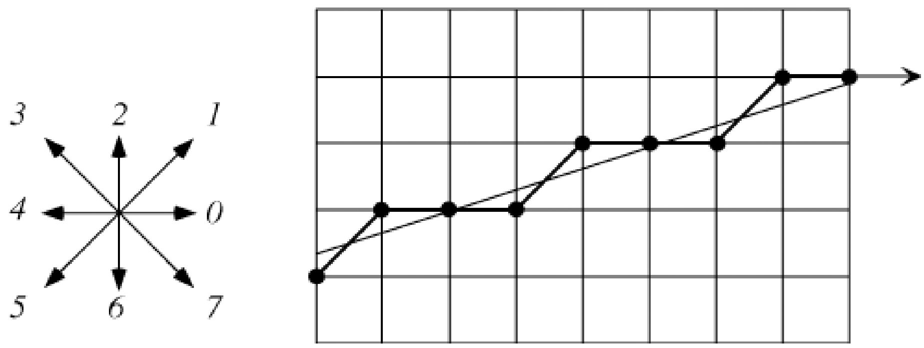
Digital and Cellular Rays

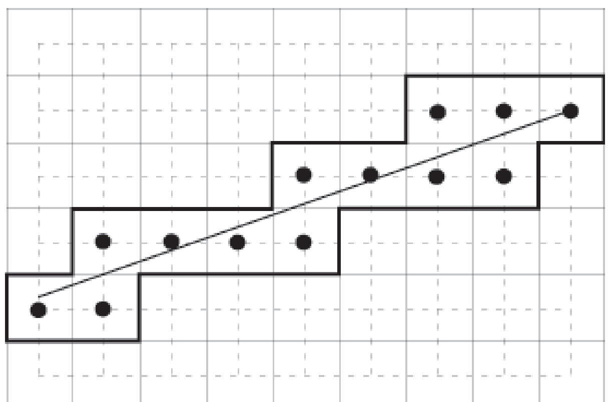
digital rays $i_{\alpha,\beta} = i_{\alpha,\beta}(0) i_{\alpha,\beta}(1) \ldots$ as defined in Lecture 04

slope α and *intercept* β; if α is rational, then we have a *rational digital ray*, and an *irrational digital ray* otherwise

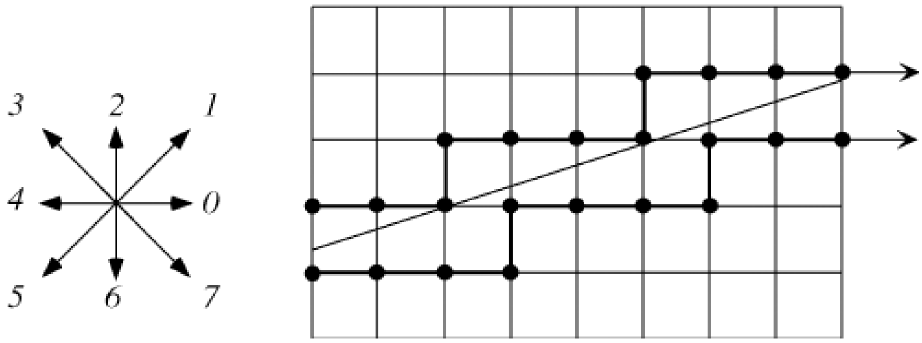


8-path $\quad i_{0.33,1.3} = 10010010\ldots$

alternative digitization model: all 2-cells having a non-empty intersection with the given arc (i.e., *outer Jordan digitization* or *supercover*)



a *cellular straight line segment*: may contain 2×2 blocks of 2-cells translate a cellular straight ray by (0·5,0·5) so that its 2-cells are in grid point positions, its frontier consists of an infinite 4-curve which can be split into the *upper* and *lower digital 4-rays*



Definition 1 *A DSS (4-DSS) is a nonempty finite subpath of a digital ray (upper or lower 4-ray).*

note: instead of *8-DSS* or *8-ray* it is common to use *DSS* or *digital ray*, respectively

Let $U_n = \lceil \alpha n + \beta \rceil$ and $L_n = \lfloor \alpha n + \beta \rfloor$.

($\lceil a \rceil$ denotes the smallest integer greater or equal to a.)

The differences between successive U_ns (L_ns) define the following *chain codes*:

$$\begin{aligned} u_{\alpha,\beta}(n) &= 0 \quad \text{if } U_n = U_{n+1} \\ &= 02 \quad \text{if } U_n = U_{n+1} - 1 \end{aligned}$$

for $n \geq 0$ (analogously for $l_{\alpha,\beta}(n)$). In accordance with our assumption that $0 \leq \alpha \leq 1$, we need to use only codes 0 and 2.

Words

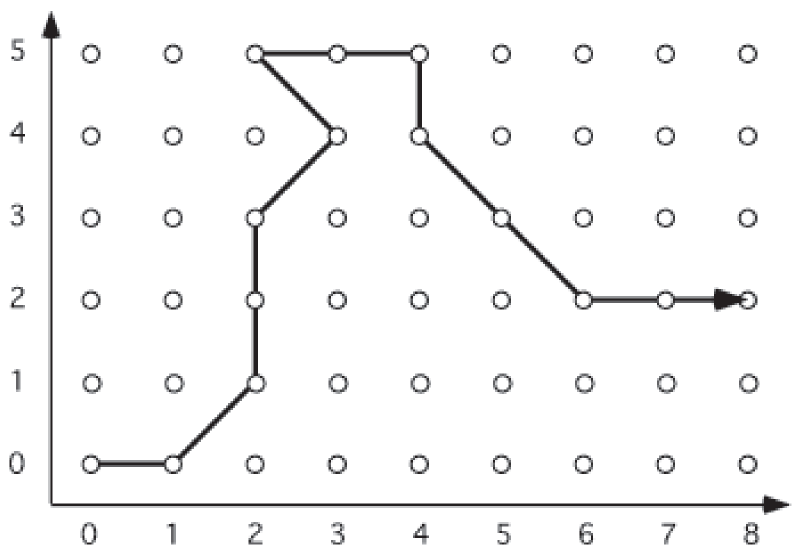
Assuming that a DSS (4-DSS) starts at a recent pixel of interest, we can assume that it starts at $p = (0, 0)$ and ends at a grid point $q = (i, j)$, and is uniquely described by a word

$$w(1) \ldots w(i+1) \in \{0,1\}^\star$$
$$(w(1) \ldots w(m) \in \{0,2\}^\star, \text{ with } m \geq i+1).$$

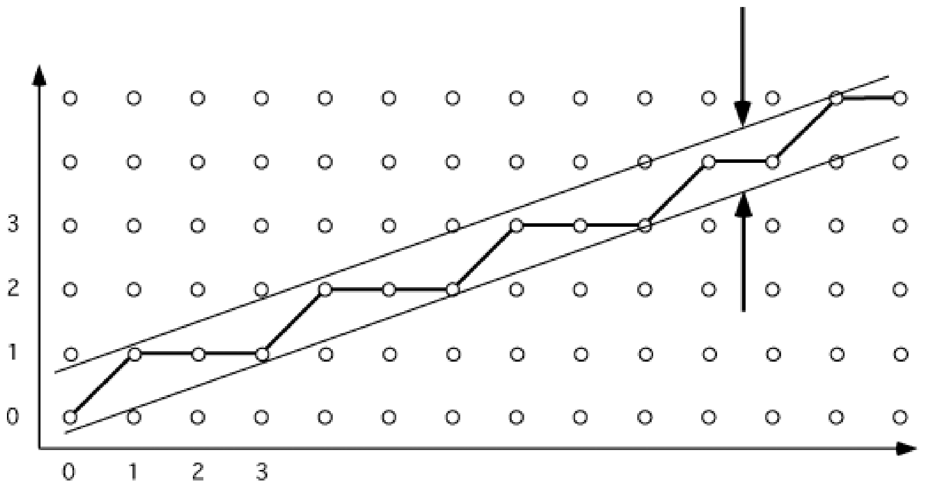
From now on, we simply identify DSS*s* (4-DSS*s*) with such nonempty words on the alphabet $\{0, 1\}$ (alphabet $\{0, 2\}$).

Let $w(1)w(2) \ldots w(m)$ be a word of directional codes. Let $G(w) = \{p_0, p_1, \ldots, p_{m-1}\}$ be the *assigned set of grid points* such that $p_0 = (0, 0)$ and w connects p_0 with p_{m-1} via a sequence of horizontal, vertical, or diagonal steps (as encoded in w) through $p_1, \ldots, p_{m-2}$.

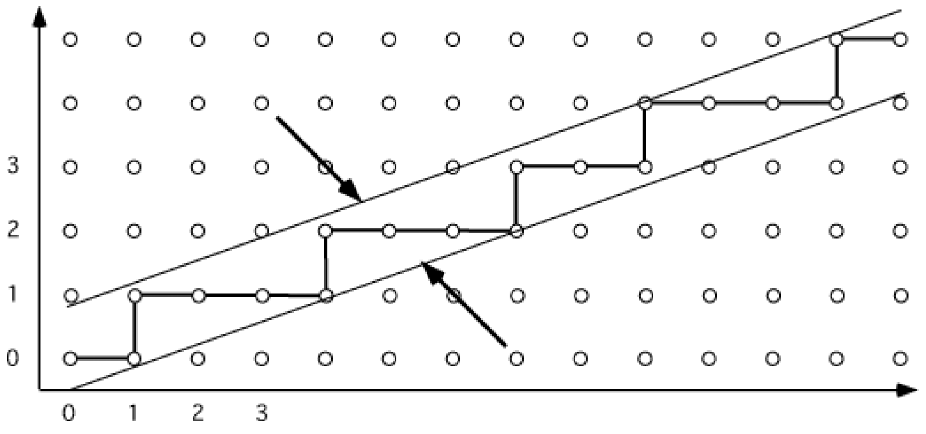
Example: $w = 0122130067700$



Theorem 1 *A word $w \in \{0,1\}^\star$ is an 8-DSS iff $G(w)$ lies between or on two parallel lines with a distance apart (in the y direction) that is less than 1.*



Theorem 2 *A word $w \in \{0,2\}^\star$ is a 4-DSS iff $G(w)$ lies between or on two parallel lines with a distance apart in the main diagonal direction that is less than $\sqrt{2}$.*



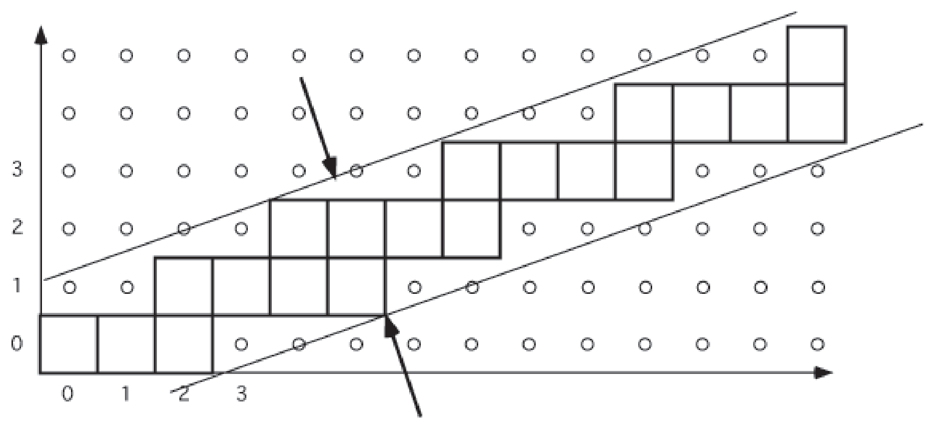
There are four possible oriented diagonals in a grid square. The (oriented) *main diagonal* for a pair of parallel lines is the one that maximizes the dot product with the normal to the lines. It makes angle $135°$ with the positive x-axis if the digitized line has a slope in [0,1).

Supporting Lines

Two parallel lines at minimum diagonal distance that have $G(w)$ between or on them are called a *pair of supporting lines* of w.

Cellular straight segments can also be characterized by a pair of supporting lines. The distance between a pair of parallel lines is measured in the direction of the normal to the lines. Let M be a bounded set in the plane and θ a direction ($0 \leq \theta < 2\pi$). The *width* $w_\theta(M)$ is the minimum distance between a pair of parallel lines such that θ is the direction of the normal to the lines and M lies between or on them. Let $R_{2\times 2}$ be a square formed by four 2-cells.

Theorem 3 *A 1-connected set M of 2-cells is cellularly straight iff there exists a direction θ such that $w_\theta(\bigcup M) \leq w_\theta(R_{2\times 2})$.*



Self-Similarity

The following is an 8-DSS:

0101010101010101011010101010101010110101010101010
    10110101010101010101101010101010101011

An initial formulation of necessary conditions for self-similarity of digital straight lines was given by H. Freeman in 1970:

> "To summarize, we thus have the following three specific properties which all chains of straight lines must possess:
>
> (F1) at most two types of elements can be present, and these can differ only by unity, modulo eight;
>
> (F2) one of the two element values always occurs singly;
>
> (F3) successive occurrences of the element occurring singly are as uniformly spaced as possible."

For the example above: 0 occurs singly, and the run length of the 1′*s* is as follows:

111111121111111211111112111111121111112

Here, 2 occurs singly, the run lengths 1 and 2 only differ by 1, and the run length of the 1*s* is as follows:

87777

Here, 8 occurs singly, the run lengths 7 and 8 only differ by 1, and there is finally just one run of 7*s*.

Let s be a finite word, and let $l(s)$ and $r(s)$ be the run lengths of nonsingular letters to the left of the first singular letter and to the right of the last singular letter in s.

Reduction operation $R(s)$ produces a word that results from s by replacing by their run lengths all subwords of nonsingular letters in s that are between two singular letters in s and deleting all other letters in s. Starting with a word u, it produces a sequence of words $u_0 = u$, $u_1 = R(u_0)$, $u_2 = R(u_1)$, and so forth.

Definition 2 *A finite chain code u has the* DSS property *iff $u = u_0$ and any nonempty word $u_n = R(u_{n-1})$ satisfies*

(L1) *There are at most two different letters a and b in u_n, and, if there are two, then $|a - b| = 1$ (modulo 8 in the case of u_0).*

(L2) *If there are two different letters in u_n, at least one of them is singular.*

as well as the following two conditions:

(S1) *If u_n contains only one letter a or two different letters a and $a + 1$, then $l(u_{n-1}) \leq a + 1$ and $r(u_{n-1}) \leq a + 1$.*

(S2) *If u_n contains two different letters a and $a + 1$ and a is nonsingular in u_n, then u_n starts with a if $l(u_{n-1}) = a + 1$ and ends with a if $r(u_{n-1}) = a + 1$.*

Syntactic Characterization

Following L. D. Wu (1982), we have the following:

Theorem 4 *A finite 8-arc is an 8-DSS iff its chain code satisfies the DSS property.*

$$
\begin{aligned}
u_0 &= 110111011101110111101110111011110111011101110\\
&\qquad\quad 1111011101110111011110111011101110111\\
s(0) &= 0, \quad n(0) = 1, \quad l(0) = 2, \quad r(0) = 3\\
u_1 &= 3334334334333334334\\
s(1) &= 4, \quad n(1) = 3, \quad l(1) = 3, \quad r(1) = 0\\
u_2 &= 2232\\
s(2) &= 3, \quad n(2) = 2, \quad l(2) = 2, \quad r(2) = 1\\
u_3 &= \varepsilon
\end{aligned}
$$

The input chain code u_0 is now represented by a *syntactic code*, which is as follows for the example:

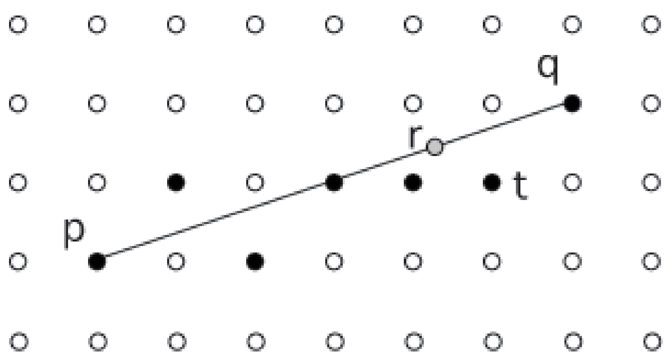
	s	n	l	r
0	0	1	2	3
1	4	3	3	0
2	3	2	2	1

This code consists of integers in four columns s, n, l, and r. The DSS property imposes constraints on these integers so that a given word u can be classified as being a DSS or not.

The Chord Property

A. Rosenfeld gave in 1974 a first formal characterization of digital straight lines, which led to a better specification of Freeman's property (F3), and finally to Wu's theorem.

Definition 3 *A set M of grid points satisfies the* chord property *iff, for any two distinct p and q in M and any point r on the (real) line segment pq, there exists a grid point $t \in M$ such that* $\max(|x_r - x_t|, |y_r - y_t|) < 1$.



An 8-arc is called *irreducible* iff its set of grid points does not remain 8-connected if a nonendpoint is removed from it.

Theorem 5 *A finite irreducible 8-arc $u \in \{0, 1\}^\star$ is a DSS iff $G(u)$ satisfies the chord property.*

DSS Recognition Algorithms

examples of possible applications:

(i) segment borders (8-paths) or frontiers (4-paths) of regions into subsequent DSS*s* of maximum length (the resulting polygonal approximation supports further processing or analysis)

(ii) approximate edges in pictures (e.g., road borders in a driver support system) or calculated 4- or 8-arcs (e.g., skeletons in 2D or 3D picture analysis)

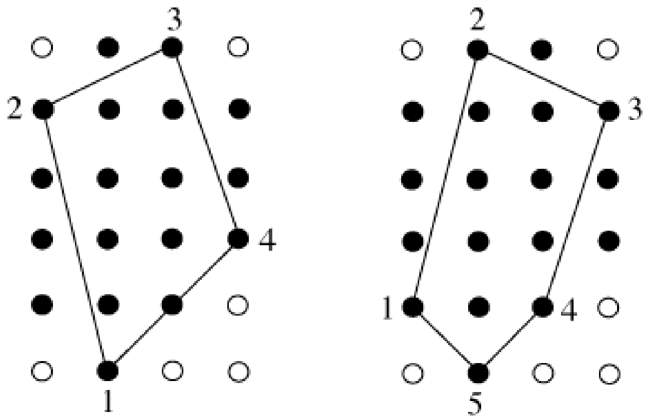


illustration to (i): the start point and the orientation of tracing influences in general the resulting segmentation into maximum-length DSS*s*.

Properties of Algorithms

DSS recognition algorithms can be designed applying selected characterizations as given above

The computational problem is as follows: the input is a sequence of chain codes $i(0), i(1), \ldots$ where (without loss of generality) $i(k) \in \{0, 1\}$ and $k \geq 0$.

An *offline DSS recognition algorithm* decides whether a finite word $u \in \{0, 1\}^\star$ is a DSS.

An *online DSS recognition algorithm* reads the successive chain codes $i(0), i(1), \ldots$ and determines the maximum $k \geq 0$ such that $i(0), i(1), \ldots, i(k)$ is a DSS but $i(0), i(1), \ldots, i(k), i(k+1)$ is not.

A recognition algorithm has linear run time behavior (is a *linear algorithm*) if it runs in $\mathcal{O}(n)$ time (i.e., it performs at most $\mathcal{O}(|u|)$ computation steps for any finite input word $u \in \{0, 1\}^\star$). An online algorithm is linear if it uses *on the average* a constant number of operations for each input chain code symbol.

Analogous definitions can be given for 4-DSS recognition algorithms.

Coursework

Related material in textbook: Sections 9.1, 9.2, 9.3.1, 9.3.2, and the beginning of 9.6 (up to the end of 9.6.1). Solve Exercise 11 on page 337.

A.15. [6 marks] Implement algorithm **CHW1982a** as described in Section 9.6.2. Generate randomly (at least 100) 8-DSS*s* of varying length (i.e., number of chain codes) within a (virtual) 1000×1000 binary picture, run **CHW1982a** on them and compare your measurements about run-time and length of the syntactic code (i.e., maximum of k in the table where elements changed after initialization) with the formulas on page 330.

(a) You may apply Bresenham's algorithm for generating DSSs (or a line drawing program with subsequent tracing of generated line segments for reading the chain code sequence).

(b) The start pixel can always be the same [e.g., the origin (0,0)], and considering (i.e., generating) DSSs in the first octant only is also fine, because these will also produce any possible type of "syntactic code complexities".

(c) Length $k_{\max} = 9$ should be sufficient for segments generated in a (virtual) 1000×1000 picture.

(d) Run-time measurements will (!) be scattered around a line. A sliding mean can be used to achieve a better appearance as being "a line".

(e) Optional **[1 mark]** you may also generate examples of two subsequent DSSs - then you will also see cases where "no" is returned.