Graphical Models and Image Processing **61,** 16–43 (1999)
Article ID gmip.1999.0488, available online at http://www.idealibrary.com on IDEAL®

Euclidean Paths: A New Representation of Boundary of Discrete Regions[1]

Jean-Pierre Braquelaire and Anne Vialard

Laboratoire Bordelais de Recherche en Informatique, UMR 5800, Université Bordeaux 1, 351, cours de la Libération, F-33405 Talence, France
E-mail: Jean-Pierre.Braquelaire@labri.u-bordeaux.fr, Anne.Vialard@labri.u-bordeaux.fr

Received February 5, 1998; accepted January 21, 1999

The aim of this work is to provide a means to approximate the real boundary underlying the discrete boundary of a digitized 2D region. We require that the sampling of the reconstructed boundary be exactly the discrete one. To this end, we propose a new representation of the boundary of a discrete region that we call Euclidean paths. This paper fully describes the method used to build a Euclidean path and gives several examples of applications both for image analysis and image synthesis.

1. INTRODUCTION

A discrete region generally comes from the digitization of a real one. When the discrete region is the result of an image segmentation there is no information about the boundary of the original real region. The drawbacks of the boundary of the discrete region (or discrete boundary) are directly involved by the loss of information resulting from the digitization. First of all, discrete boundaries are intrinsically jagged, a feature which appears particularly when they are displayed with a larger scale. Second, geometric transforms are not well-defined in the discrete plane which is the support of discrete images. Finally, digitization does not allow us an accurate estimation of all geometric characteristics of discrete boundaries.

We thus aim at reconstructing a real boundary from its digitized representation. We require that two constraints be satisfied:

1. The real representation of the discrete boundary should not modify the initial digital information. In other words the discrete boundary can be recovered from its "real" representation. As a matter of fact it is important in some applications, for example in medical imaging, to be accurate to within a pixel.

[1] This work is supported by a grant from the Région Aquitaine and the CNRS.

1077-3169/99 $30.00

2. The real representation should not depend on the traversal of the discrete boundary (starting point and orientation). This second constraint preserves a correct definition of the boundary: a region and its complement share exactly the same boundary. Moreover, this condition provides an automatic reconstruction of the real boundary.

Usual approximation techniques such as polygonal approximation or spline curves can be considered as a means to represent a discrete boundary. The polygonal approximation of a discrete boundary consists in approximating it by consecutive line segments. This technique has many drawbacks, such as the angular join points between two successive segments. Moreover, the result of a polygonal approximation depends highly on the particular traversal of the discrete boundary, and thus, the second condition is not satisfied.

Spline curves can be defined generally as mathematical models which associate a continuous representation with a discrete set of points. The points defining a spline curve are called control points. It does not seem possible to use spline curves for automatically smoothing discrete boundaries. As a matter of fact, for representing a discrete boundary by a spline, we could choose either to have all the boundary points as control points or to select some of them. Some simple tests with uniform B-splines have shown that the first solution is not possible: in order to eliminate oscillation in the resulting curve due to the jagged nature of the discrete boundary, we have to use splines of very high degree. This implies a very strong smoothing which does not preserve some characteristic points, such as angular points. If we choose, on the contrary, to keep only a few points of the boundary as control points, we have to choose these points correctly. This last solution involves either user intervention or an analysis of the discrete boundary in order to extract its characteristic points [18]. In both cases it is not possible to ensure no loss of information and the first condition is not satisfied.

Both approaches described above are either discrete or continuous. We propose here a mixed approach by defining a discrete boundary lying on a continuous support. The basic idea is to move each point of the discrete boundary from its initial position to a real point that is close enough, in order to smooth the boundary. In other words, we try to reconstruct the real boundary by moving the discrete boundary points in the Euclidean plane. This construction is called a Euclidean path.

Euclidean paths are not restricted to boundaries of discrete regions. They can be used to smooth either open or closed discrete curves, according either to 4-connectivity or to 8-connectivity. However, in this work we focus on the representation of discrete boundaries and thus we only consider close discrete curves, also called *discrete contours*. In the following section we state precisely which kinds of discrete contours are used to define the boundary of a discrete region. In Section 3 we give the general definition of a Euclidean path and we consider several construction methods of Euclidean paths based on usual smoothing techniques. Since none of these methods is satisfying we introduce a new one based on tangent computation along the processed discrete contour. Section 4 gives an algorithmic definition of the discrete tangent at a point of an interpixel boundary, and Section 5 describes the construction of tangent-driven Euclidean paths. Finally, we present in Section 6 several examples of applications of tangent-driven Euclidean paths.

2. INTERPIXEL BOUNDARIES

Let us first give the definitions necessary for this paper. A digital image is a rectangular grid of pixels. A pixel is represented either by a unit square or by a grid intersection. From

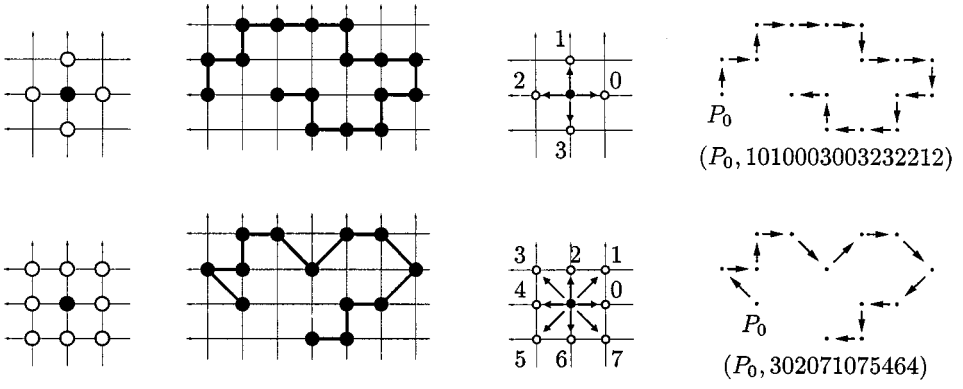


FIG. 1. The upper part of the figure illustrates 4-adjacency: the 4-neighbors of a discrete point, an example of 4-connected path, the four Freeman directions, and the Freeman code of the 4-connected path are represented from left to right. The discrete plane is represented by a grid. The lower part of the figure illustrates 8-adjacency similarly.

a theoretical point of view, we consider an infinite digital image in bijection with the plane $\mathbb{Z}^2$, also called the *integer plane* or *discrete plane*. A *discrete point* or *integer point* P is a point of the discrete plane. We denote by (X_P, Y_P) the coordinates of the point P in this plane. In this work we generally consider pixels represented by unit squares centered on integer points. As such the pixel of center (X_P, Y_P) is identified with the discrete point P.

Two discrete points P and Q are said to be *4-adjacent* if $|X_P - X_Q| + |Y_P - Y_Q| = 1$. They are said to be *8-adjacent* if $\max(|X_P - X_Q|, |Y_P - Y_Q|) = 1$. A 4-connected (resp. 8-connected) *discrete path* is a sequence of discrete points $(P_0, \ldots, P_n)$ such that $\forall i$, $1 \leq i \leq n$, P_{i-1} and P_i are 4-adjacent (resp. 8-adjacent). These definitions are illustrated in Fig. 1.

A discrete path composed of $n+1$ points $(P_0, \ldots, P_n)$ can be represented by its first point P_0 and its Freeman code, the sequence $(m_0, \ldots, m_{n-1})$, where m_i encodes the elementary move from P_i to P_{i+1} [13]. The four (resp. eight) elementary vectors between a discrete point and its neighbors are numbered from 0 to 3 (resp. 7). The elementary moves are numbered counterclockwise, the east move being numbered 0 (see Fig. 1).

According to the 4-connectivity (resp. 8-connectivity) two consecutive Freeman directions define a *quadrant* (resp. an *octant*). The first quadrant (resp. first octant) is the part of the plane bounded by the two half-lines starting from the origin of the coordinates system and of Freeman directions 0 and 1. The quadrants (resp. octants) are numbered counterclockwise (see Fig. 2).

A *discrete region* (or simply a region) is a connected set of pixels of an image. A region can be described explicitly by the set of its points or implicitly by its boundary. In the Euclidean plane the boundary of a region is a set of disjoint closed Jordan curves (i.e. continuous closed, not self-intersecting curves). Moreover, a closed Jordan curve separates the plane into two disconnected regions. The bounded region is said to be inside the curve and the other one is said to be outside [14]. This result is known as the Jordan theorem.

Two approaches are commonly used to define the boundary of a discrete region. The first one, which we call the *pixel-oriented approach*, consists in taking a subset of the pixels of the region. The boundary of an 8-connected region $\boldsymbol{R}$ is defined as the set of points of $\boldsymbol{R}$ having at least a 4-neighbor outside $\boldsymbol{R}$. The boundary of a 4-connected region $\boldsymbol{R}$ is similarly defined as the set of points of $\boldsymbol{R}$ having at least an 8-neighbor outside $\boldsymbol{R}$. See Fig. 3.

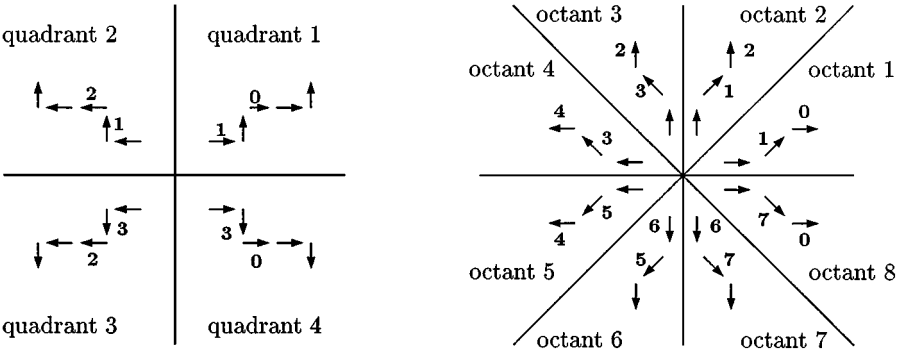


FIG. 2. The left (respectively right) part illustrates the convention used to number the four quadrants (resp. eight octants) and the associated Freeman directions.

These definitions raise the problem of transposing the topological properties of a real boundary into the integer plane. It is well known that the pixel-oriented approach does not provide a natural discrete version of the Jordan theorem [9, 15]. Moreover, with these definitions of a region boundary, two adjacent regions do not share a part of their boundary.

These problems are solved with the second approach, also called the *interpixel oriented* approach. It was introduced in 1970 by Brice and Fennema [8] for the purpose of image segmentation. A theoretical framework of interpixel boundaries was developed later, in particular by Kovalevsky [15] and by Ahronovitz *et al.* [1]. Interpixel contours can be seen as discrete 4-connected paths of the *half-integer plane*, which is the set of points in which both coordinates are integer multiples of $\frac{1}{2}$ (see [6, 11, 7]).

DEFINITION 1. The *half-integer plane* $\mathcal{P}_{1/2}$ is deduced from the integer plane $\mathbb{Z}^2$ by the translation of vector $(\frac{1}{2}, \frac{1}{2})$:

$$\mathcal{P}_{1/2} = \left\{ \left(i + \frac{1}{2}, j + \frac{1}{2} \right), (i, j) \in \mathbb{Z}^2 \right\}.$$

The half-integer plane has the same structure as the integer plane. We can naturally derive the notion of a half-integer path from the notion of an integer path. In the following we call, in a wider sense, a half-integer path a discrete path and a half-integer point a discrete point.

With the square representation of pixels the points of an interpixel boundary are located on the pixel corners. We can consider without loss of generality that regions are simply

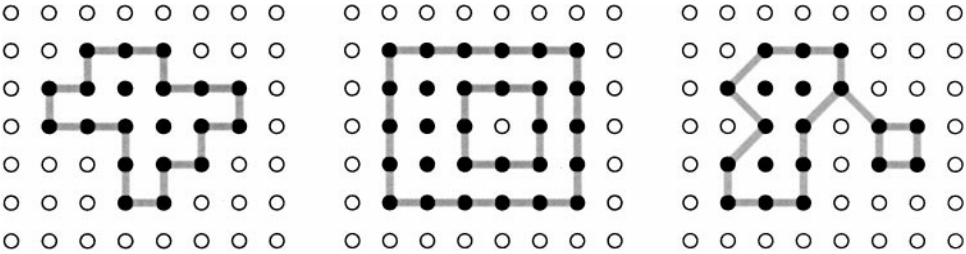


FIG. 3. The left and middle parts of the figure represent 4-connected contours which are pixel boundaries of 4-connected regions. The region of the middle part is holed and thus its boundary consists of two closed paths. The right part of the figure shows an 8-connected contour which is the pixel boundary of an 8-connected region.

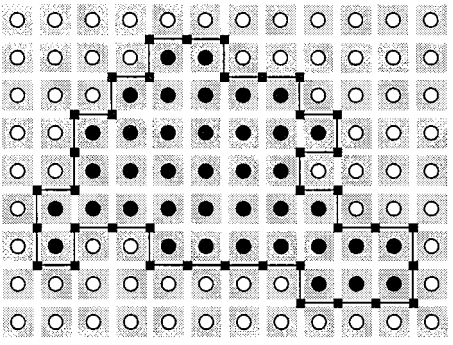


FIG. 4. By considering pixels as elementary squares (in gray), we can define the interpixel boundary of a discrete region (in black) as a discrete path of half-integer points.

connected regions, i.e. regions without holes. According to this hypothesis, the discrete boundary of a 4-connected region is a closed 4-connected half-integer path with no self-intersection (see Fig. 4). Such a path separates the integer plane into two disconnected discrete regions. This property is the transposition to the discrete plane of the Jordan theorem. Moreover, adjacent regions share their common part of a boundary.

In the following we consider the construction of Euclidean paths of half-integer region boundaries and, thus, of 4-connected discrete paths.

3. GENERAL EUCLIDEAN PATHS AND CONSTRUCTION METHODS

3.1. *General Model*

In the following, we denote by upper case letters the coordinates of the discrete points and by lower case letters the coordinates of the Euclidean points. We present here some definitions and basic properties on which the model of Euclidean paths lies.

DEFINITION 2. Let $P = (X, Y)$ be a discrete point. The *cell* of P is the set of points $(x, y) \in \mathbb{R}^2$ verifying:

$$|X - x| < \frac{1}{2}, \quad |Y - y| < \frac{1}{2}.$$

In other words, the cell of a point P is the open unit square centered on P.

DEFINITION 3. Let P be a discrete point. We say that the real point p is a *Euclidean point associated with* P if and only if p belongs to the cell of P.

The definition of a discrete point cell ensures the reversibility of the transformation of a discrete point P into an associated Euclidean point p; by rounding the coordinates of p to the nearest half-integer coordinates, we recover the coordinates of P. The two previous definitions are illustrated in Fig. 5.

DEFINITION 4. Let $\mathbf{\Pi}$ be the discrete path $(P_0, P_1, \ldots, P_N)$. By associating (in the sense of Definition 3) a Euclidean point p_i with each discrete point P_i, we obtain a sequence of real points $\overline{\mathbf{\Pi}} = (p_0, p_1, \ldots, p_n)$ that we call a *Euclidean path associated with* $\mathbf{\Pi}$.

As the transformation of a discrete point into an associated Euclidean point is reversible, the transformation of a discrete path into an associated Euclidean path is reversible, too.

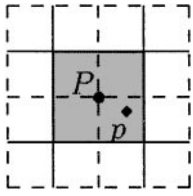


FIG. 5. A half-integer point P and an example of an associated Euclidean point p. The cell of P is displayed as a gray full square. The integer grid is represented by solid lines and the half-integer grid by dashed lines.

There is, however, an infinity of possible Euclidean paths associated with a discrete path. In other words, we transform $\mathbf{\Pi}$ to $\overline{\mathbf{\Pi}}$ by moving each point of $\mathbf{\Pi}$ inside its cell and the choice of each move is free. By choosing a relevant setting for each Euclidean point in the cell of the corresponding discrete point, we can smooth the initial discrete path. Figure 6 gives an example of a Euclidean path associated with a discrete path. In that example, the Euclidean points have been set in order to reduce the jags of the initial discrete path.

We present now some automatic construction methods for Euclidean paths. Most of them will not be retained and we explain why. We finally introduce an original construction method based on tangent recognition along the processed discrete path.

3.2. *Rewriting Methods*

A very simple technique to obtain a Euclidean path consists in using a rewriting system based on the patterns of the initial discrete path. Let us consider a closed 4-connected path $\mathbf{\Pi} = (P_0, \ldots, P_n)$. If we choose for example to process patterns composed of three points, the position of the Euclidean point p_i associated with P_i is given by the positions of P_{i-1}, P_i, and P_{i+1}($\mathbf{\Pi}$ being a closed path, we have $P_{-1} = P_n$ and $P_{n+1} = P_0$). Two cases occur:

- The points P_{i-1}, P_i, and P_{i+1} are in line. We can take in this case $p_i = P_i$.
- The points P_{i-1}, P_i, and P_{i+1} are not in line. As $\mathbf{\Pi}$ is 4-connected, they form a right angle. In this case, we choose to set p_i on the bisector of the angle $P_{i-1}\widehat{P_i}P_{i+1}$ in order to reduce the gaps along the discrete path.

Note that a displacement of length $\sqrt{2}/4$ provides a rectilinear Euclidean path for a discrete curve composed of alternating horizontal and vertical steps. This example of rewriting

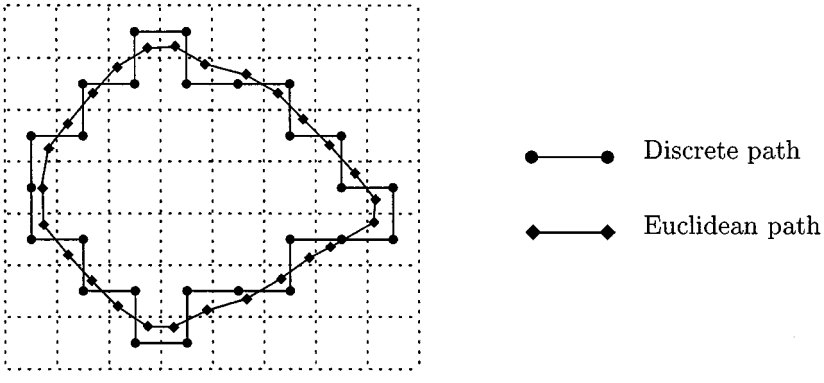


FIG. 6. The smoothed polyline represents a Euclidean path associated with a closed 4-connected discrete path. Note that the Euclidean path and the discrete one have the same number of points.

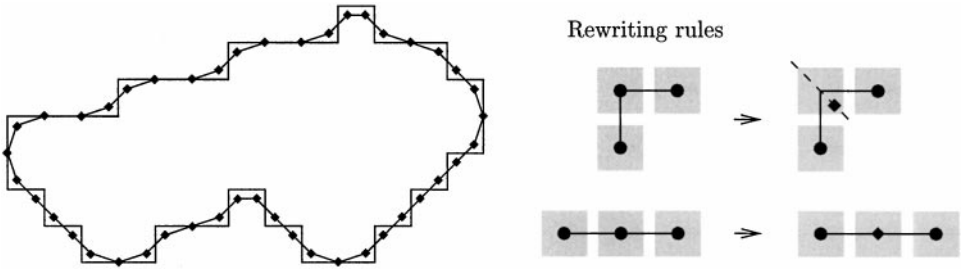


FIG. 7. Use of rewriting rules to construct a Euclidean path.

rules is used in Fig. 7. It allows us to correctly recover the vertical, horizontal, and diagonal line segments. On the other hand, line segments of different slopes produce oscillations along the path (see the upper left part of Fig. 7). We could take into account other orientations by increasing the size of the considered neighborhood around each point. The number of rewriting rules then increases exponentially. We thus do not retain this approach.

3.3. *Euclidean Path Construction Based on a Polygonal Approximation of the Initial Discrete Path*

The *polygonal approximation* of a discrete path is a decomposition of the path into a sequence of discrete line segments [10, 12, 19]. More formally, a polygonal approximation of a discrete path $\mathbf{\Pi} = (P_0, \ldots, P_n)$ is a sequence of discrete line segments $(\mathbf{\Pi}_0, \ldots, \mathbf{\Pi}_m)$ with $m \leq n$ and $\mathbf{\Pi}_0 = (P_0, \ldots, P_i)$, $\mathbf{\Pi}_1 = (P_i, \ldots, P_j), \ldots, \mathbf{\Pi}_m = (P_l, \ldots, P_n)$. Let us consider a 4-connected line segment. It is possible to associate with each of its discrete points a Euclidean point belonging to a real segment of the polygonal approximation. We can show that the point of intersection between the real line and the diagonal of each cell is strictly included in the cell. This construction is illustrated in Fig. 8. By applying this construction to each discrete segment $\mathbf{\Pi}_i$ composing $\mathbf{\Pi}$, we obtain a Euclidean path associated with $\mathbf{\Pi}$.

This method is obviously efficient for reconstructing polygon boundaries from their digitizations but it does not allow us to correctly restore a curved contour: the junction between two successive line segments are clearly visible (see Fig. 9). On this example, we can also notice an irregularity due to the choice of the first point of the disc polygonal approximation. This problem appears for all types of contours. Moreover, two different polygonal approximations (choice of the starting point, of the contour orientation, of the segment lengths) of the same boundary will give two different Euclidean paths. This solution is thus not satisfying.

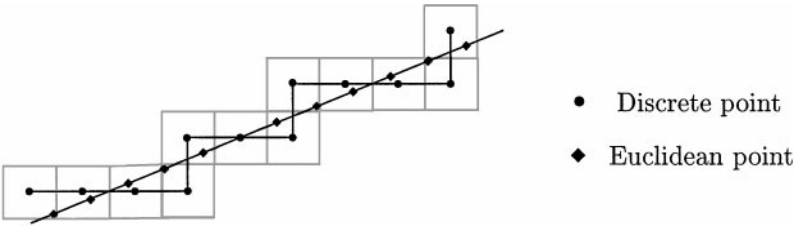


FIG. 8. Euclidean path deduced from a 4-connected discrete line segment.

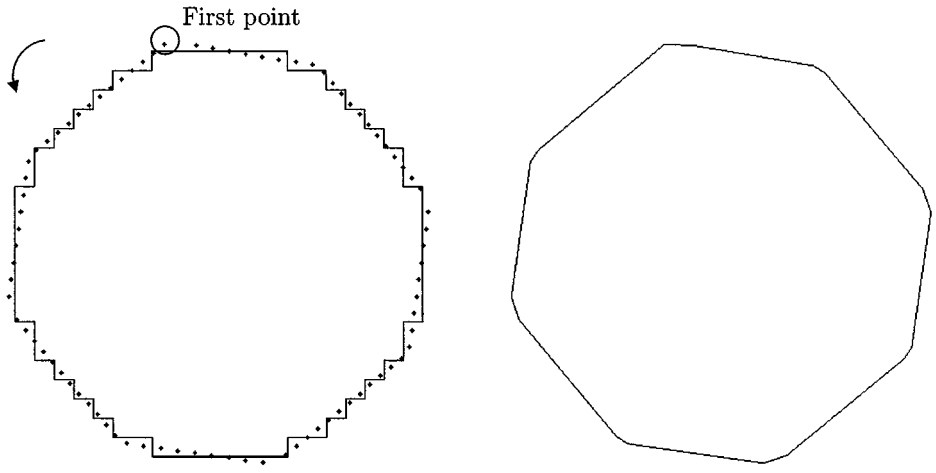


FIG. 9. Example of a Euclidean path based on the polygonal approximation of a discrete contour. The initial discrete path is the interpixel boundary of the discrete disc of radius 10 pixels.

3.4. *Euclidean Paths Based on a Discrete Convolution*

By analogy with smoothing techniques used in the field of signal processing, we can construct a Euclidean point associated with a discrete point by convolving the neighborhood of the discrete point.

Let us consider a normalized convolution mask $W_k = [w_{-k}, w_{-k+1}, \ldots, w_0, \ldots, w_k]$ with $\sum_{j=-k}^{j=k} w_j = 1$. The Euclidean point p_i associated with the discrete point P_i is obtained by the convolution computation

$$p_i = \sum_{j=-k}^{j=k} w_j P_{i+j}.$$

Two very frequent definitions of the mask W_k are:

- the triangular distribution (which gives a Gaussian distribution by iteration)

$$W_1 = \frac{1}{4}[1, 2, 1], \ldots, W_k = \frac{1}{(k+1)^2}[k + 1 - |i|]_{i=-k\ldots k};$$

- the binomial distribution

$$W_1 = \frac{1}{4}[1, 2, 1], \ldots, W_k = \frac{1}{2^{2k}}\left[C_{2k}^i\right]_{i=0\ldots 2k}.$$

Let us consider the effects of convolving a discrete path with a binomial mask. It is obvious that the smoothing effect is linked to the size of the mask. The example of Fig. 10 emphasizes the limits of this technique. If the mask size is small ($k = 2$), the sides of the square are not sufficiently smoothed and this appears as oscillations on the resulting contour. With a greater mask size ($k = 6$), the smoothing if effective but the obtained sequence of real points is obviously not a Euclidean path such as we have defined it (see Definition 4); the real points associated with the square corners are too distant from the discrete points. The initial discrete information is not preserved. Figure 11 shows this loss of information

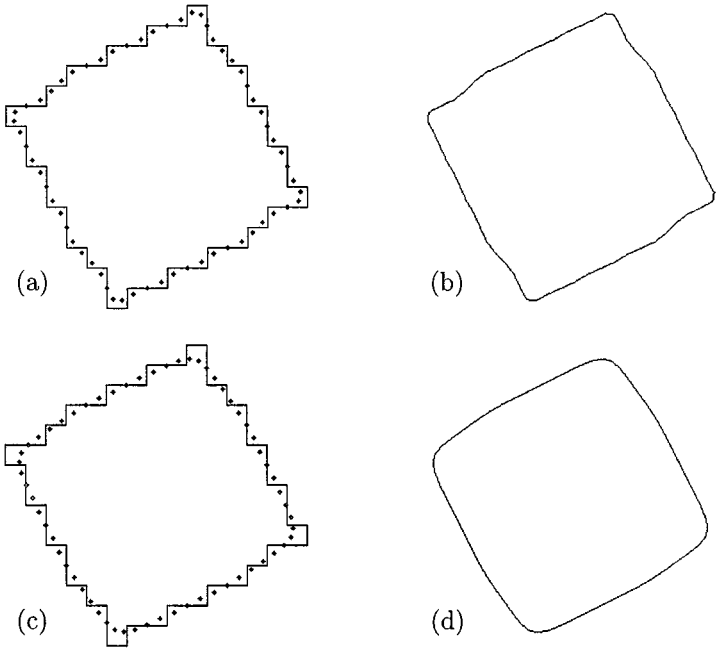


FIG. 10. The initial discrete contour is the interpixel boundary of the square of side 10 rotated 30°. Parts (a) and (b) (resp. (c) and (d)) result from a convolution of the discrete points with the binomial mask of parameter $k = 2$ (resp. $k = 6$).

on an example that we can consider as the worst case in 4-connexity. In short, it is not possible to smooth correctly any discrete path with a mask of fixed size without some loss of information.

3.5. *Tangent-Driven Euclidean Paths*

It is important to associate a unique Euclidean path to a given discrete boundary. If it was possible to associate several different Euclidean paths with a discrete boundary, for instance according to the point from which the processing is initiated or to some weighting parameters, it would also be possible to get two different Euclidean paths from the common

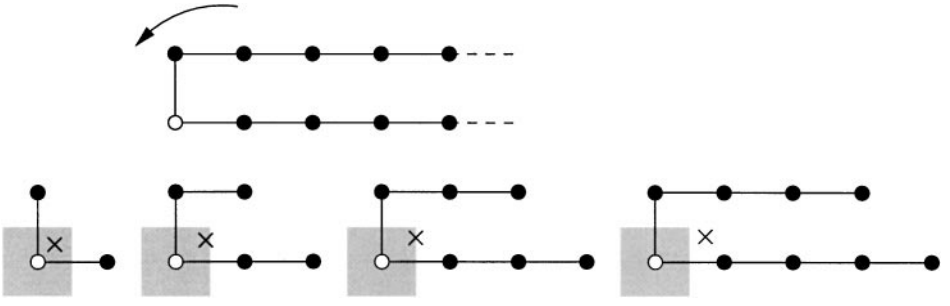


FIG. 11. The cell of the processed point (white point) is represented by a gray square. The real point obtained by convolution with a binomial mask is represented by a cross. Parameter k defining the size of the binomial mask varies from 1 (lower left part of the figure) to 4 (lower right part). From $k = 3$, the real point is outside of the cell.

boundary of two complementary regions. Thus the constraint 2, stated in the Introduction, would not be satisfied. For this reason we have developed a construction method independent of the traversal of the boundary. In other words the construction produces the same result, regardless of the starting point and the orientation of the boundary path. This is possible thanks to a definition of the discrete tangent at a boundary point. This definition depends only on the point and on its neighborhood. The construction is based on the computation of the discrete tangent at each point of the processed boundary. Given the discrete tangent, we give an approximation of the real tangent in the neighborhood of the considered boundary point. The Euclidean point associated with the boundary point is then located on this real tangent. The Euclidean paths built with this method are called *tangent-driven Euclidean paths*. Note that this solution processes a neighborhood adapted to each boundary point.

4. TANGENTS OF A DISCRETE BOUNDARY

This section is first devoted to the recognition of a discrete 4-connected line segment. The algorithm that we propose is based on an adapted version of Debled's incremental algorithm for the recognition of 8-connected segments [10]. It allows us to recognize a discrete 4-connected segment by adding incrementally a point at one or the other of its extremities. This work leads to an algorithmic definition of the discrete tangent at a point of an interpixel boundary.

4.1. *Discrete Line: Definition and Recognition Algorithm*

The basic definition of a discrete line used in this work is the arithmetic one given by Reveilles [17, 10].

DEFINITION 5 [17]. A *discrete line* $\boldsymbol{L}$ is the set of points (x, y) of $\mathbb{Z}^2$ which satisfy the double inequality $\mu \leq Ax - By < \mu + \omega$ with $A, B, \mu \in \mathbb{Z}, \omega \in \mathbb{N}$.

The four coefficients (A, B, μ, ω) are called the characteristics of the discrete line $\boldsymbol{L}$. The fraction A/B is the slope of $\boldsymbol{L}$. The integer μ describes its position in the discrete plane and ω its thickness. If ω is equal to $\max(|A|, |B|)$, $\boldsymbol{L}$ is an 8-connected discrete line. If ω is equal to $|A| + |B|$, $\boldsymbol{L}$ is a 4-connected one.

DEFINITION 6. The *upper and lower leaning lines* of a discrete line $\boldsymbol{L}$ are the two real lines of respective equations,

$$Ax - By = \mu,$$
$$Ax - By = \mu + \omega - 1,$$

with the same notations as in Definition 5.

It is clear that we can also define the points of the discrete line as the integer points that are inside the strip bounded by the two leaning lines and including these two lines. An *upper* (resp. *lower*) *leaning point* is a discrete point belonging to the upper (resp. lower) leaning line. Note that the leaning points are points of the discrete line. If we consider only a segment of a discrete line, we can define the upper and lower leaning points of minimum abscissa and denote them respectively by U_m and L_m. In the same way, we denote by U_M

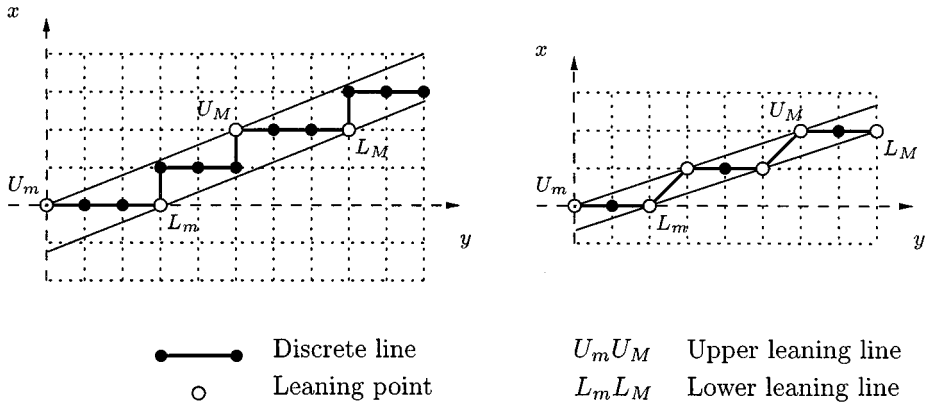


FIG. 12. The left part of the figure shows a segment of the 4-connected discrete line of characteristics ($A=2$, $B=5$, $\mu=0$, $\omega=7$). The right part shows a segment of the 8-connected discrete line of characteristics ($A=1$, $B=3$, $\mu=0$, $\omega=3$). In both cases the four leaning points of minimum and maximum abscissa U_m, L_m, U_M, and L_M are emphasized.

and L_M the upper and lower leaning points of maximum abscissa. Figure 12 is an illustration of the previous definitions. In the following, we omit the thickness of a discrete line insofar as it can be deduced from the connectivity of the line.

We now present a modified version of Debled's algorithm for incrementally recognizing an 8-connected discrete line segment [10]. Let S be an 8-connected discrete segment having characteristics (a, b, μ) with $0 \le a < b$ (S belongs to the first octant). The first point of the segment is chosen as the origin of the coordinates system. Let us denote by Q the last point of S. To incrementally solve the recognition problem we have to know in which cases the path S' equal to the segment S augmented by a point $P=(x_P, y_P)$ is also a discrete line segment and, if this is true, which are its characteristics (a', b', μ'). Debled's initial presentation of this test, designed to be incrementally applied, is based on updates of U_m and L_m. We have modified this test by also updating the positions of L_M and U_M. Why this adaptation is useful will be explained in the following.

First of all, as the segment S belongs to the first octant, S' can be a discrete segment only if $(x_P, y_P)=(x_Q+1, y_Q)$ or $(x_P, y_P)=(x_Q+1, y_Q+1)$, i.e. if there is an east move or a north-east move from Q to P. The four following cases are then possible. The path S' can be a line segment with the same slope as S, with a higher slope, or with a lower slope. It is also possible that S' is not a discrete line segment.

1 If the coordinates of P verify $\mu \le ax_P - by_P < \mu + b$ then S' is a discrete line segment with the **same characteristics** as S: $(a', b', \mu')=(a, b, \mu)$.
- 1a if $ax_P - by_P = \mu + b - 1$, the point P is a lower leaning point. It is, moreover, the new lower leaning point of maximum abscissa, so $L'_M = P$.
- 1b if $ax_P - by_P = \mu$, the point P is an upper leaning point. It is the new upper leaning point of maximum abscissa, so $U'_M = P$.

2 If the coordinates of P verify $ax_P - by_P = \mu - 1$, S' is a discrete line segment of **higher slope** than S. Its characteristics are $(a', b', \mu')=(y_P - y_{U_m}, x_P - x_{U_m}, a'x_P - b'y_P)$. The point P is the new upper learning point of maximum abscissa: $U'_M = P$. The leaning points L_M and U_m are unchanged. The lower leaning point of minimum abscissa is updated, so $L'_m = L_M$.

TABLE 1
Adding a Point $P=(x_P, y_P)$ at the Positive Extremity of a Discrete 8-Connected Line Segment

Case	$r=ax_P-by_P$	Slope	New characteristics $(a', b', \mu', U'_m, U'_M, L'_m, L'_M)$
1	$\mu<r<\mu+b-1$	$\rightarrow$	$(a, b, \mu, U_m, U_M, L_m, L_M)$
1a	$\mu+b-1$	$\rightarrow$	$(a, b, \mu, U_m, U_M, L_m, P)$
1b	μ	$\rightarrow$	$(a, b, \mu, U_m, P, L_m, L_M)$
2	$\mu-1$	$\nearrow$	$(y_P-y_{U_m}, x_P-x_{U_m}, ax_P-by_P, U_m, P, L_M, L_M)$
3	$\mu+b$	$\searrow$	$(y_P-y_{L_m}, x_P-x_{L_m}, ax_P-by_P-b+1, U_M, U_M, L_m, P)$
4	Else		S' is not a discrete line segment

Note. The new characteristics and leaning points of the augmented segment S' are given accordingly to the different values of $r=ax_P-by_P$.

3 If the coordinates of P verify $ax_P-by_P=\mu+b$, S' is a discrete line segment of **lower slope** than S. Its characteristics are $(a', b', \mu')=(y-y_{L_m}, x-x_{L_m}, ax-by-b+1)$. The point P is the new lower leaning point of maximum abscissa: $L'_M=P$. The leaning points U_M and L_m are unchanged. The upper leaning point of minimum abscissa is updated, so $U'_m=U_M$.

4 In all the other cases S' is not a discrete 8-connected line segment.

Table 1 sums up the updates of the characteristics and of the leaning points of the augmented segment S' according to the different values of $r=ax_P-by_P$. All the cases are illustrated in Fig. 13.

To translate this work into an algorithmic form, we have to describe the initial conditions. Let us consider the two first points of the discrete 8-connected path set in the first octant that we try to recognize as a line segment. The initial conditions of the algorithm come from the pair of points from which the line recognition is started. According to the hypothesis, these two first points can be separated either by an east step or by a north-east one. In the first case, $(a, b, \mu)=(0, 1, 0)$, $U_m=L_m=(0, 0)$, $U_M=L_M=(1, 0)$. In the second case $(a, b, \mu)=(1, 1, 0)$, $U_m=L_m=(0, 0)$, $U_M=L_M=(1, 1)$.

We have just seen a technique for recognizing a discrete 8-connected line segment. With this method, the segment to be recognized can be extended at only one of its extremities (the one of positive abscissa). The algorithm can, however, easily be extended to continue the segment by points located at both of its extremities. In fact, continuing a line segment of the first octant with a point of negative abscissa is symmetric to the previously studied case (adding a point of positive abscissa). The algorithm for adding a "negative point" P is summed up in Table 2 which gives the new characteristics and leaning points of the augmented segment S' according to the different values of $r=ax_P-by_P$.

We have just detailed the technique for recognizing an 8-connected discrete line segment. However, the goal of this paper is to process interpixel boundaries that are 4-connected discrete contours. We show here that the problem of recognizing a 4-connected discrete line segment is equivalent to recognizing an 8-connected line segment deduced from it.

PROPOSITION 1. *Let us consider $\mu\in\mathbb{Z}^2$ and $(a, b)\in\mathbb{N}^2$ such that $0\le a<b$. The discrete 4-connected line of characteristics $(a, b-a, \mu)$ and the discrete 8-connected line of characterictics (a, b, μ) have the same Freeman code.*

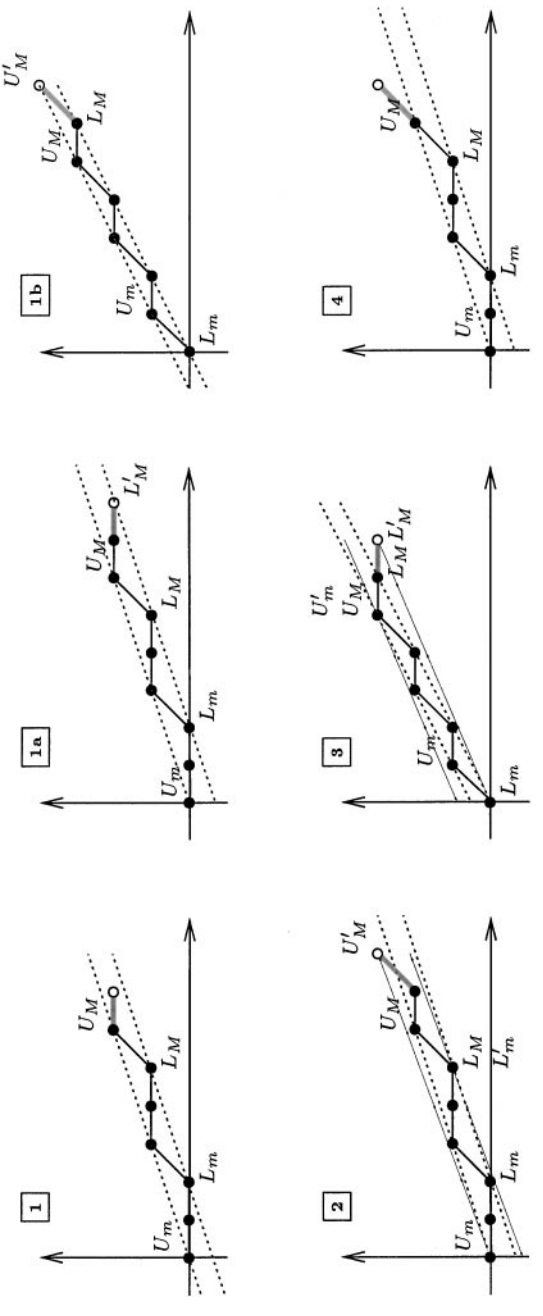


FIG. 13. Adding a point to a discrete 8-connected line segment. The added point is represented as a white point. The leaning lines of the initial segment are represented by dotted lines. The leaning lines of the augmented segment are represented by solid thin lines when they are different from the initial leaning lines. The new extreme leaning points are mentionned only if they differ from the initial ones.

TABLE 2

Adding a "Negative" Point $P = (x_P, y_P)$ to a Discrete 8-Connected Line Segment

Case	$r = ax_P - by_P$	Slope	New characteristics $(a', b', \mu', U'_m, U'_M, L'_m, L'_M)$
1	$\mu < r < \mu + b - 1$	$\rightarrow$	$(a, b, \mu, U_m, U_M, L_m, L_M)$
1a	$\mu + b - 1$	$\rightarrow$	$(a, b, \mu, U_m, U_M, P, L_m)$
1b	μ	$\rightarrow$	$(a, b, \mu, P, U_m, L_m, L_M)$
2	$\mu - 1$	$\searrow$	$(y_{U_M} - y_P, x_{U_M} - x_P, ax_P - by_P, P, U_m, L_m, L_m)$
3	$\mu + b$	$\nearrow$	$(y_{L_M} - y_P, x_{L_M} - x_P, ax_P - by_P - b + 1, U_m, U_m, P, L_M)$
4	Else		S' is not a discrete line segment

Note. The origin of the coordinates is a point of the initial segment. The point P added at the extremity of the segment has a negative abscissa.

Proof. Let $\boldsymbol{L}_4$ be a 4-connected line in the first quadrant. We associate with each point (x, y) of $\boldsymbol{L}_4$ the parameter $t = x + y$. It allows us to define a bijection between $\boldsymbol{L}_4$ and $\mathbb{Z}$ (sequential numbering of the line's points).

As it belongs to the first quadrant, $\boldsymbol{L}_4$ is only composed of east and north steps. Let us consider a point M of $\boldsymbol{L}_4$ and N the point following M along $\boldsymbol{L}_4$. The coordinates of N are defined by

$$(x_N, y_N) = (x_M + 1, y_M) \quad \text{or} \quad (x_N, y_N) = (x_M, y_M + 1).$$

The parameter associated with the point N is in both cases $t_N = t_M + 1$. The proposed parameterization of $\boldsymbol{L}_4$ is thus a curvilinear abscissa along $\boldsymbol{L}_4$.

In the following, we denote by $(x(t), y(t))$ the coordinates of the point of parameter t.

Let us consider the sequence of points $\boldsymbol{L}_y = (t, y(t)), t \in \mathbb{Z}$. This sequence corresponds to the variation of the y coordinate along $\boldsymbol{L}_4$. Figure 14 gives an example of the variation of the y coordinate along a 4-connected line according to the parameterization just described.

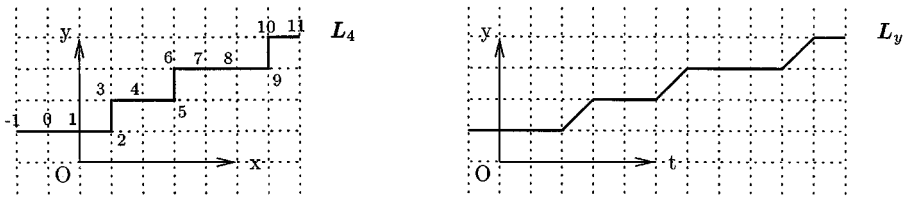
As $\boldsymbol{L}_4$ is a 4-connected discrete line of the first quadrant, we have for each $t \in \mathbb{Z}$:

$$y(t + 1) = y(t) \quad \text{or} \quad y(t + 1) = y(t) + 1.$$

It implies that $\boldsymbol{L}_y$ is an 8-connected discrete path and its Freeman code is composed of 0s and 1s. A horizontal step on $\boldsymbol{L}_4$ gives a horizontal step on $\boldsymbol{L}_y$. In the same way, a vertical step on $\boldsymbol{L}_4$ gives a diagonal step on $\boldsymbol{L}_y$. These two paths are thus represented by the same Freeman code respectively considered as 4-connected or 8-connected.

Let $(a, b - a, \mu)$ be the characteristics of $\boldsymbol{L}_4$. Its equation is

$$\mu \leq ax - (b - a)y < \mu + a + (b - a);$$



Freeman's code of $\boldsymbol{L}_4$: ...000100100010... y-variation along $\boldsymbol{L}_4$: ...000100100010...

FIG. 14. The relation between 4-connected and 8-connected discrete lines.

as $x + y = t$, we obtain

$$\mu \leq a(t - y) - (b - a)y < \mu + b,$$

which is equivalent to

$$\mu \leq at - by < \mu + b.$$

$\boldsymbol{L}_y$ is an 8-connected discrete path and all its points verify the previous equation. $\boldsymbol{L}_y$ is, thus, the 8-connected discrete line of characteristics (a, b, μ). ■

Proposition 1 states that the recognition of a 4-connected discrete line segment is equivalent to the recognition of an 8-connected segment. More precisely, given the Freeman code of a 4-connected line segment of the first quadrant, we can find the characteristics (a, b, μ) of $\boldsymbol{L}_y$ (y-variation) by using the algorithm for the 8-connected case. The characteristics of the 4-connected segment are then $(a, b - a, \mu)$.

4.2. *Discrete Tangent at a Boundary Point*

In the continuous case one can consider a small enough neighborhood of a curve point as a linear segment being the support of the tangent at this point. By analogy with the continuous case, we choose to define the discrete tangent at a point P of an interpixel boundary $\mathbf{\Pi}$ as the longest part of $\mathbf{\Pi}$ centered on P that is a discrete line segment (the definitions and properties of the previous section can obviously be transposed in the half-integer grid).

From an algorithmic point of view, searching the discrete tangent at P corresponds to searching a 4-connected segment centered on P. This section details our algorithmic definition of the tangent at a boundary point.

We consider an interpixel boundary $\mathbf{\Pi}$ defined by its Freeman code and oriented counterclockwise. As $\mathbf{\Pi}$ is closed, we can consider the next point of any point of $\mathbf{\Pi}$. We do not need to define an absolute position of $\mathbf{\Pi}$ in the plane as a tangent is defined relative to a point.

Let us consider a point P of $\mathbf{\Pi}$ that we will call the *current point*. The points following P (resp. preceding P) along $\mathbf{\Pi}$ are numbered increasing from 1 (resp. decreasing from -1).

Let P_{i_f} ($i_f \geq 0$) be the first point at which a change in direction occurs. In other words, if we denote by code (P_i) the Freeman code of the elementary step of origin P_i, we have

$$\text{code}(P_{i_f - 1}) \neq \text{code}(P_{i_f});\ \forall_j,\ 0 \leq j < i_f, \quad \text{code}(P_{j-1}) = \text{code}(P_j).$$

Two different and consecutive codes on a 4-connected path define a unique quadrant (see Fig. 2). It implies that code$(P_{i_f - 1})$ and code(P_{i_f}) define a quadrant that we call the forward quadrant and denote by Q_f. In the same way, we define P_{i_b} ($i_b \leq 0$) as the first point preceding P, where an orientation change occurs. We denote by Q_b (for backward) the corresponding quadrant.

Let us now denote by Q the nearest quadrant of P, that is:

- if $i_f < -i_b$, $Q = Q_f$,
- if $i_f > -i_b$, $Q = Q_b$,
- if $i_f = -i_b$ and if $Q_f = Q_b$, $Q = Q_f$.

This does not describe the case where $i_f = -i_b$, but $Q_f \neq Q_b$. To avoid an arbitrary choice, we define in that case the discrete tangent as a vertical or horizontal discrete segment.

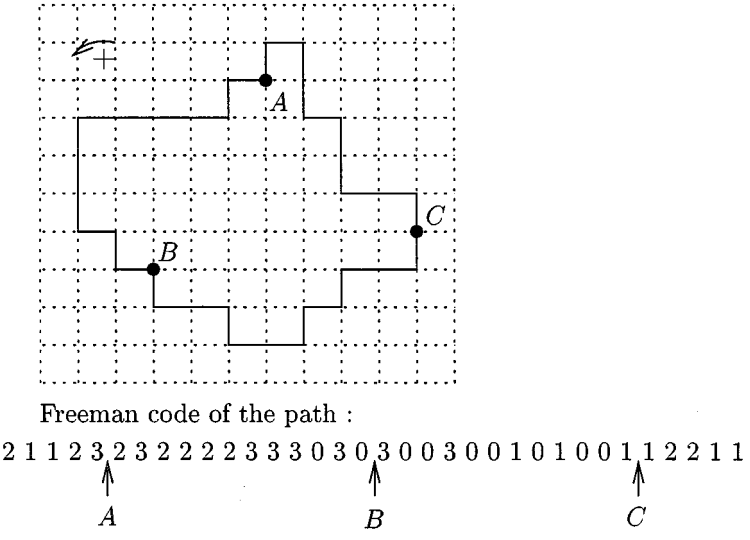


FIG. 15. Discrete interpixel boundary.

The point $P = C$ of Fig. 15 is an example of that case: $i_f = -i_b = 1$, $Q_f = 2$, and $Q_b = 1$. The discrete tangent is defined as a vertical discrete segment of length 2 (see Fig. 18).

In the other cases, we consider the part of the boundary around P composed of the two codes defining the quadrant Q. In the case where $P = B$, we have $i_f = i_b = 0$ and $Q = 4$. The selected part of the path is 333030300300 (see Fig. 16). We replace each code element c by c *modulo* 2 to transpose the problem to the first quadrant.

We have shown that the recognition of a 4-connected segment of the first quadrant is equivalent to the recognition of an 8-connected segment of the first octant. According to that result, we consider the obtained code composed of 0s and 1s as an 8-connected code defined around the current point P.

We then search the longest discrete segment centered on P. At each step of the process, we add a pair of points at the extremities of the segment. If the pair of points does not allow us to continue the discrete line segment, neither of them is added. It would be possible to add

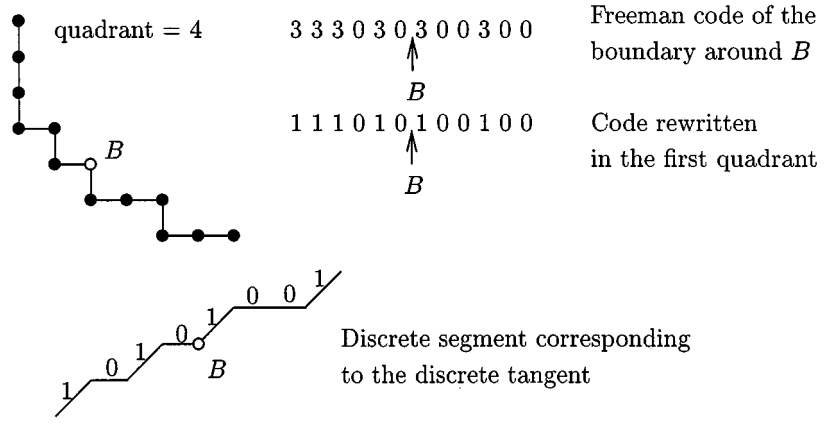


FIG. 16. Discrete tangent search at B.

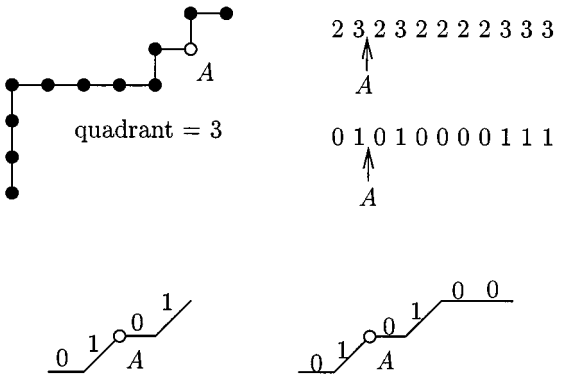


FIG. 17. Discrete tangent search at A.

to the current discrete segment one of the two points, but it would then be necessary to arbitrarily choose between the two candidates. Since there is not an objective criterion to perform this choice, we prefer to discard both points.

For example, if $P = B$, the point of number -5 does not allow us to continue the segment. The discrete tangent at B will have a length of 8 (see Fig. 16 and Fig. 18).

The case we described is the general case, where the adding of new points is performed symmetrically. There is a particular case in which the adding is not symmetric, when reaching a change of quadrant. In that case, the recognition stops at the change, but it continues at the other side, as illustrated below.

Consider the case of point $P = A$. In that case we have $i_f = i_b = 0$. The quadrant of the discrete tangent is thus the quadrant of number 3 (directions 3 and 2 around A). The selected part of the path is 23232222333. We deduce from it a code composed of 0s and 1s. The recognition of the longest discrete segment around A leads to the segment of code 0101. We are stopped on the part of the code preceding A by a change of quadrant. In that case, the search is continued at the other extremity of the segment (it involves no arbitrary choice). In this example, we can add two more points, which leads to a discrete tangent of length 6 (see Fig. 17 and Fig. 18).

Figure 18 sums up the discrete tangents found at the points A, B, and C.

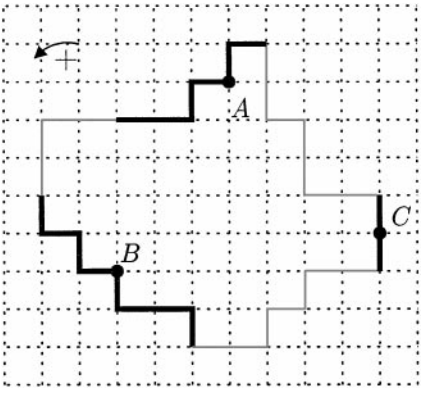


FIG. 18. Discrete tangents.

The algorithm for extracting the discrete tangent at a point of an interpixel boundary is the following.

- Search the quadrant Q of the tangent and extract the part of the boundary around P that corresponds to Q.
- Transpose the problem to the first quadrant; each code element c is replaced by c *modulo* 2.
- Recognize in this code the longest 8-connected segment of the first octant around P. We add systematically a pair of points around P. If the search stops at one side of P due to the end of the selected code portion, the search continues at the other side. This stage ends when we cannot add any new point.

Although these three stages are presented here independently, they are mixed in practice to avoid traversing the path many times. The resulting discrete tangents are independant of the way in which the path is traversed.

The characteristics (a, b, μ) of the 8-connected segment of the first octant recognized by the algorithm allows us to compute the characteristics of the 4-connected segment corresponding to the discrete tangent at the point P. Remember that these characteristics are computed in the coordinate system of origin P. The search for the discrete tangent at a point P of an interpixel boundary is based on the y-variation around P (Section 4.1). We obtain with this method the characteristics (a, b, μ) of an 8-connected line of the first octant. We deduce from it the characteristics of the discrete tangent. For quadrants 1, 2, 3, and 4, the characteristics of the discrete tangent are respectively $(a, b-a, \mu)$, $(a, a-b, -\mu-b+1)$, $(-a, a-b, \mu)$, and $(-a, b-a, -\mu-b+1)$.

5. CONSTRUCTION OF TANGENT-DRIVEN EUCLIDEAN PATHS

We consider now a canonical continuous approximation of a discrete line and we combine this notion with the notion of a discrete tangent in order to define an approximation of the underlying real tangent at a point of a discrete curve. We have already considered two real lines directly linked to a discrete line, the leaning lines (Definition 6). These two lines define a strip bounding the points of the discrete line. We can now define *the centered line* as the line located in the middle of the two leaning lines.

DEFINITION 7. The *centered line* of a discrete line $\boldsymbol{L}$ is the real line located in the middle of the two leaning lines of $\boldsymbol{L}$. Its equation is

$$Ax - By = \mu + \frac{\omega - 1}{2}$$

with the same notations as in Definition 5.

This definition is illustrated in Fig. 19 for the 4-connected case.

PROPOSITION 2. *A parametric equation of the centered line associated with the 4-connected line of the first quadrant of characteristics* $(a, b-a, \mu)$ *is*

$$\left.\begin{aligned} x &= \tfrac{b-a}{b}t + \alpha \\ y &= \tfrac{a}{b}t - \alpha \end{aligned}\right\} \quad \text{with } \alpha = \frac{-1+b+2\mu}{2b}.$$

This parametric equation is easily deduced from the cartesian equation of the line (Definition 7), combined with the parameterization $t = x + y$.

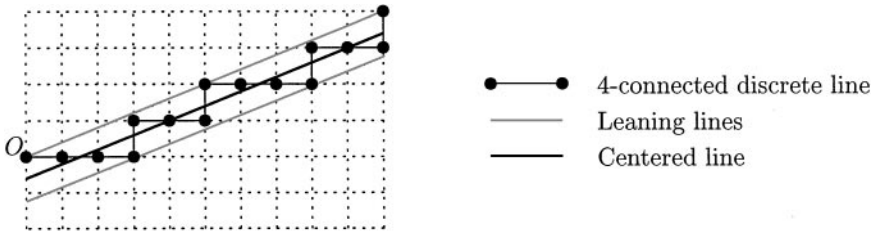


FIG. 19. Centered line of a discrete 4-connected line.

If we consider a discrete line of the half-integer plane as the boundary of two adjacent regions, the centered line separates equitably the two regions. It can thus intuitively be considered as the "real" boundary between the two regions. We therefore define the *real tangent* at a boundary point as the centered line of the discrete tangent.

We have seen that recognition of the discrete tangent at a point P of an interpixel boundary is based on the y-variation around P. We obtain with this method the characteristics (a, b, μ) of a discrete 8-connected line of the first octant. We deduce from it the characteristics of the discrete tangent and, thus, the equation of the real tangent according to Definition 7. Figure 20 shows the approximation of the real tangent at points A, B, and C of Fig. 15. The real tangent is considered to be a good local approximation of the underlying real boundary. We thus choose the Euclidean point associated with the boundary point on the real tangent. Which point is chosen is detailed in the following.

LEMMA 1. *Let* $\boldsymbol{L}$ *be an 8-connected discrete line located in the first octant and of characteristics* (a, b, μ). *Let us suppose that the origin of the coordinates system is a point of* $\boldsymbol{L}$. *We have*

$$\left| \frac{-1 + b + 2\mu}{2b} \right| < \frac{1}{2}.$$

Proof. As $\boldsymbol{L}$ belongs to the first octant, its points verify:

$$\mu \leq ax - by < \mu + b \quad \text{with } 0 \leq a < b.$$

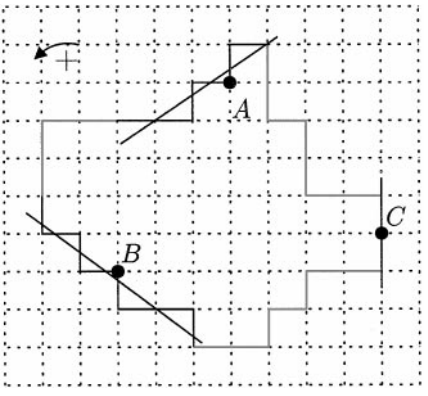


FIG. 20. Real tangent approximation at a point of an interpixel boundary.

Moreover, the point (0, 0) belongs to $\boldsymbol{L}$ which gives

$$\mu \leq 0 < \mu + b.$$

It implies the sequence of equivalent inequalities:

$$\begin{aligned}
\mu &\leq 0 \\
2\mu &\leq 0 \\
2\mu &< 1 \quad (\mu \text{ has an integer value}) \\
-1 + 2\mu &< 0 \\
-1 + b + 2\mu &< b \\
\frac{-1 + b + 2\mu}{2b} &< \frac{1}{2} \quad (b \text{ is strictly positive})
\end{aligned}$$

and

$$\begin{aligned}
\mu + b &> 0 \\
\mu + b &\geq 1 \quad (b \text{ and } \mu \text{ have integer values}) \\
2(\mu + b) &\geq 2 \\
2(\mu + b) &> 1 \\
-1 + b + 2\mu &> -b \\
\frac{-1 + b + 2\mu}{2b} &> -\frac{1}{2} \quad (b \text{ is strictly positive});
\end{aligned}$$

thus we have

$$\left| \frac{-1 + b + 2\mu}{2b} \right| < \frac{1}{2}. \quad \blacksquare$$

Let us consider a point P of an interpixel boundary. We suppose here that the discrete tangent at P belongs to the first quadrant. We have just seen that the parametric equation of the real tangent in the coordinate system of origin P is

$$\left.\begin{aligned} x(t) &= \tfrac{b-a}{b}t + \alpha \\ y(t) &= \tfrac{a}{b}t - \alpha \end{aligned}\right\} \quad \text{with } \alpha = \frac{-1 + b + 2\mu}{2b}.$$

This equation comes from a parameterization of the discrete tangent that associates with each point the sum of its two coordinates. The real point $p = (x(t_P), y(t_P))$ is called the *canonical projection* of P on the real tangent. As P is the origin of the coordinates system, $t_P = x_P + y_P = 0$. The canonical projection of P on its tangent is thus the point $p = (\alpha, -\alpha)$. We can choose this point as the Euclidean point associated with P (Lemma 1 ensures that p verifies the constraint of Definition 3). Figure 21 illustrates the construction of p. The displayed discrete path corresponds to the discrete tangent at P. The point p is located at the intersection of the real tangent and of the line of equation $y = -x$.

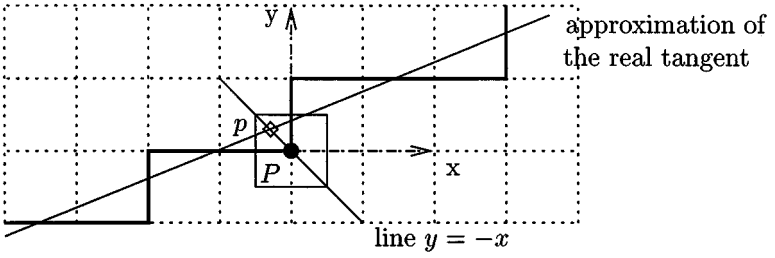


FIG. 21. Euclidean point associated with a boundary point. The 4-connected discrete path in black corresponds to the discrete tangent at P.

Figure 22 shows the symmetric configurations of the example of Fig. 21 in quadrants 2, 3, and 4. The recognition algorithm of the discrete tangent at P leads in the four cases to the recognition of the same 8-connected line segment of the first octant and, thus, to the computation of the same value of α. In each case the coordinates of p are deduced by symmetry from the coordinates of the point $(\alpha, -\alpha)$. For the respective quadrants 1, 2, 3, and 4, the coordinates of p are $(\alpha, -\alpha)$, $(-\alpha, -\alpha)$, $(-\alpha, \alpha)$, and (α, α).

Figures 23, 24, and 25 show the smoothing and reconstruction qualities of the tangent-driven Euclidean paths. On the left part of each figure are displayed the discrete boundary (solid black line) and the associated Euclidean path (black points). On the right part, the Euclidean path is displayed as a polygon linking the Euclidean points.

Figure 23 shows the Euclidean path associated with the interpixel boundary of a square. The square has been rotated to process a visually irregular discrete path. It can be shown that the Euclidean points associated (according to the tangent-driven method) with a 4-connected discrete line are aligned on its centered line. The construction method performs thus an implicit recognition of the discrete segment. This allows us to approximately reconstruct a polygon from its discrete representation. Moreover, the junction between two consecutive segments of the polygon is smoothed.

Figure 24 shows the Euclidean path associated with the interpixel boundary of a disc. The initial circle is not exactly restored because of the loss of information involved by the

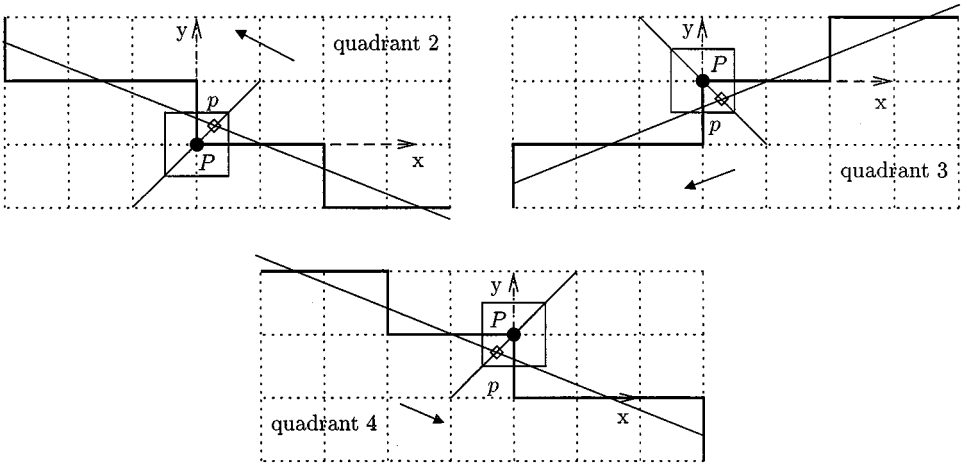


FIG. 22. Symmetric configurations in the computation of a Euclidean point associated with a boundary point. The arrow indicates the boundary orientation.

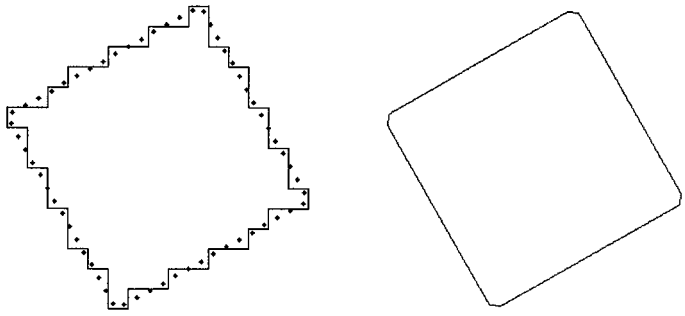


FIG. 23. Tangent-driven Euclidean path. The initial discrete path is the interpixel boundary of the square of side 10 pixels rotated 30°.

digitization of the disc. However, the Euclidean path allows us to recover the circular aspect of the initial object.

Finally, the example of Fig. 25 illustrates the smoothing of an interpixel boundary coming from an image segmentation.

The **time complexity** of the tangent-driven Euclidean path computation is the same as the complexity of the tangent recognition algorithm. As a matter of fact the Euclidean point associated with a discrete point is directly deduced from the characteristics of the discrete tangent at this point (α computation).

The time complexity of the construction algorithm is $O(l \times n)$, where l is the average length of the discrete line segments contained in the discrete contour and n is the number of points of the discrete contour. The computation time depends thus on the shape of the contour. The more a discrete curve is rectilinear the more the computation of the associated Euclidean path is costly. However, for a given image size the shape of the discrete curve to be processed becomes more irregular when its length increases. In other words, the value of l tends to decrease when the value of n increases. Nevertheless, experimental results give us running times suitable for an interactive use of this method. Some of these times are presented in Table 3. The processed contours are classified according to their number of points. Globally the computation time increases with the number of contour points.

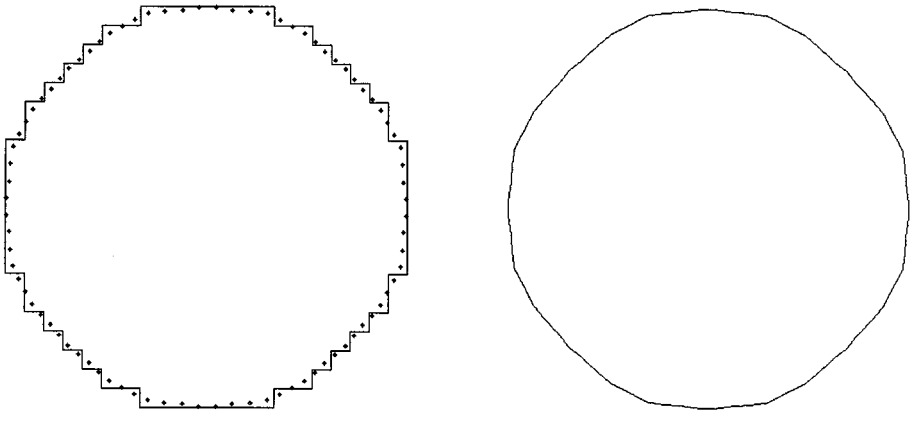


FIG. 24. Tangent-driven Euclidean path. The initial discrete path is the interpixel boundary of the discrete disc of radius 10 pixels.

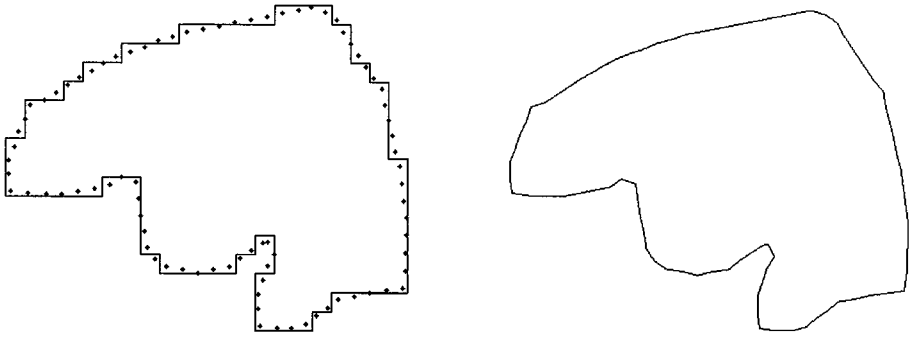


FIG. 25. Tangent-driven Euclidean path. The initial discrete path is the interpixel boundary of a region coming from an image segmentation.

Contours like the house boundary or the rectangle boundary need more time than the others relative to their number of points because they are polygonal contours.

6. APPLICATIONS

As pointed out in the Introduction of this paper, tangent-driven Euclidean paths were designed in order to smooth the boundaries of digitized regions. We are now going to present several applications we have developed from this representation.

The construction of Euclidean paths can be extended to a set of intersecting contours, such as the set of interpixel boundaries resulting from a segmentation of a digital image. It is, for instance, possible to independently compute the Euclidean path associated with each region's boundary. Consider two adjacent regions R_1 and R_2 which respective contours C_1 and C_2 and let P be a point belonging both to C_1 and C_2. Now, let E_1 be the Euclidean point associated to P according to C_1 and let E_2 be the Euclidean point associated to P according to C_2. Since the construction of E_1 and E_2 can involve different sets of contour points, it is possible to have $E_1 \neq E_2$ (note that it does not contradict constraint 2—see the Introduction). In that case E_1 and E_2 can be replaced by the average of E_1 and E_2. It is

TABLE 3
Computational Times of the Construction of the Tangent-Driven Euclidean Paths Associated with Different Types of Interpixel Boundaries

Region	Contour size	Computation time (s)
Disc of radius 20	164	0.01
Rectangle of side 200 rotated at 25°	1068	0.33
Flower in close-up	1352	0.11
Lena face	1804	0.26
Irregular shape segmented from a landscape	9352	0.71
House	972	0.18

Note. They are computed on a Iris Indigo2—150 MHz processor. The size of each contour is its number of points. The first two lines are two geometrical shapes: a disc which corresponds to the best configuration for the algorithm and a rectangle which corresponds to the worst one. The other examples come from natural images: a flower, the well-known Lena image, and a part of a landscape the contour of which does not countain long linear parts, and a house the contour of which has relatively long linear segments.

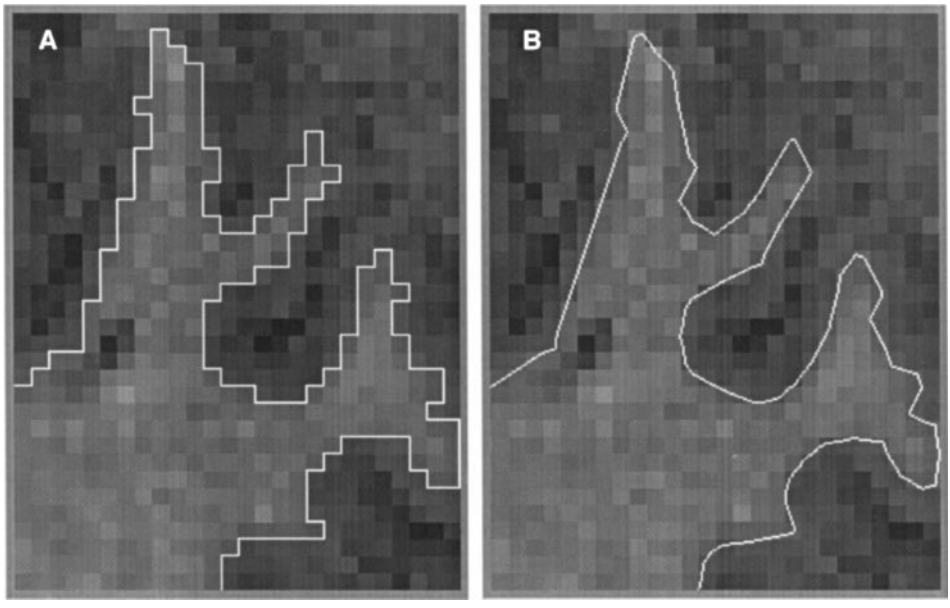


FIG. 26. An enlargement of a rectangular part of a brain MRI is shown on the left picture. An interpixel discrete boundary resulting from a segmentation is superimposed on the picture. The same enlarged part of the brain MRI, on which is superposed the Euclidean path associated with the discrete boundary, is displayed in the right image.

straightforward to verify that the resulting point is also a Euclidean associated with P. Such a construction preserve the topology of the set of contours [5].

Since tangent-driven Euclidean paths provide a smoothed boundary of a discrete region, such a smoothed boundary can be enlarged without staircase effects. Moreover, since each point of the smoothed boundary falls into the unit cell of the associated interpixel point, the enlarged boundary may be superimposed on the enlarged discrete image without distortion. An example of such an enlargement is shown in Fig. 26 on an area of size 26×34. On the left image is displayed the interpixel boundary of a segmented region and on the right, the associated enlarged Euclidean path. Despite the enlargement, the smoothed boundary belongs to the strip made of the pixels adjacent to the interpixel boundary. Thus, the enlarged smoothed boundary can be used to manipulate the discrete boundary to within one pixel.

Even when displayed at scale one, tangent-driven Euclidean paths improve the visual aspect of discrete boundaries. Figure 27 shows a region segmentation of a brain MRI. The displayed boundaries are interpixel boundaries drawn in the integer plane. Thus, the interpixel boundaries have to be translated by $(\frac{1}{2}, \frac{1}{2})$ in order to be displayed.

A better visual result is obtained by digitizing the tangent-driven Euclidean path associated with the interpixel boundary. As a matter of fact, the Euclidean point associated with an interpixel point P may be located near one or the other of the four neighboring integer points of P. Thus, the displayed point may be any of these four neighbors and not only the translation of P. As it can be seen in Fig. 28 the visual aspect of boundaries is improved, even when displayed at scale one.

It is also possible to compute a tangent-driven Euclidean path from an 8-connected discrete path. This case is very similar to the 4-connected one. It allows us to process 8-connected contours (pixel boundaries) as well as 4-connected contours (interpixel

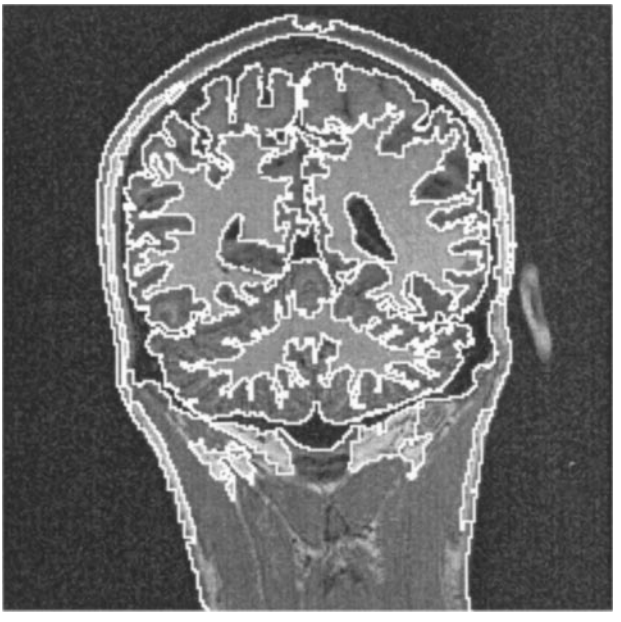


FIG. 27. The interpixel boundaries displayed in this image have been translated by $(\frac{1}{2}, \frac{1}{2})$ and then superimposed on the original image.

boundaries). The following applications are based on 8-connected contours. In each case the method could also be applied to 4-connected contours.

Since a Euclidean path is based on a tangent approximation, it naturally provides a method of normal approximation of a discrete curve. Figure 29 shows two examples of normal approximation. The first example is the contour of a digitized disc of radius 20

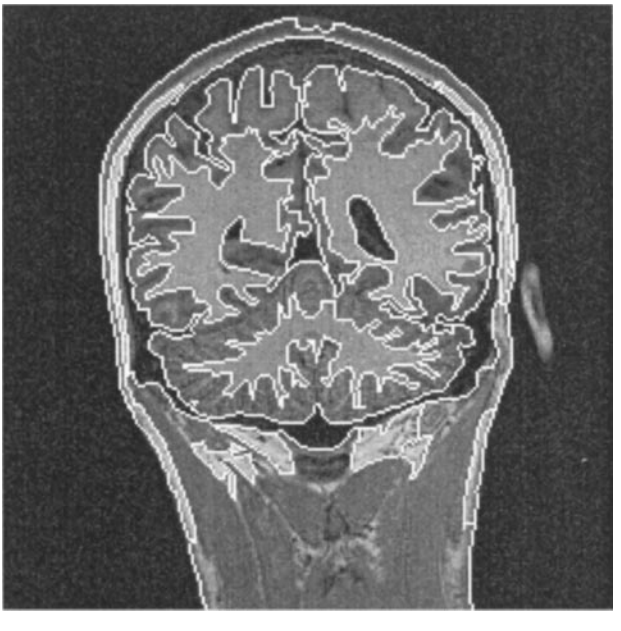


FIG. 28. The interpixel boundaries displayed in this image result from digitizing the associated Euclidean paths.

FIG. 29. The left example shows some normal vectors obtained with the method of tangent approximation at some points of a digitized circle. In this theoretical example the approximation is particularly accurate. The right example shows normal directions of the contour of a segmented region.

pixels and the second one is the contour of a human head resulting from the segmentation of a MRI image. Tangent-driven Euclidean paths have also been used to estimate the length and the curvature of a discrete boundary. Both for normal approximation and for parameters estimation, tangent-driven Euclidean paths improve the usual methods [22].

It is also possible to derive a method of region antialiasing from Euclidean paths. The problem of region antialiasing appears when superimposing a discrete region on a background image. The superimposed region may, for instance, be a part of an image to be combined with the background image. This processing is known as *image compositing* [2]. The simply way to perform image compositing consists in replacing each pixel of the background image by the related pixel of the region. Of course this trivial method produces staircase effects on the boundary. Thus, the boundary of the region is anti-aliased by blending the color of the removed pixel with the color of the region pixel. When the superimposed region is a real region (for instance, a region produced by image synthesis), it is possible to compute the blending coefficients at each pixel boundary from the ratio between the area of the background pixel covered by the superimposed region and the area of the pixel [16, 2]. On the other hand, when the superimposed region is a discrete region (for instance, a segmented region of a digitized image), the area of the background pixel covered by the superimposed region is undefined.

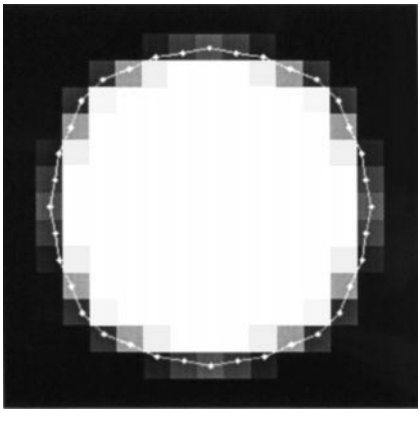


FIG. 30. This is an example of superimposition of a discrete while disc of radius 6 on a black background. The contour is the Euclidean path built from the disc boundary.

Several methods have been proposed to derive blending coefficients from a discrete image [3, 20], but these methods are restricted to polygonal regions. We have proposed using tangent-driven Euclidean paths, based on pixel contours in order to compute blending coefficients, as is done with Euclidean regions. We have shown that in theoretical cases it is possible to retrieve the anti-aliasing of a Euclidean region and that in real cases this approach gives better results than the usual ones [4]. An example of an anti-aliased discrete disc is given in Fig. 30.

Euclidean paths can be used to perform geometrical processing of discrete uniform regions. A typical example is the processing of digitized characters of a font. For instance, a slanted character may be produced by shearing an upright one. Several examples illustrating the utility of Euclidean paths for such processing have been developed in other works [21, 23].

7. CONCLUSION

We have presented a new representation of discrete interpixel boundaries. The basic idea of the Euclidean paths is to associate each boundary point with a real point located in its near neighborhood. This model preserves all the information of the initial discrete boundary and, thus, allows us to work to within a pixel.

We have developed a construction method of smooth Euclidean paths based on tangent recognition at each point of the processed discrete boundary. With this method, we consider for each boundary point a neighborhood adapted to the local configuration of the discrete boundary. This allows us to preserve the characteristic points (vertices of a polygon, for example) and also to correctly smooth the rectilinear or rounded parts of the boundary.

The construction of tangent-driven Euclidean paths does not need any arbitrary choice; it gives the same result for any orientation of the boundary and for any choice of the starting point. This technique does not need, either, any interpretation of the boundary, unlike approaches such as polygonal approximation.

Tangent-driven Euclidean paths have many applications: visualization at different scales and possibly with geometric transforms of interpixel boundaries, anti-aliasing of the boundary of discrete regions in the domain of image compositing, and computation of some geometric characteristics of the boundaries.

Color information is not used at present to construct a Euclidean path: our technique is only based on the geometry of the discrete boundary. If the boundary comes from an image segmentation, it would be interesting to take into account the colorimetric information of the image near the boundary. In particular, we think that it is possible to make the normal computation at a boundary point more accurate by using at the same time the gradiant direction of the image and the geometrical configuration of the boundary.

We are currently working on the extension of this work to three dimensions. The analogy is obvious between pixels and voxels (cubic volume elements); a discrete 3D boundary is composed of the outside faces of the voxels located at the boundary of the considered object, and the Euclidean path is analogous to a *Euclidean grid*. However, the extension of our method to three dimensions is not straightforward. For now we investigate local rewriting analogous to the one illustrated in Fig. 7.

REFERENCES

1. E. Ahronovitz, J. P. Aubert, and C. Fiorio, The star-topology: A topology for image analysis, in *5th Discrete Geometry for Computer Imagery, Proceedings, September 1995*, pp. 107–116.

2. J. F. Blinn, Compositing, part 1: Theory, *IEEE Comput. Graphics Appl.* September 1994, 83–87.
3. J. Bloomenthal, Edge inference with application to antialiasing, in *ACM Computer Graphics Proc. SIGGRAPH, 1983*. Vol. 17, pp. 157–162.
4. J.-P. Braquelaire and A. Vialard, A new antialiasing approach for image compositing, *Visual Comput.* **13**(5), 1997, 218–227.
5. J. P. Braquelaire, L. Brun, and A. Vialard, Inter-pixel euclidean paths for image analysis, in *Discrete Geometry for Computer Imagery*, Lecture Notes in Computer Science, Vol. 1176, pp. 193–204, Springer-Verlag, New York/Berlin, 1996.
6. J. P. Braquelaire and J. P. Domenger, Intersection of discrete contours, in *Proc. of Compugraphics'91, 1991*. pp. 14–23.
7. J. P. Braquelaire and J. P. Domenger, Representation of segmented images with discrete geometric maps, *Image Vision Comput.*, to appear.
8. C. R. Brice and C. L. Fennema, Scene analysis using regions, *Artif. Intell.* **1**, 1970, 205–226.
9. J. M. Chassery and A. Montanvert, *Géométrie discrète en analyse d'images*, Hermès, Paris, 1991.
10. I. Debled-Renesson and J. P. Reveilles, A linear algorithm for segmentation of discrete curves, *Int. J. Pattern Recognit. Artif. Intell.* **9**(6), December 1995.
11. J.-P. Domenger, *Conception et implémentation du noyau graphique d'un environnement 2D$\frac{1}{2}$ d'édition d'images discrètes*, Ph.D. thesis, Univ. Bordeaux I, April 1992.
12. L. Dorst and A. W. Smeulders, Decomposition of discrete curves into piecewise segments in linear time, *Contemp. Math.* **119**, 1991, 169–195.
13. H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* **10**(2), 1961, 260–268.
14. T. Y. Kong and A. Rosenfeld, Digital topology: Introduction and survey, *Comput. Vision Graphics Image Process.* **48**, 1989, 357–39.
15. V. A. Kovalevsky, Finite topology as applied to image analysis, *Comput. Vision Graphics Image Process.* **46**, 1989, 141–161.
16. T. Porter and T. Duff, Compositing digital images, In *SIGGRAPH 84, 1984*, pp. 253–259.
17. J. P. Reveilles, Géométrie discrète, calcul en nombres entiers et algorithmique, Thèse d'état, Université Louis Pasteur, Strasbourg, 1991.
18. L. Shao and H. Zhou, Curve fitting with bézier cubics, *Graphical Models and Image Processing* **58**(3), May 1996, 223–232.
19. Y. M. Sharaiha and N. Christofides, An optimal algorithm for the straight segment approximation of digital arcs, *Comput. Vision Graphics Image Process.* **55**(5), 1993, 397–407.
20. C. van Overveld. Application of morphological filter to tackle discretisation artefacts, *Visual Computer* **8**, 1992, 217–232.
21. A. Vialard, *Chemins Euclidiens: un Modèle de Représentation des Contours Discrets*, Ph.D. thesis, Université Bordeaux I, 1996.
22. A. Vialard, Geometrical parameters extraction from discrete paths, in *Discrete Geometry for Computer Imagery*, Lecture Notes in Computer Science, Vol. 1176, pp. 24–35, Springer New York/Berlin/Verlag, 1996.
23. A. Vialard, Euclidean paths for representing and transforming scanned characters, in *Proceedings of GREC'97*, Second IAPR Workshop on Graphics Recognition, 1997. [To be published in Lecture Notes in Computer Science, Springer Verlag.]