v\} = T_1 \cap T_2 = s_1 \cap s_2$ and $s_1 \cup s_2$ belongs to $fr(|\mathcal{C}|)$, then $v \in fr(|\mathcal{C}|)$. Since $\mathcal{C}$ is 3-connected, there exists a tile $T_3 \in \mathcal{C}$ with vertex v, hence $T_3 \in \mathcal{B}(\mathcal{C})$ and T_3 is a 3-neighbor of T_1 and of T_2. In consequence, $\mathcal{B}(\mathcal{C})$ is 3-connected.

(2) Algorithm 1 follows the curve $fr(|\mathcal{C}|)$ without interruption to include into *Cont*$(\mathcal{C})$ all tiles of $\mathcal{C}$ that intersect $fr(|\mathcal{C}|)$, no matter whether this intersection is a triangle side or a vertex. As result,

[1] Recall that a Jordan curve is a closed planar curve that does not touch itself. Formally, it is a set $\gamma = f([0,1]) \subset \mathbb{R}^2$ where $f : [0,1] \subset \mathbb{R} \to \mathbb{R}^2$ is a continues function which is injective on $[0,1)$ and such that $f(0) = f(1)$. A Jordan curve has a well-defined interior $Int(\gamma)$ being a bounded open region encircled by γ and exterior $\mathbb{R}^2 \setminus (Int(\gamma) \cup \gamma)$.

all tiles of $\mathcal{B}(\mathcal{C})$ are contained in *Cont*$(\mathcal{C})$, hence *Cont*$(\mathcal{C})$ is a boundary 3-path.
(3) By the construction made by Algorithm 1, *Cont*$(\mathcal{C})$ is a closed 3-path that lies in $\mathcal{B}(\mathcal{C})$. By the previous part (2), $\mathcal{B}(\mathcal{C})$ is contained in the set of tiles belonging to *Cont*$(\mathcal{C})$. Hence $\mathcal{B}(\mathcal{C})$ is equal to the set of tiles contained in *Cont*$(\mathcal{C})$.
(4) Let $B = (T_1, T_2, \cdots, T_n)$ be any (closed) boundary 3-path that traces the boundary $\mathcal{B}(\mathcal{C})$ in counterclockwise sense. Then, when traveling through the sequence B, each tile T_i visited touches $fr(|\mathcal{C}|)$ such that at the "right side" of T_i, the intersection $a_i = T_i \cap fr(|\mathcal{C}|)$ is a vertex or a side of T_i, with the result that $a_1 \cup a_2 \cup \cdots \cup a_n$ reproduces the curve $fr(|\mathcal{C}|)$ in counterclockwise sense. This implies that for any $T_i \in B$, the next tile $T_{i+1} \in B$ (computing $i+1$ and $i-1$ modulo n) is the first object tile which can be found when inspecting the 3-neighbors of T_i in counterclockwise order. This coincides with the procedure of Algorithm 1 to find each next tile of the 3-contour. Since $T_0 \in \mathcal{B}(\mathcal{C})$, it appears in B. But T_0 also belongs to *Cont*$(\mathcal{C})$. As a consequence, the cyclic sequence B coincides with *Cont*$(\mathcal{C})$ up to an appropriate shifting.

The following characterization of 3-contours is useful because it uses neither Algorithm 1 nor the curve $fr(|\mathcal{C}|)$.

Lemma 2 *Let $\mathcal{C}$ be a 3-connected object with at least two tiles and such that $|\mathcal{C}|$ has no hole, in a triangular tiling $\mathcal{T}$, and $T \in \mathcal{C}$. Then T belongs to the 3-contour Cont$(\mathcal{C})$ if and only if T has a 12-neighbor $Q \in (\mathcal{T} \setminus \mathcal{C})$.*

Proof: $T \in$ *Cont*$(\mathcal{C}) \subset \mathcal{B}(\mathcal{C})$ implies that T has a common point with $fr(|\mathcal{C}|)$ which is a vertex of T, hence T has a 12-neighbor $Q \in (\mathcal{T} \setminus \mathcal{C})$. On the other hand, if $T \in \mathcal{C}$ has a 12-neighbor $Q \in (\mathcal{T} \setminus \mathcal{C})$ then T intersects $fr(|\mathcal{C}|$, hence $T \in \mathcal{B}(\mathcal{C})$. By Lemma 1 and since $fr(|\mathcal{C}|$ is a Jordan curve, there exists a boundary 3-path that traces the boundary counterclockwise and contains T, hence this 3-path coincides with *Cont*$(\mathcal{C})$ which consequently contains T.

Applying Lemma 2, it is easy to find a tile T_0 belonging to the 3-contour of a 3-connected object $\mathcal{C}$ with at least two tiles and such that $|\mathcal{C}|$ has no hole. Then if $q \notin \mathcal{C}$ is a 3-neighbor of T_0, a modified version of Algorithm 1 can start with these tiles T_0 and q to construct the 3-contour.

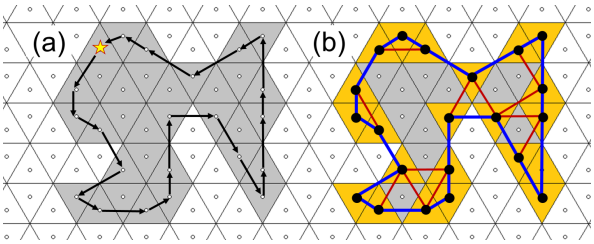

Fig. 12. The 3-connected object from Figure 10 also is 12-connected, hence Algorithm 1 can determine its 12-contour which is illustrated in (a) by black arrows. (b) shows the 12-contour as induced subgraph of the 12-adjacency graph, the tiles participating in the 12-contour are filled yellow, the graph edges corresponding to pairs in the 12-contour are drawn as bold blue lines, other graph edges between tiles belonging to the 12-contour are depicted as brown lines

Although the boundary of a 12-connected object is 12-connected, similar properties as those in Lemma 1 do not hold in general for 12-connected objects. Figure 12 shows the 12-contour of the 3-connected object from Figure 10. Recall also Figure 11 where the tiles belonging to the 12-contour form a proper subset of the boundary. These examples confirm that for 12-connected objects the boundary may not coincide with the set of tiles belonging to the 12-contour, hence the 12-contour is not a boundary 12-path, and not every boundary 12-path counterclockwise tracing the boundary coincides with the 12-contour.

5 Comparison with Previous Boundary Tracing Algorithms

Besides our papers [16, 18] where a boundary tracing algorithm for 3-connected objects was proposed, it seems that only two works from the literature consider boundary tracing in relation to sets of triangular tiles. The first one from [13] uses boundary tracing as preprocessing for an algorithm to determine the minimal perimeter polygon (MPP) of so-called normal complexes, which corresponds to vertex-connected objects without end tiles, in certain polygonal tilings. The second work, published in [14, 15] and briefly

summarized in [8], develops oriented adjacency graphs to model discrete sets and presents a generic contour tracing algorithm.

Both cited works do not explicitly treat boundary tracing for objects in triangular tilings, but the abstract structures developed there include that case. It turns out that the boundary tracing algorithm from [13] is erroneous, and that our Algorithm 1 is a specified and completed version of the "Border Mesh Determination" from [15].

5.1 Boundary Tracing in Polygonal Tilings

The articles [12, 13] are considered pioneering works on the development of the minimal perimeter polygon (MPP) and its application to study convexity properties of digital objects whose elements are identified with the tiles of polygonal tilings of the plane. Whereas rectangular tilings are used in [12], theoretical foundations are developed in [13] where the pixels are modelled by convex polygons belonging to a tiling $\mathcal{P}$ where for any two tiles T_1, T_2 that intersect each other, $T_1 \cap T_2$ is a full side or a vertex of both polygons and $T_1 \cup T_2$ forms a convex polygon. In [13], a non-empty finite set $\mathcal{C} \subset \mathcal{P}$ is called a *normal complex* if it has no end tile and $|\mathcal{C}|$ is simply connected in $\mathbb{R}^2$. For a normal complex $\mathcal{C} \subset \mathcal{P}$, the existence of a unique MPP of $\mathcal{C}$ is proved and an algorithm to determine the MPP is presented in [13].

The assumptions in [13] are satisfied for any triangular tiling $\mathcal{T}$. In our terminology, a normal complex $\mathcal{C} \subset \mathcal{T}$ is a 12-connected object without end tile where $|\mathcal{C}|$ has no hole. If $T_1, T_2 \in \mathcal{C}$ such that $T_1 \cap T_2 = \{v\}$ for a common vertex v of both tiles, v is called a *cut point* for $\mathcal{C}$ in [13]. Any 12-connected object which is not 3-connected, has a cut point, such as the object in Figure 13.

The MPP algorithm in [13] needs as preprocessing step the determination of a boundary 12-path due to Definition 2. For example, for the object in Figure 13, the "Algorithm M" in Section "MPP algorithm" of [13] aims first to construct the sequence $(c_1, c_2, c_3, c_3, c_4, c_5, c_6, c_7, \cdots)$ of boundary tiles, this order corresponds to visit the curve $fr(|\mathcal{C}|)$ in counterclockwise sense.

How to find this path of boundary tiles of an object $\mathcal{C}$, is only detailed in the Appendix "An illustrative implementation of Algorithm M" of [13] as follows: a data structure is constructed that contains for each tile T whether it belongs to $\mathcal{C}$ or not. The vertices of the convex polygon T are counterclockwise ordered and for each pair (v_i, v_{i+1}) of consecutive vertices of T, a pointer is established to the "bordering tile" T', that is, where $T \cap T' = \overline{(v_i, v_{i+1})}$ is a common side. This creates an order of pointers from T to all tiles that have a common side with T. Boundary tracing starts at a boundary tile $c_1 \in \mathcal{C}$ that has a side belonging to $fr(|\mathcal{C}|)$, and the pointer from c_1 to some tile $S \notin \mathcal{C}$. Then the subsequent pointers from c_1 are used to inspect the addressed tiles until finding a first object tile which is taken as c_2. For each last found boundary tile c_k, the pointer from c_k to c_{k-1} is considered and the next pointer from c_k starts to inspect the addressed tiles, until finding a first object tile which is taken as c_{k+1}. The process stops when $c_{k+1} = c_1$, then the boundary sequence is affirmed in [13] to have been completed.

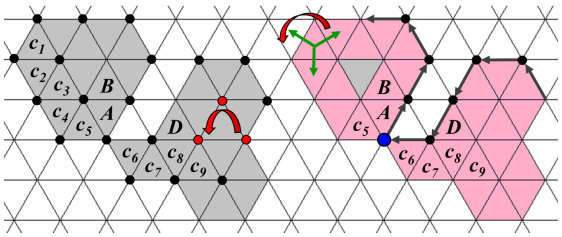

Fig. 13. A 12-connected object (shaded grey) in a triangular tiling which illustrates the performance of the boundary tracing algorithm from [13]. The boundary is highlighted in pink in the right figure. The algorithm pretends to find the sequence $(c_1, c_2, c_3, c_4, c_5, c_6, c_7, \cdots)$ of boundary tiles. See the text for explanations

In a triangular tiling, by the definition of the pointers in [13] an object is considered under 3-connectivity, and the boundary tracing determines the boundary 3-path of any 3-connected object in the same manner as our Algorithm 1, but no ending situation treatment is performed in [13]. More importantly, consider the 12-connected object $\mathcal{C}$ in Figure 13 which is not 3-connected. The boundary tracing of [13] correctly determines the first boundary tiles c_1, c_2, c_3, c_4, c_5. Then the pointer from c_5 back to c_4 is considered, the next

pointer from c_5 indicates a background tile, but the next pointer addresses the object tile denoted as A in Figure 13. Restarting from A, the tile B results as the next boundary tile found. The cut point $c_5 \cap c_6$ (drawn as blue dot in Figure 13) is ignored and c_6 cannot be found as the next boundary tile. So, the algorithm from [13] is erroneous for 12-connected objects that are not 3-connected.

5.2 Border Mesh Determination in Oriented Adjacency Graphs

The monographs [14, 15] develop theoretical foundations for digital image modelling based on adjacency graphs, originally named neighborhood structures or neighborhood graphs, this work much later was briefly summarized in Section 4.3 of [8]. The nodes of an *adjacency graph* (D, A) represent the elements of a discrete set $D \subset \mathbb{R}^2$, for example, the pixels of $(c\mathbb{Z})^2$ ($c \in \mathbb{R}, c > 0$) [4, 5, 8] or the centre points of all tiles of a triangular tiling. The graph edges are given by a non-reflexive symmetric binary relation A on D called adjacency or neighborhood relation, related elements are called neighbors. The 3- and 12-adjacencies on a triangular tiling are such relations.

In [14, 15], supposing that each node $p \in D$ has a finite number $v(p)$ of neighbors, these are cyclically ordered in $z(p) = (q_1, q_2, \cdots, q_{v(p)})$, the set $Z = \{z(p) : p \in D\}$ is called an *orientation* of (D, A). Then for $p, q, r \in D$ and the directed edges (p, q) and (q, r), (p, q) is defined as *predecessor* of (q, r) if $z(q) = (\cdots, p, r, \cdots)$. Using the predecessor relation, any directed edge of nodes generates a uniquely defined path called a *mesh*. An *oriented (adjacency) graph* (D, A, Z, M) is a graph (D, A) with an orientation Z and a set of meshes M. Assuming that (D, A) is connected, an *object* given as a finite proper subset C of D generates an induced oriented subgraph (C, A_C, Z_C, M_C) where A_C and Z_C are obtained by restricting the edges and the cyclic orders to nodes that belong to C. It turns out that $M_C \neq M$, a mesh from $(M_C \setminus M)$ is called a *border mesh* of C. Intuitively, if C is a connected object without holes, it has a unique border mesh which is a path tracing the boundary of C.

Algorithm 2.2-1 of [15] called "Border Mesh Determination", also reported as Algorithm 4.3 of [8], determines a border mesh given by a cyclic sequence of nodes $(p_0, p_1, p_2, \cdots, p_n)$ of an object C in any oriented adjacency graph (D, A, Z, M). The algorithm starts finding $p_0 \in \mathcal{C}$ that has a neighbor $q \notin \mathcal{C}$, then $q \in z(p_0)$. Inspecting the nodes that follows q in the sequence $z(p_0)$, the first object node provides p_1. Then, for each last found p_k, from the successors of p_{k-1} in $z(p_k)$, the first object node is taken as p_{k+1}. The algorithm stops when $p_{k+1} = p_0$, then the border mesh is affirmed to have been completed.

Now let $\mathcal{T}$ be a triangular tiling and $k \in \{3, 12\}$. For the k-adjacency graph $(\mathcal{T}, A_k)$, the cyclic orders of all k-neighbors of each tile $T \in \mathcal{T}$ due to the Freeman codes in Figure 9 provide an orientation Z_k and hence an oriented graph $(\mathcal{T}, A_k, Z_k, M_k)$. For a k-connected object $\mathcal{C}$ with at least two tiles and without holes, the border mesh is its k-contour. Our Algorithm 1 performs in Steps 1 and 2 the procedure suggested in the "Border Mesh Determination". Nevertheless, the "End Condition Test" of our algorithm additionally attends the situation that the k-contour could touch itself at the starting tile T_0, for example, when its cyclic sequence looks like $(T_0, T_1, \cdots, T_s = T_0, T_{s+1} \neq T_1, T_{s+2}, \cdots, T_n)$ such that $T_{n+1} = T_0$, $T_{n+2} = T_1$. Then it would not be correct to stop the algorithm when reaching again $T_0 = T_s$, as proposed in [8, 15].

6 Conclusion and Future Work

This paper presents a boundary tracing algorithm (Algorithm 1) for objects in triangular tilings modelled for $k \in \{3, 12\}$ as oriented k-adjacency graphs. For the special case $k = 3$, Algorithm 1 coincides with the preliminary version from [16, 18] which is applied there within another algorithm to find the minimal perimeter polygon (MPP) of certain type of objects. The MPP has been successfully applied, for example in [12, 13], to study convexity properties of such objects.

To develop our Algorithm 1, the "Border Mesh Determination" algorithm from [8, 14, 15] was completed and adapted to the oriented k-adjacency graphs ($k \in \{3, 12\}$) of triangular tilings.

This article also proves some properties of boundaries and k-contours of objects of triangular tiles which can be applied in future works for object description and recognition.

Future research includes to continue to study properties of 12-contours for 12-connected objects in triangular tilings, to consider other adjacencies, and to analyze digital Jordan curves and their topological separation properties. Another aim is to look for applications of k-contours in digital image analysis and pattern recognition, where modeling objects as sets of triangular tiles results as useful, for example, for the triangular covers from [2].

Acknowledgments

We would like to thank the reviewers for their comments and criticism which contributed to improve the manuscript.

References

1. **Avkan, A., Nagy, B., Saadetoglu, M. (2020).** Digitized rotations of 12 neighbors on the triangular grid. Annals of Mathematics and Artificial Intelligence, Vol. 88, pp. 833–857.
2. **Biswas et al., A. (2018).** Triangular covers of a digital object. Journal of Applied Mathematics and Computation, Vol. 58, pp. 667–691. DOI: 10.1007/s12190-017-1162-8.
3. **Deutsch, E. (1972).** Thinning algorithms on rectangular, hexagonal, and triangular arrays. Communications of the ACM, Vol. 15(9), pp. 827–837.
4. **Gonzalez, R., Woods, R. (2018).** Digital Image Processing (Global Edition). Pearson Education Limited, USA, 4rd edition.
5. **Gonzalez, R., Woods, R., Eddins, S. (2020).** Digital Image Processing using Matlab. Gatesmark Publishing LLC, USA, 3rd edition.
6. **Grünbaum, B., Shephard, G. (1978).** Tilings and Patterns. W.H. Freeman and Company, USA.
7. **Kardos, P., Palágyi, K. (2015).** Topology preservation on the triangular grid. Annals of Mathematics and Artificial Intelligence, Vol. 75, pp. 53–68. DOI: 10.1007/s10472-014-9426-6.
8. **Klette, R., Rosenfeld, A. (2004).** Digital Geometry - Geometric Methods for Digital Picture Analysis. Morgan Kaufmann Publisher, USA.
9. **Nagy, B. (2022).** Diagrams on the hexagonal and on the triangular grids. Acta Polytechnica Hungarica, Vol. 19(4), pp. 27–42.
10. **Nagy, B. (2024).** A Khalimsky-like topology on the triangular grid. In **Brunetti, S., et al.**, editors, Proc. of DGMM 2024. Springer, LNCS 14605, Switzerland, pp. 150–162. DOI: 10.1007/978-3-031-57793-2_12.
11. **Saha, P., Rosenfeld, A. (2001).** Local and global topology preservation on locally finite sets of tiles. Information Sciences, Vol. 137, pp. 303–311. DOI: doi:10.1016/S0020-0255(01)00107-4.
12. **Sklansky, J., Chazin, R., Hansen, B. (1972).** Minimum perimeter polygons of digitized silhouettes. IEEE Trans. on Computers, Vol. 21(3), pp. 260–268. DOI: 10.1109/TC.1972.5008948.
13. **Sklansky, J., Kibler, D. (1976).** A theory of nonuniformly digitized binary pictures. IEEE Trans. on Systems, Man, and Cybernetics, Vol. 6(9), pp. 637–647. DOI: 10.1109/TSMC.1976.4309569.
14. **Voss, K. (1988).** Theoretische Grundlagen der digitalen Bildverarbeitung. Akademie-Verlag Berlin, German Democratic Republic.
15. **Voss, K. (1993).** Discrete Images, Objects, and Functions in $\mathbb{Z}^n$. Springer, Serie Algorithms and Combinatorics Vol.11, USA.
16. **Wiederhold, P. (2024).** Computing the minimal perimeter polygon for digital objects in the triangular tiling. Discrete Applied Mathematics, Vol. (under review), pp. 1–33.
17. **Wiederhold, P. (2024).** Computing the minimal perimeter polygon for sets of rectangular tiles based on visibility cones. Journal of Mathematical Imaging and Vision, Vol. 66, pp. 873–903. DOI: 10.1007/s10851-024-01203-z.
18. **Wiederhold, P. (2024).** On the minimal perimeter polygon for digital objects in the triangular tiling. In **Mezura-Montes, E., et al.**, editors, MCPR 2024 Mexican Conf. on Pattern Recognition. Springer, LNCS Vol. 14755, Switzerland, pp. 141–154. DOI: 10.1007/978-3-031-62836-8_14.

Article received on 20/06/2024; accepted on 15/09/2024.
**Corresponding author is Petra Wiederhold.*