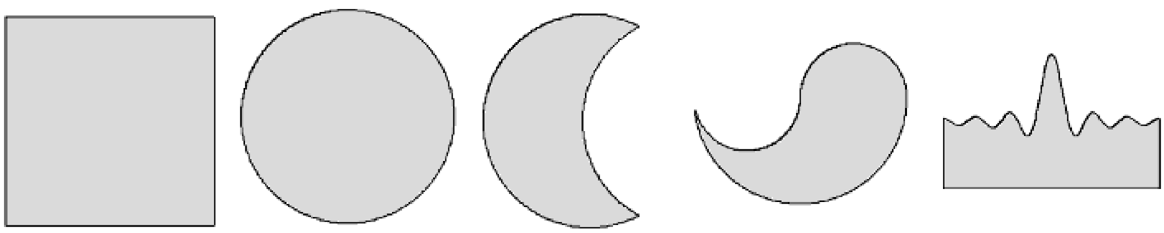
Arcs and Curves

Arcs and curves can be defined by parametrizations (C. Jordan in 1893) or by topological definitions (one-dimensional manifolds, see P. Urysohn in 1923, or K. Menger in 1932). Here we simply assume that *arcs* or *curves* in the Euclidean plane intersect the lines of the orthogonal regular grid a finite number of times, and that arcs or curves do have a measurable length.

An arc has exactly two endpoints; a curve is the general case. An arc or curve is *simple* if it does not intersect itself. A *simple curve* is topologically equivalent to a circle.

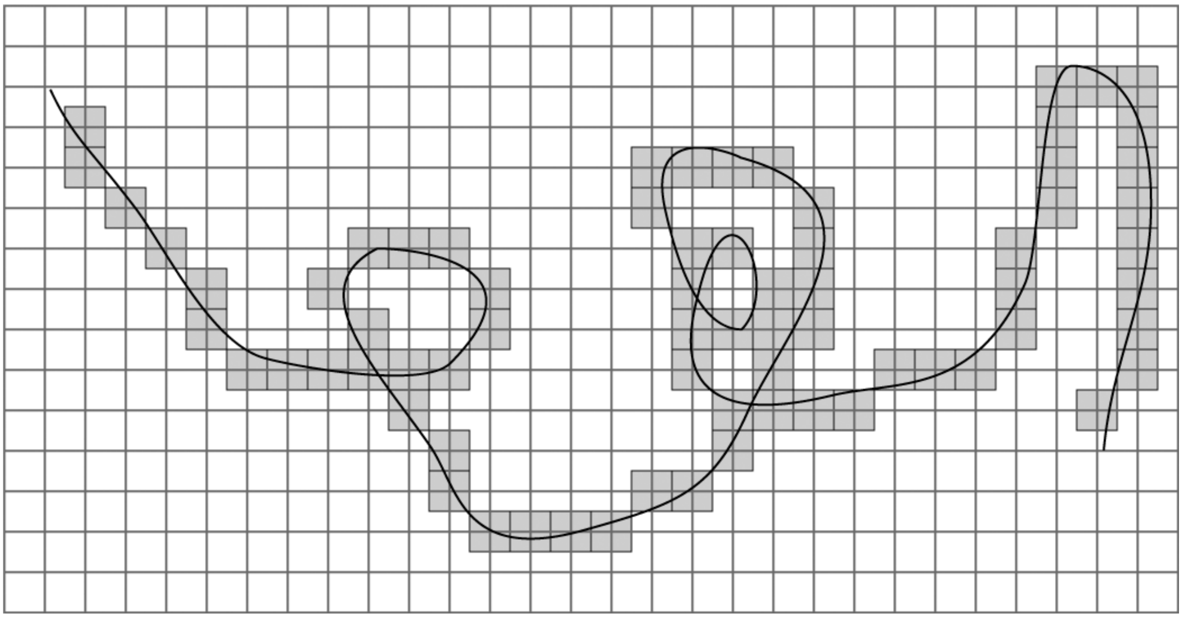
Picture analysis is often concerned with analyzing digitized arcs or curves (e.g., engineering drawings or historic documents).



Simple curves are also the frontiers of simply-connected regions in the plane. The measurement of perimeters of digitized simply-connected regions is another example for curve analysis based on digital pictures.

Grid-Intersection Digitization

The *grid-intersection digitization* $R(\gamma)$ of a planar curve or arc γ is the set of all grid points (i, j) that are closest (in Euclidean distance) to the intersection points of γ with the grid lines.



An intersection point may have the same minimum distance to two different grid points; such an intersection point contributes two grid points to $R(\gamma)$.

The figure above shows all the grid cells (2-cells) around all grid points in $R(\gamma)$.

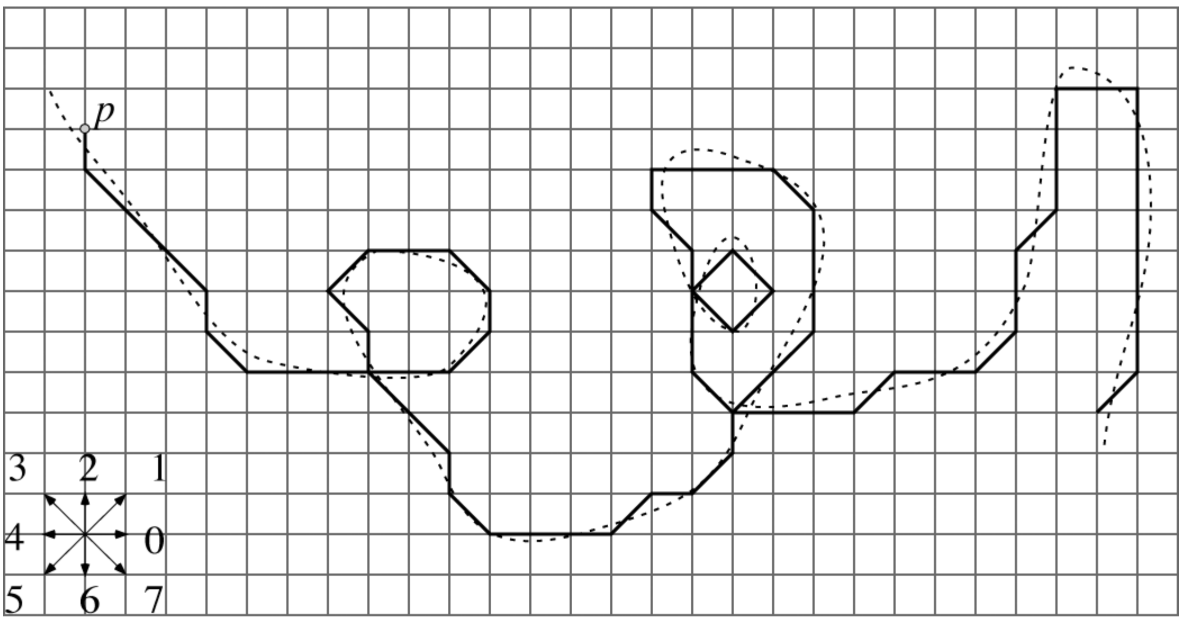
Ordered Sequence of Grid Points

A traversal of γ defines an ordered sequence (list) of grid points in $R(\gamma)$.

Assumptions:

(i) if an intersection point is at the same minimum distance from two grid points, we list only the grid point that has the larger x-coordinate, or, if their x-coordinates are equal, the one with larger y-coordinate; and

(ii) if consecutive intersection points have the same closest grid point, we list that grid point only once.



The resulting ordered sequence of grid points is called the *digitized grid-intersection sequence* $\rho(\gamma)$ of γ.

Directional encoding: starting at p, the arc can be represented by the sequence of codes $677767000001\ldots65$.

Digitization of Straight Segments

Digitization of *straight segments* (i.e., segments of finite length of straight lines) is a standard routine in computer graphics.

We digitize a segment of a line $y = ax + b$ in the first octant (i.e., with slope $a \in [0, 1]$).

(By interchanging the startpoints and endpoints of the segment, we can handle octants "to the left of the y-axis". In the eighth octant, we use a y-increment of -1, and in the second and seventh octants, we interchange the roles of the x- and y-coordinates.)

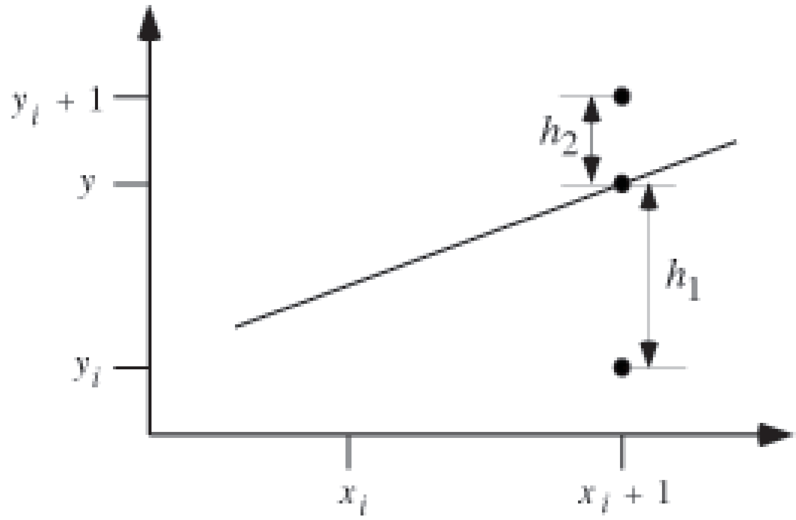
To draw the resulting *digital straight segment* (DSS for short), we increase the x-coordinate stepwise by $+1$; the y-coordinate is "occasionally" increased by $+1$ and remains constant otherwise.

The DSS is a sequence of grid points $(x_i, y_i), i \geq 1$. The point (x_1, y_1) is the grid point closest to the endpoint of the real segment. If we already have point (x_i, y_i), the next point has x-coordinate x_{i+1}, and, for its y-coordinate, we must decide between y_i and $y_i + 1$.

Bresenham's DSS algorithm

(published in 1965)

$$y = a(x_i + 1) + b$$
$$h_1 = y - y_i = a(x_i + 1) + b - y_i$$
$$h_2 = (y_i + 1) - y = y_i + 1 - a(x_i + 1) - b$$



To decide between y_i and $y_i + 1$, we use the following difference:

$$h_1 - h_2 = 2a(x_i + 1) - 2y_i + 2b - 1$$

We choose $(x_i + 1, y_i)$ if $h_1 < h_2$ and $(x_i + 1, y_i + 1)$ otherwise.

For reasons of efficiency (minimization of operations, and integer arithmetic only), we do not use $h_1 - h_2$ for this decision.

Rather, let $p = (x_1, y_1)$ and $q = (x_q, y_q)$ be the grid points closest to the endpoints of the segment, and let $dx = x_q - x_1$ and $dy = y_q - y_1$.

Let $e_i = dx \cdot (h_1 - h_2)$; thus $e_i = 2(dy \cdot x_i - dx \cdot y_i) + b'$, where $b' = 2dy + 2dx \cdot b - dx$ is independent of i.

Thus e_i can be updated iteratively for successive decisions at $x_i + 1$ and $x_i + 2$:

$$e_i = 2dy \cdot x_i - 2dx \cdot y_i + b'$$
$$e_{i+1} = 2dy \cdot x_{i+1} - 2dx \cdot y_{i+1} + b'$$

Thus

$$e_{i+1} - e_i = 2dy(x_{i+1} - x_i) - 2dx(y_{i+1} - y_i) = 2dy - 2dx(y_{i+1} - y_i)$$

because $x_{i+1} = x_i + 1$; this is sufficient for deciding about the y-increment. Let $x_1 = x_p = 0$ and $y_1 = y_p = 0$ give the starting value:

$$e_1 = 2dy \cdot x_1 - 2dx \cdot y_1 + 2dy + dx(2b - 1) = 2dy - dx$$

Bresenham algorithm for the first octant:

1. Let $dx = x_q - x_p$, $dy = y_q - y_p$, $x = x_p$, $y = y_p$, $b_1 = 2 \cdot dy$, $error = b_1 - dx$, and $b_2 = error - dx$.
2. Repeat Steps 3 through 6 until $x > x_q$. Stop when $x > x_q$.
3. Change the value of (x, y) in P to the value of a line pixel.
4. Increment x by 1.
5. If $error < 0$, let $error = error + b_1$, or else increment y by 1 and let $error = error + b_2$.
6. Go to Step 2.

At Step 1, we have $error = e_1 = 2dy - dx$.

The values $b_1 = 2 \cdot dy$ and $b_2 = 2 \cdot dy - 2 \cdot dx$ are used to efficiently update the variable $error$.

The algorithm runs in $\mathcal{O}(x_q - x_p)$ time because, for each i, it involves only a constant number of operations: setting one pixel value, two simple logical tests, one addition, and one or two increments.

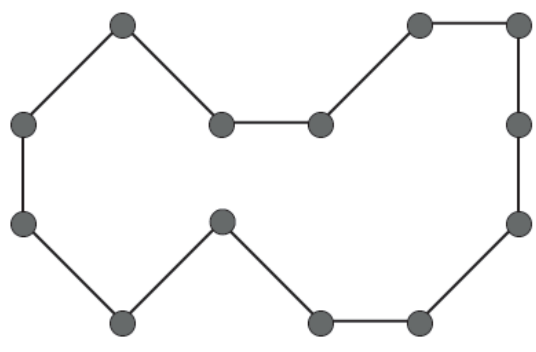
Directional Encoding

Successive pairs of grid points in $\rho(\gamma)$ define steps of length 1 along grid lines and diagonal steps of length $\sqrt{2}$.

The directions of the steps can be represented with codes $0, 1, \ldots, 7$ as shown at the lower left of the figure on page 3; code i represents a step that makes angle $(45 \cdot i)^\circ$ with the positive x-axis (proposed by H. Freeman in 1961).

The directional codes are usually called *chain codes*. The ordered sequence $\rho(\gamma)$ is uniquely defined by a start point p and a chain code.

A *chain* is an ordered finite sequence of code numbers. The *length* of a chain is the number of code numbers in it; note that this length is not related to the geometric length of the arc or curve represented by the chain.

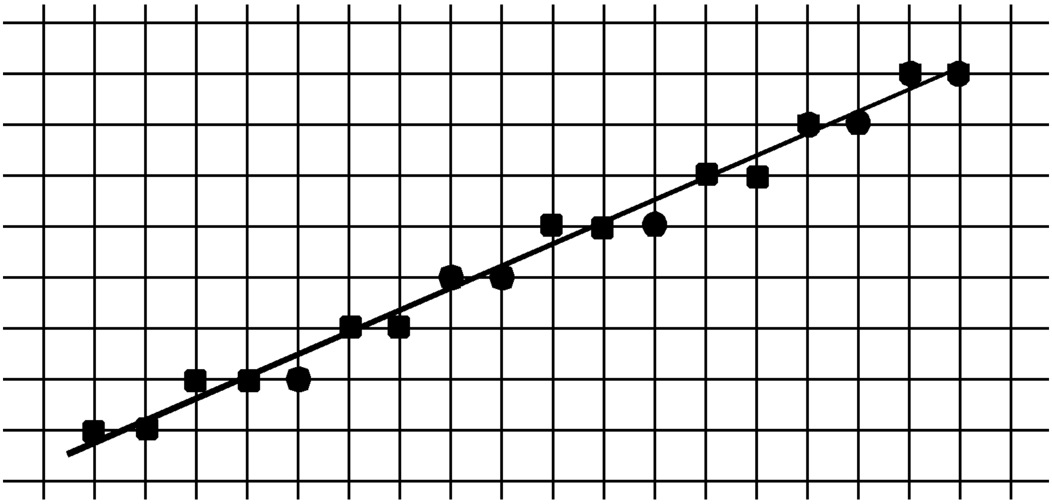


chain code:

length of chain:

length of polygonal curve:

Chain Codes of DSS*s*



chain code: 010010101001010100

note: the y-coordinate was "occasionally" increased by $+1$ and remained constant otherwise; each code value 1 corresponds to an increase by $+1$. Code 1 is *singular* in this sequence, and code 0 is *nonsingular*.

The shown straight segment (prior to digitization) has length 18.72 (assuming grid resolution $h = 1$).

length of chain:

length of polygonal curve:

Chain codes of DSS*s* satisfy interesting properties which can be used in picture analysis when we are interested to recognize DSS*s*.

Chain Codes of Rays

For understanding properties of chain codes of DSSs it is appropriate to consider grid-intersection digitizations of rays (i.e., to eliminate “end-point discussions”):

$$\gamma_{a,b} = \{(x, ax+b) : 0 \leq x < +\infty\}$$

Because of the symmetry of the grid, we can assume (for such studies) that $0 \leq a \leq 1$ (i.e., rays in the first octant).

$\gamma_{a,b}$ has a sequence of intersection points $p_0, p_1, p_2, \ldots$ with the vertical grid lines at $n \geq 0$. Let $(n, I_n) \in \mathbb{Z}^2$ be the grid point closest to p_n, and let the following be true:

$$I_{a,b} = \{(n, I_n) : n \geq 0 \ \wedge \ I_n = \lfloor an + b + 0{\cdot}5 \rfloor\}$$

If there are two closest grid points, we choose the upper one (consistent with our previous assumptions).

($\lfloor a \rfloor$ denotes the largest integer less or equal to a.)

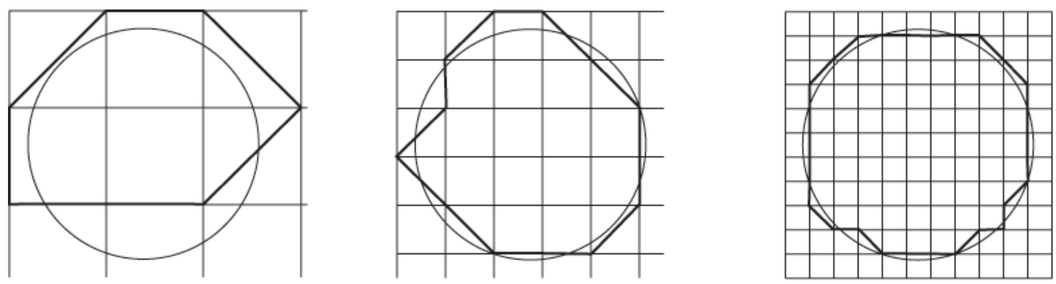
The differences between successive I_ns define the following (infinite) *chain codes* $i_{a,b}$, with

$$\begin{aligned} i_{a,b}(n) = \ I_{n+1} - I_n \ &= 0 \quad \text{if } I_n = I_{n+1} \\ &= 1 \quad \text{if } I_n = I_{n+1} - 1 \end{aligned}$$

for $n \geq 0$. In accordance with our assumption that $0 \leq a \leq 1$, we need to use only the codes 0 and 1. These infinite words on the alphabet $\{0, 1\}$ have been studied for the benefit of straight segment recognition in picture analysis. – More on this follows in Lecture 15.

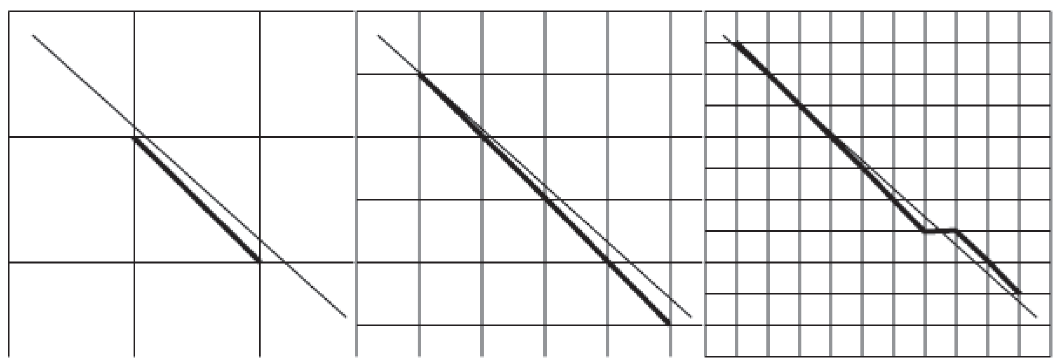
Representations for Varying Resolutions

Flatbed scanners allow to scan originals at different resolutions, standard cameras project objects according to projective geometry, CT-scans can vary in spatial resolution, and so forth:



Digitized geometric shapes are modified by digitization. Moving a circle into different midpoint positions produces (in general) different digital versions of this circle.

Recognition of geometric shapes in pictures often requires some degree of invariance with respect to scale (grid resolution), position or orientation. Line segment of length $0.84085\ldots$, and its digitizations for $h = 4, 8$, and 16:



length of polygonal curve: $\sqrt{2}/4$, $4\sqrt{2}/8$, and $(8\sqrt{2}+1)/16$

Coursework

Related material in textbook: Section 2.3.3 and solve Exercise 7 on page 71.

A.4. [4 marks] Implement the experiments described in Exercise 13 on page 72. Digitize straight segments in a grid of size $2^n \times 2^n$, for $n = 8, 9, 10$. These different grid sizes will also allow to discuss deviations and maximum deviations for digitized straight segments of varying length.