DTU Library

Digital squares

Forchhammer, Søren; Kim, Chul E

Published in:
9th International Conference on Pattern Recognition,

Link to article, DOI:
10.1109/ICPR.1988.28324

Publication date:
1988

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Forchhammer, S., & Kim, C. E. (1988). Digital squares. In 9th International Conference on Pattern Recognition, (Vol. Volumen 2, pp. 672-674). IEEE. DOI: 10.1109/ICPR.1988.28324

DIGITAL SQUARES

Søren Forchhammer and Chul E. Kim

Institute of Telecommunication, 343,
Technical University of Denmark,
DK 2800 Lyngby, Denmark

Computer Science Department
Washington State University
Pullman, Washington 99164-1210

Abstract

Digital squares are defined and their geometric properties characterized. A linear time algorithm is presented that given a convex digital region, determines whether or not it is a digital square. The algorithm also determines the range of the values of the parameter set of its preimages.

1. Introduction

This paper extends the results on digital rectangles in [4] to digital squares and investigates the range of parameter values of the preimages of a digital square. To analyze these problems we transform the boundary of a digital region into parameter space of slope and y-intercept.

2. Preimages of digital line segments

Suppose that $Q = \{d_0, d_1, ..., d_n\}$, where $d_j = (x_j, y_j)$, is a digital line segment and assume without loss of generality that $x_{j+1} = x_j + 1$. (Refer to [4] for definitions). Let $H(Q) = \{v_0, v_1, ..., v_m\}$ be the convex hull of Q with its vertices listed in counterclockwise order. Let $v_j v_{j+1}$ be the longest edge of $H(Q)$ and assume that the points of Q lie on and above its extension. Let f be the line segment which is an extension of $v_j v_{j+1}$ to span Q so that $Q = I(f)$, the digital image of f. Such a line segment is called a preimage of Q. Obtain line segment f' by a parallel translation of f downward until f' contains points of $\neg Q$ with the least vertical distance from f. Since v_j and v_{j+1} are the leftmost and rightmost digital points on f, we also denote them by v_ℓ and v_r, respectively. Now let u_ℓ and u_r denote the leftmost and rightmost digital points on f', where u_ℓ and u_r coincide when f' has only one digital point. Points u_ℓ, u_r, v_ℓ and v_r, are called the limiting points of digital line segment Q.

Given a digital arc $Q = \{d_0, d_1, ... d_n\}$, it can be decided in $O(n)$ time whether or not Q is a digital line segment, and if it is, then its limiting points v_ℓ, v_r, u_ℓ are also computable in linear time [1].

3. Digital edges and corner points

A convex digital region R is a digital square if there is a square whose digital image is R, and the boundary B of R may also be called a digital square if R is. (See [3] for definition of digital convexity).

Let $Q = \{d_0, d_1, ... d_n\}$ be a digital line segment. Then its chain $C_Q = (c_1, c_2, ... c_n)$ consists of at most two distinct chain links. We say that Q lies in the first octant if its chain C_Q consists of two distinct chain links (1,0) and (1,1). If C_Q consists of chain link (1,1), then Q may be considered to lie in either the first or second octant. A digital line segment lying in other octants is similarly defined.

A line segment uv from $u = (x_u, y_u)$ to $v = (x_v, y_v)$, where x_u, x_v, y_u and y_v are real and $x_v - x_u > 0$ and $y_v - y_u > 0$, is said to lie in the first octant if $(y_v - y_u)/(x_v - x_u) < 1$ and in either the first or second octant if $(y_v - y_u)/(x_v - x_u) = 1$. Other octants in which a line segment may lie are defined similarly.

A digital region R is an odd(even) digital quadrangle if there is a quadrangle with edges in each of the four odd(even) octants whose digital image is R. Both odd and even digital quadrangles are called a digital quadrangle. Thus, in this paper a digital quadrangle is either an odd or even digital quadrangle, not just any digital quadrangle.

Let R be a convex digital region with its boundary $B = \{d_0, d_1, ... d_n\}$ listed in counterclockwise order. Partition B into B_1, B_2, B_3 and B_4 as follows: a boundary point $d_j = (x_j, y_j)$ is in B_1 if $x_j - x_{j-1} = 1$ and $y_j - y_{j-1} = 0$ or 1, in B_2 if $x_j - x_{j-1} = 0$ or -1 and $y_j - y_{j-1} = 1$, in B_3 if $x_j - x_{j-1} = -1$ and $y_j - y_{j-1} = 0$ or -1, and in B_4 if $x_j - x_{j-1} = 0$ or 1 and $y_j - y_{j-1} = -1$. If B_i is not a digital line segment for some i, then R is not an odd digital rectangle. Now suppose that for each $1 \leq i \leq 4$, B_i is a digital line segment and without loss of generality assume that $B_1 = \{d_0, d_1, ... d_{j_1}\}$, $B_2 = \{d_{j_1+1}, ... d_{j_2}\}$, $B_3 = \{d_{j_2+1}, ... d_{j_3}\}$ and $B_4 = \{d_{j_3+1}, ..., d_{j_4}\}$, where $d_{j_4} = d_n$ (Figure 1).

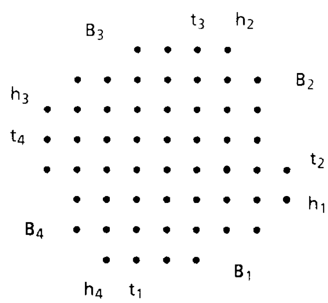

Figure 1. Partition of B.

If the first point d_{j_i+1} of B_{i+1} is a 4-neighbor of the last point d_{j_i} of B_i, then the digital edge Q_{i+1} is B_{i+1}. Suppose that d_{j_i+1} is not a 4-neighbor of d_{j_i}. If $\{d_{j_i}\} \cup B_{i+1}$ is not a digital line segment, then d_{j_i} is not contained in a quadrangle formed by any preimages of B_i's and so R is not an odd digital quadrangle. Therefore, for R to be an odd digital quadrangle, $\{d_{j_i}\} \cup B_{i+1}$ must be a digital line segment and the digital edge Q_{i+1} is $\{d_{j_i}\} \cup B_{i+1}$. For each $1 \leq i \leq 4$, the first point of Q is denoted t_i and called the tail of Q_i and the last point of Q_i is denoted h_i and called its head. Note that t_{i+1} may coincide h_i and if not, t_{i+1} is a 4-neighbor of h_i (See Figure 2).

4. Digital quadrangles

In the following we investigate necessary and sufficient conditions for a convex digital region R with four odd digital edges to be an odd digital quadrangle. As stated above, whether or not R is an odd quadrangle depends on the existence of preimages of digital edges that form a quadrangle not containing points in $\neg R$ which are neighbors of the digital corner points of R. Hence, we

CH2614-6/88/0000/0672$01.00 © 1988 IEEE

pay attention to the corner points and their neighbors. Consider

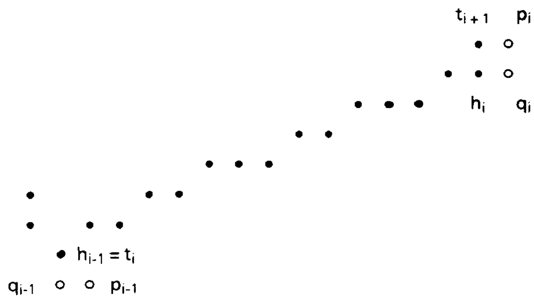

Figure 2. A digital edge

digital edge Q_i of R as shown in Figure 2 after rotation by a multiple of $\pi/2$. Point q_i is the 4-neighbor of h_i to its right and p_i is the 4-neighbor of q_i which is an 8-neighbor of h_i toward Q_{i+1}. The following lemma offers the desired necessary and sufficient conditions.

Lemma 4.1. Let R be a convex digital region with four odd digital edges Q_i, $1 \le i \le 4$. Then R is an odd digital quadrangle if and only if there are some preimages of Q_i's forming a quadrangle that does not contain q_i for any $1 \le i \le 4$.

Proof. If R is an odd digital quadrangle, then there is an odd quadrangle r whose digital image is R. Thus, trivially Q_i is the digital image of an extension of an edge of r to span Q_i for each i. Now suppose that there is an odd quadrangle r with its edges formed by some preimages of Q_i's such that r does not contain q_i for any $1 \le i \le 4$. Since Q_i is the digital image of an extension of an edge of r for all i, r cannot contain any point in $\neg R$, except possibly q_i's, that are neighbors of the boundary points of R. By assumption, however, q_i's are contained in r and thus, R is the digital image of r and hence, an odd digital quadrangle. □

Suppose that for each $1 \le i \le 4$, either $Q_i \cup \{q_i\}$ or $\{q_i\} \cup Q_{i+1}$ is not a digital line segment. Then no quadrangle formed by any preimages of Q_i's contains q_i. For, any preimage of Q_i or Q_{i+1} passes between h_i and q_i, excluding q_i from the corner formed by any preimages of Q_i and Q_{i+1}.

So now assume that for some i, $Q_i \cup \{q_i\}$ and $\{q_i\} \cup Q_{i+1}$, are both digital line segment. If neither $Q_i \cup \{p_i\}$ nor $\{h_i\} \cup Q_{i+1}$ is a digital line segment, then q_i is contained inside the corner formed by any preimages of Q_i and Q_{i+1}, and so R is not an odd digital quadrangle. q_i is excluded from inside the corner formed by every preimage of $Q_i \cup \{p_i\}$ combined with Q_{i+1} and Q_i combined with $\{h_i\} \cup Q_{i+1}$. If one and only one of $Q_i \cup \{p_i\}$ and $\{h_i\} \cup Q_{i+1}$ is a digital line segment then the legal change of Q_i or Q_{i+1} is performed. If both are digital line segments then $Q_i \cup \{p_i\}$ and Q_{i+1} are considered along with Q_i and $\{h_i\} \cup Q_{i+1}$.

These observations lead to the algorithm below that determines whether or not a given digital region R is an odd digital quadrangle. Moreover, it returns a set of sets of four digital line segments whose preimages produce all preimages of R.

Algorithm odd-quadrangle (R,S).

// Given a convex digital region R, returns a set S of all quadruples of digital line segments (Q_1, Q_2, Q_3, Q_4) whose preimages produce quadrangles with R as their digital image if R is an odd digital quadrangle and returns ∅ otherwise.//

1. Obtain the boundary of R, $B = \{d_0, d_1, \ldots, d_n\}$, listed in counterclockwise order, and partition B into B_1, B_2, B_3 and B_4.

2. If for some i, B_i is not a digital line segment

then set S←∅ and return.

3. For each $1 \le i \le 4$, obtain Q_i from B_i and determine h_i and t_i.

4. If for each $1 \le i \le 4$, either $Q_i \cup \{q_i\}$ or $\{q_i\} \cup Q_{i+1}$ is not a digital line segment

then S ←{(Q_1,Q_2,Q_3,Q_4)} and return.

5. S ←{(Q_1,Q_2,Q_3,Q_4)}

6. For i = 1 to 4 do the following:

6.1. if $B_i \cup \{q_i\}$ and $\{q_i\} \cup B_{i+1}$ are both digital line segment

6.2. then for all current (Q_i,Q_{i+1},Q_{i+2},Q_{i+3}) of S do

$Q_i' \leftarrow Q_i \cup \{p_i\}$

$Q_{i+1}' \leftarrow \{h_i\} \cup Q_{i+1}$

$S' \leftarrow S \setminus \{(Q_i, Q_{i+1}, Q_{i+2}, Q_{i+3})\}$

if Q_i' is a digital line segment

then $S' \leftarrow S' \cup \{(Q_i', Q_{i+1}, Q_{i+2}, Q_{i+3})\}$

if Q_{i+1}' is a digital line segment

then $S' \leftarrow S' \cup \{(Q_i, Q_{i+1}', Q_{i+2}, Q_{i+3})\}$

S←S'

7. Return .

Theorem 4.2. Algorithm odd-quadrangle determines whether or not a given convex digital region R is an odd digital quadrangle in $O(n)$ time, where n is the number of boundary points of R. Moreover, if R is an odd digital quadrangle, it returns sets of four digital edges whose preimages form all odd quadrangle r which are a preimage of R.

Proof. The correctness of the algorithm is due to Lemma 4.1 and discussions preceding the algorithm as S contains all the possible quadruples of digital line segments that excludes every q_i. It takes $O(n)$ time to obtain the boundary B of R assuming that R is given by a run-length code and to partition B because one traversal of B is sufficient. So step 1 takes $O(n)$ time. For each i, $1 \le i \le 4$, whether or not B_i is a digital line segment may be determined in time linear to the number of points in B_i and thus step 2 in $O(n)$ time.

Trivially, step 3 takes constant time. Obviously step 4 also takes $0(n)$ time. Since steps 6.1 and 6.2 take $0(n)$ time and are iterated four times, and in each iteration there are at most 16 sets of digital edges, step 6 is $0(n)$ time. Step 7 is of $0(1)$ time, and so all steps together take $0(n)$ time. □

Similarly an algorithm, called even-quadrangle, may be designed to determine whether or not a convex digital region is an even digital quadrangle and return its four digital edges if it is.

5. Digital squares

A square can be described by four parameters ($\alpha,\delta,\gamma,\rho$), where α is the slope of its edge in the first (second) octant, δ and γ are the y-intercept and x-intercept, respectively of the extension of the edge , and ρ its side length.

Consider an odd digital quadrangle R and its four digital edges $\{Q_1,Q_2,Q_3,Q_4\}$ determined by the algorithm 'odd-quadrangle'. R is a digital square if and only if there is a preimage ℓ_i of Q_i for each $1 \le i \le 4$ such that ℓ_i's form a square r. Let α_i, γ_i and δ_i be the slope, x-intercept and y-intercept of the extension of ℓ_i, respectively. Then $\alpha_1 = \alpha_3$, $\alpha_2 = \alpha_4$ and $\alpha_2 = -1/\alpha_1$. Also if ρ_1 is the distance between ℓ_1 and ℓ_3 and ρ_2 is the distance between ℓ_2 and ℓ_4, then $\sqrt{1+\alpha_1^2}\,\rho_1 = \delta_3 - \delta_1$ and $\sqrt{1+(1/\alpha_2)^2}\,\rho_2 = \gamma_2 - \gamma_4$. Denoting $\delta_3 - \delta_1$

by ρ'_1 and γ_2-γ_4 by ρ'_2, we must have $\rho'_1 = \rho'_2$ because $\rho_1 = \rho_2$ and $\alpha_1 = 1/\alpha_2$.

Consider the feasible parameter regions; (α,δ) domains of Q_1 and Q_3 and $(-1/\alpha,\gamma)$ domains of Q_2 and Q_4. These domains are each either a triangle or a convex quadrangle as shown in [2]. From (α,δ) domains of Q_1 and Q_3, we may obtain $(\alpha_1,\rho'_1$ and) domain by plotting α against $\rho'_1 = \delta_3$-δ_1, from $(-1/\alpha,\gamma)$ domains of Q_2 and Q_4, obtain $(-1/\alpha_2,\rho'_2)$ domain by plotting $-1/\alpha$ against $\rho'_2 = \gamma_2$-γ_4. (Figure 3). Then for all $1 \le i \le 4$, there is ℓ_i for Q_i forming a square r which is a preimage of R if and only if (α_1,ρ'_1) and $(-1/\alpha_2,\rho'_2)$ domains have nonempty intersection.

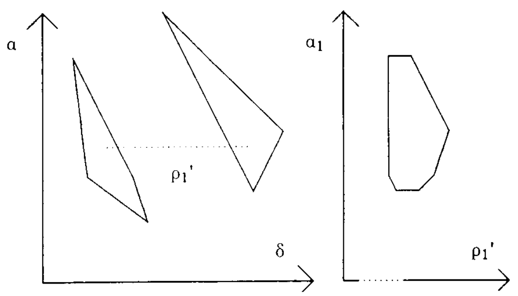

Figure 3. (α_1,ρ_1') *domain from* (α,δ) *domains of* Q_1 *and* Q_2.

If (α,ρ') is a point belonging to both (α_1,ρ'_1) and $(-1/\alpha_2,\rho'_2)$ domains, let $\alpha_1 = \alpha_3 = \alpha$, $\alpha_2 = \alpha_4 = -1/\alpha$, (α,δ_1) and (α,δ_3) be points in (α,δ) domains of Q_1 and Q_3, respectively, such that $\rho' = \delta_3$-δ_1, and $(-1/\alpha,\gamma_2)$ and $(-1/\alpha,\gamma_4)$ be points in $(-1/\alpha,\gamma)$ domains of Q_2 and Q_4, respectively, such that $\rho' = \gamma_2$-γ_4. Then ℓ_1 with (α,δ_1) ℓ_2 with $(-1/\alpha,\gamma_2)$, ℓ_3 with (α,δ_3) and ℓ_4 with $(1/\alpha,\gamma_4)$ are preimages of Q_1, Q_2, Q_3, Q_4, respectively, and form a square r that is a preimage of R. The length of the sides of r is $\rho'/\sqrt{1+\alpha^2}$. Treatment of an even digital quadrangle is similar. Based on the above discussion, we design an algorithm to decide whether or not a given convex digital region is a digital square.

Algorithm odd-square (R,a).
//If R is an odd digital square, returns (α, ρ') in a for squares r whose digital image is R and if not, returns $\varnothing$.//

1. odd-quadrangle (R,S).
2. Set $a \leftarrow \varnothing$.
3. If $S = \varnothing$ then stop.
4. For all (Q_1,Q_2,Q_3,Q_4) of S do
   - 4.1. Obtain (α,δ) domains of Q_1 and Q_3 and $(-1/\alpha,\gamma)$ domains of Q_2 and Q_4, each of which is either a triangle or a convex quadrangle.
   - 4.2. From (α,δ) domains of Q_1 and Q_3 obtain (α_1, ρ'_1) domain and from $(-1/\alpha,\gamma)$ domains of Q_2 and Q_4 obtain $(-1/\alpha_2,\rho'_2)$ domain.
   - 4.3. Compute the intersection a' of (α_1,ρ'_1) and $(-1/\alpha_2,\rho'_2)$ domains and set $a \leftarrow a \cup a'$.
5. Return

Theorem 5.1. Algorithm odd-square runs in $O(n)$ time determining whether or not a given convex digital region R is an odd digital square. Moreover, if R is an odd digital square, it returns the range of (α,ρ') values for which square r is a preimage of R.

Proof. The discussions preceding the algorithm shows that it is correct. As shown in Theorem 4.2, step 1 takes $O(n)$ time and trivially step 2 takes $O(1)$ time. Since the limiting points of Q_i are computed in $O(n)$ time for each $1 \le i \le 4$, the domain of Q_i is computable in $O(n)$ time, making step 4.1 $O(n)$ time. Since (α,δ) domains of Q_1 and Q_3 are each either a triangle or a convex quadrangle, it takes constant time to compute (α_1,ρ'_1) domain, which is a convex polygon of at most 8 vertices. Similarly, $(-1/\alpha_2,\rho'_2)$ domain can be computed in constant time and so step 4.2 is of $O(1)$ time. Step 4.3 also takes constant time, since finding the intersection of two convex polygons with at most 8 vertices can be computed in constant time [5]. Therefore, the algorithm has $O(n)$ time complexity. □

An $O(n)$ time algorithm, called even-square, may also be designed to test whether or not a given even digital quadrangle is a digital square. Therefore, given a convex digital region R, we determine whether it is a digital quadrangle in $O(n)$ time.

References

[1] T.A. Anderson and C.E. Kim, "Representation of digital line segments and their preimages", *Comput. Vision Graphics Image Processing*, vol. 30, pp. 279-288, 1985.

[2] L. Dorst and A.W.M. Smeulders, "Discrete representation of straight lines,", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 450-463, 1984.

[3] C.E. Kim, "Digital convexity, straightness, and convex polygons," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 618-626.

[4] R. Krishnaswamy and C.E. Kim, "Digital parallelism, perpendicularity, and rectangles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 316-321, 1987.

[5] F.P. Preparata and M.I. Shamos, *Computational Geometry*, New York, Springer-Verlag, 1985.