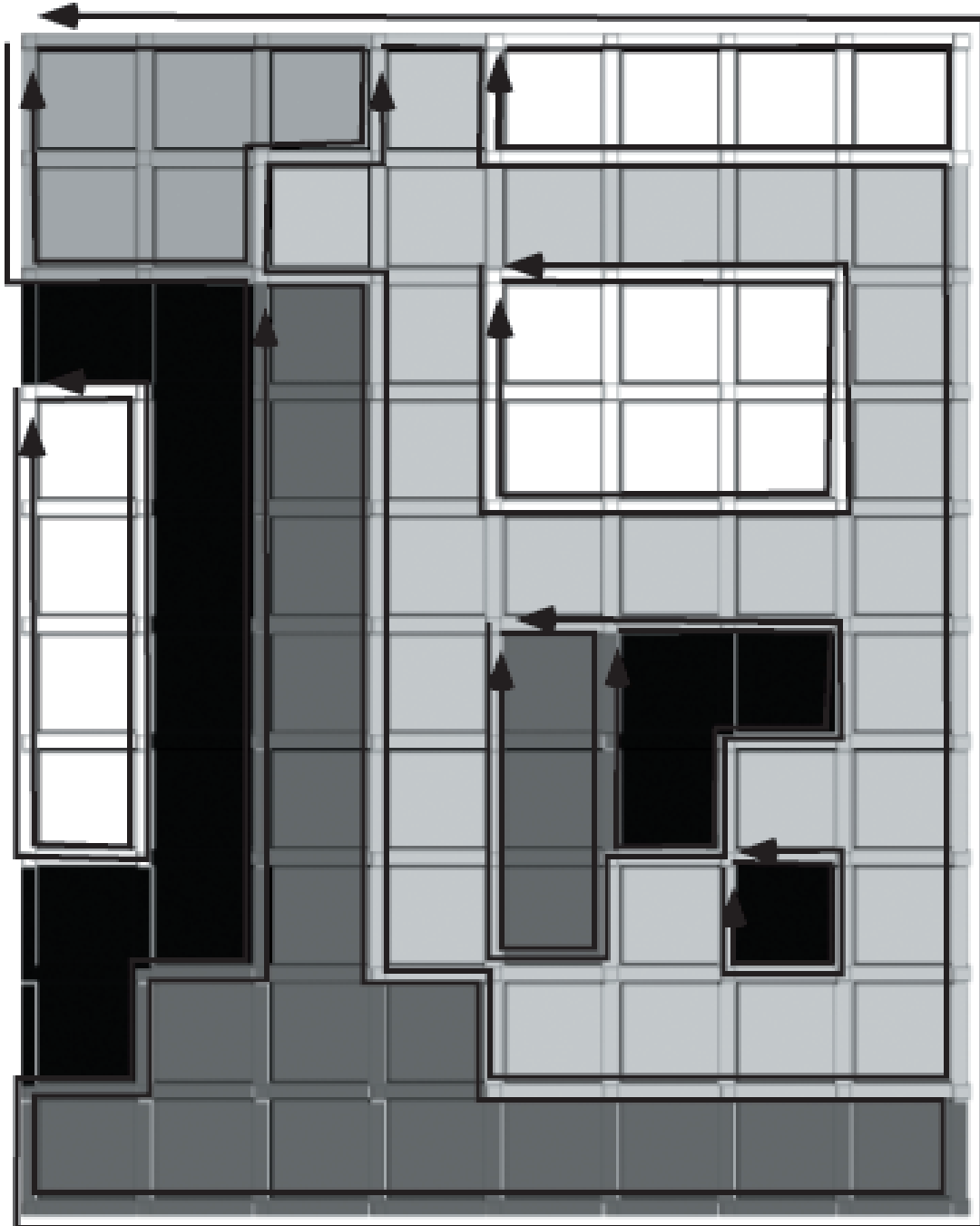


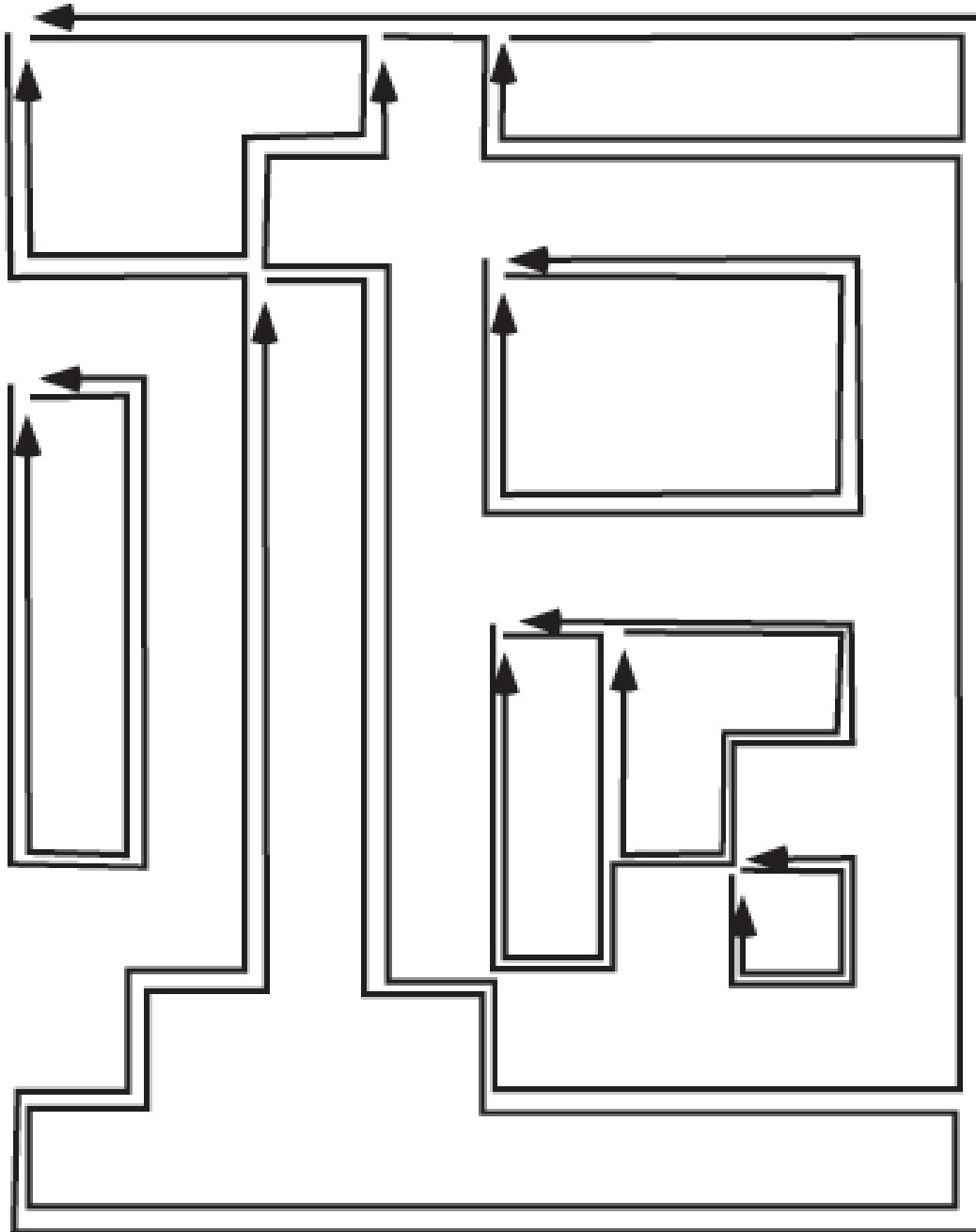


Frontier Tracing



1. take a region as is (picture as on page 5 in Lecture 13)
closure of a region = region with its frontier (a closed set)
2. trace all border cycles defining its frontier

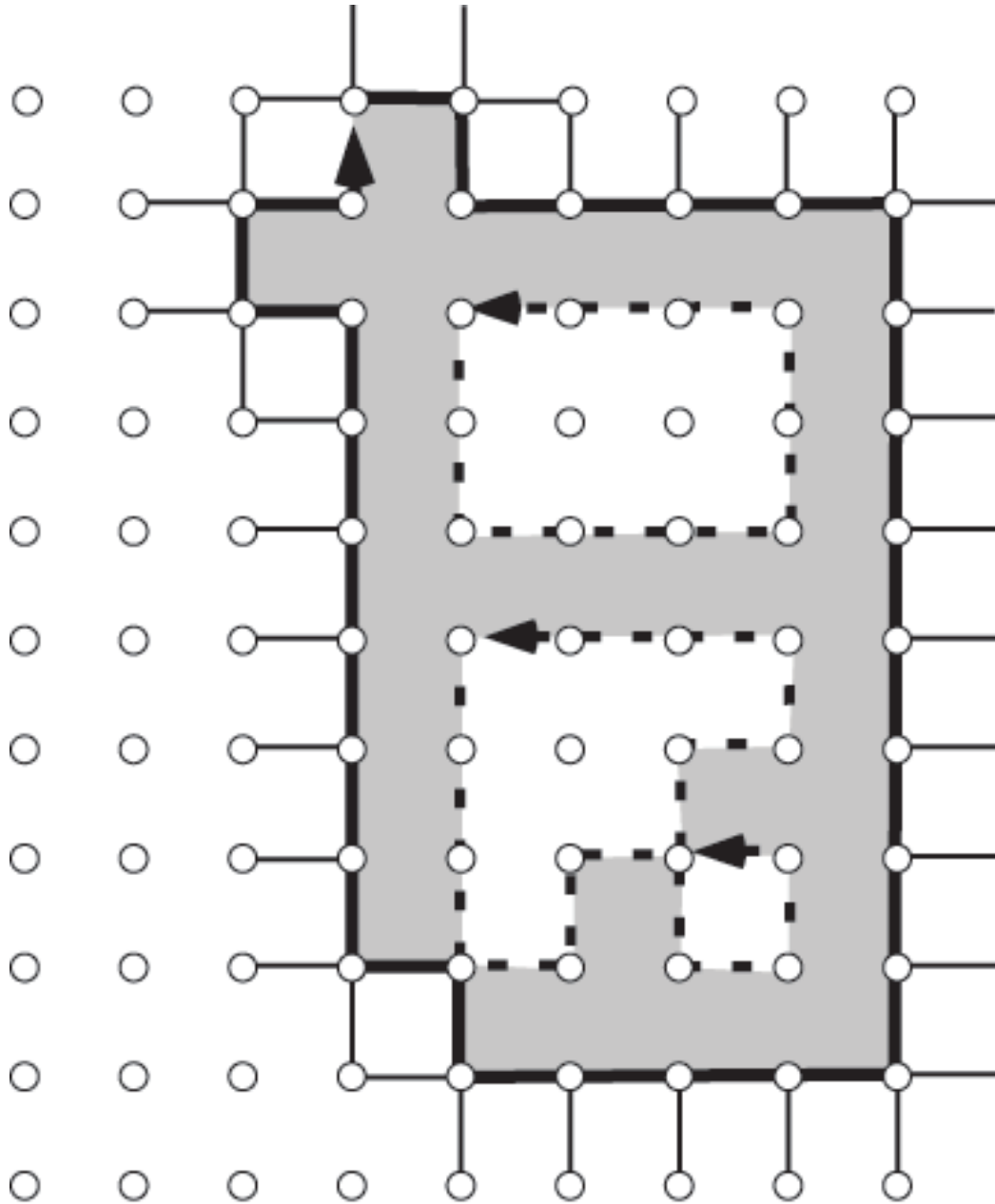
All Border Cycles in Frontier Grid



Note: assuming a black background component

Always consider just one border cycle in the frontier grid (tracing, post-processing, and so forth), then have an empty “drawing board” again for the next border cycle.

A Picture Region in the Frontier Grid



Its frontier splits into one outer border cycle (bold) and three inner border cycles (dashed). Note: no improper holes anymore.

Also shown: all invalid edges of the frontier grid assigned to the outer border cycle

Two Comments on Frontier Grids

(1) History

The 2D grid cell topology was first introduced in a (small) exercise in a 1935 textbook by P. Alexandroff and H. Hopf on topology (of course, without any link to picture processing at that time).

Interpixel frontiers (in the grid cell model) were first used by R. Brice and C.L. Fennema in 1970 for picture segmentation.

The frontier grid is also known as a “half-integer grid” and has been popularized in digital topology (see, e.g., V. Kovalevsky 1989, E. Khalimsky, R. Kopperman, and P.R. Meyer 1990, or C. Fiorio 1995).

(2) Simplification: only 4- or 6-Adjacency

Frontier grids are auxiliary (i.e., technical) tools, which do not need to be “visible” during interactive picture analysis.

The **2D grid cell topology** can be represented by the equivalent model of s -adjacency (see 2nd Equivalence Theorem); both lead to frontier representations in the 2D frontier grid. In the 2D frontier grid, we only have to deal with 4-adjacency between pixels (= grid points in the 2D frontier grid).

The **3D grid cell topology** is the adequate model for 3D picture analysis; frontiers of regions in the 3D grid cell topology can be represented in the 3D frontier grid. In the 3D frontier grid, we only have to deal with 6-adjacency between voxels (= grid points in the 3D frontier grid).

Two Coordinate Systems

Notation

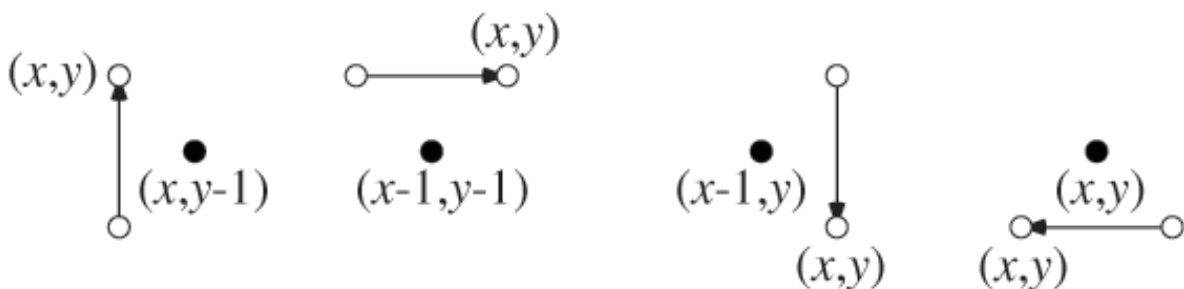
frontier grid \mathbb{F}

picture grid \mathbb{G}

\mathbb{F} and \mathbb{G} are related to one-another by a translation.

2D

In 2D, the translation is $(0.5, 0.5)$ and results into the following:



A step σ in the frontier grid \mathbb{F} (hollow dots) is shown by an arrow pointing to grid point $q = (x, y)$; the corresponding pixel $\tau_\sigma(q)$ in the picture grid \mathbb{G} (filled dot) is on the right of this arrow.

For simplification we also use integer coordinates in \mathbb{F} , and coordinates x and y - just as in \mathbb{G} .

3D

In 3D, the translation is $(0.5, 0.5, 0.5)$, and is used later when discussing surface tracing.

Frontier Tracing Algorithm

Initialization of the frontier grid:

We assume a local circular order in the frontier grid. Here we assume clockwise circular orders.

We use the following notation when q_2 is the direct successor of q_1 in the local circular order at p_i :

$$q_1 \xrightarrow[i]{} q_2$$

Scanning the picture:

Assume we arrive at a pixel (x_0, y_0) in \mathbb{G} that has an 0-cell on a “new” frontier component of a region or of the background (i.e., an s-border pixel which is not on any s-border traced before).

We then generate a 4-path in \mathbb{F} that traverses this frontier:

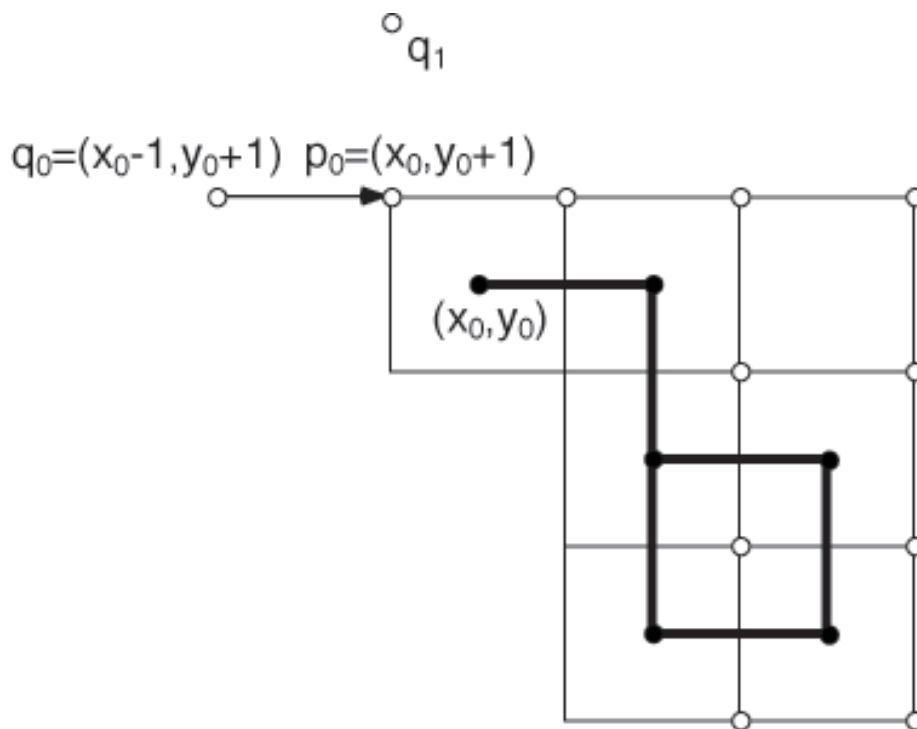
Start at the upper left 0-cell of (x_0, y_0) . This is grid point $p_0 = (x_0, y_0 + 1)$ in \mathbb{F} . Each step in \mathbb{F} , from a grid point p_i to a 4-adjacent grid point p_{i+1} on the frontier goes around a pixel (in \mathbb{G}), keeping the pixel on the right. A step

$$\sigma \in \{\text{UP, RIGHT, DOWN, LEFT}\}$$

specifies how the coordinates of the new pixel are chosen.

First Steps

The filled dots and bold edges illustrate a region in \mathbb{G} . We arrive (say, by standard picture scan) at (x_0, y_0) and notice that this is an s-border point of a region which has not yet been traced.



The first point on the new border cycle is $p_0 = (x_0, y_0 + 1)$ in the frontier grid, (q_0, p_0) is a directed invalid edge in \mathbb{F} to this new border cycle, and we take $\sigma_0 = \text{RIGHT}$ as the initializing step from q_0 to p_0 .

$i = 0$ is the index of point p_0 on the traced frontier.

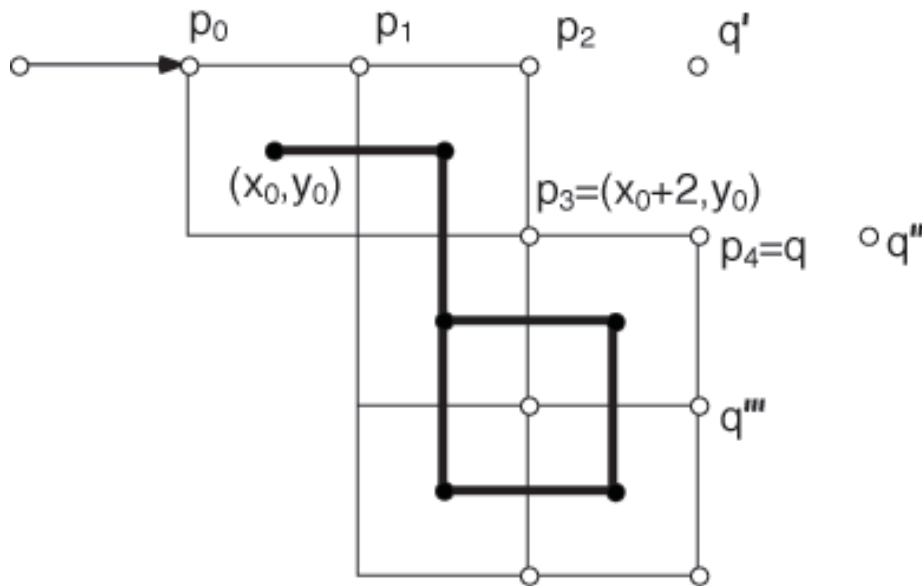
$k = 0$ is the index of q_0 in the local circular order of $p_i = p_0$.

$q_0 \xrightarrow{0} q$ specifies point q_1 ; step $\sigma = \text{UP}$ is not possible at p_0

because pixel $(x_0, y_0 + 1)$ is not in the same region; we continue with $k = 1$ and $q_k \xrightarrow{0} q$. Step RIGHT follows the frontier.

Two Examples of Steps

(1) Assume we arrived at point p_3 . We have $i = 3$.



$p_2 \xrightarrow[3]{q}$ specifies step $\sigma = \text{RIGHT}$, which is possible at p_3 .

(2) Now we have $i = 4$, $p_4 = q$, and $\sigma_4 = \text{RIGHT}$. In $p_3 \xrightarrow[4]{q}$, point q is now point q' in \mathbb{F} which is right of p_2 , describing a step $\sigma = \text{UP}$ at p_4 . This is not following the frontier. (q', p_4) is also not the initial step (q_0, p_0) , so we continue.

Let $k = 0$ and $q_0 = q'$; $q_0 \xrightarrow[4]{q}$ defines now point $q = q''$ and step $\sigma = \text{RIGHT}$ at p_4 . This is not a possible step.

Now check $q_1 = q''$; $q_1 \xrightarrow[4]{q}$ defines point $q = q'''$ and step $\sigma = \text{DOWN}$ at p_4 . This is possible, and we have $p_5 = q'''$.



The Algorithm

1. Let $q_0 := (x_0 - 1, y_0 + 1)$, $p_0 := (x_0, y_0 + 1)$,
 $\{(q_0, p_0) \text{ is a directed invalid edge to the border cycle}\}$
 $\sigma_0 := \text{RIGHT}$,
 $\{\text{initializing step from } q_0 \text{ to } p_0\}$
 $i := 0$, and $k := 0$.
2. Let $q_k \xrightarrow{i} q$ describe a step σ . If the step sequence σ_i, σ
is not possible at p_i ,
 $\{\text{i.e., not continuing the frontier traced so far}\}$
go to Step 4.
3. q is the next grid point on the frontier circuit. Let $i :=$
 $i + 1$, $p_i := q$, and $\sigma_i := \sigma$. Let $p_{i-1} \xrightarrow{i} q$ describe a step
 σ . If the step sequence σ_i, σ is possible at p_i , go to Step
3; otherwise, let $k := 0$, and go to Step 4.
4. If $(q, p_i) = (q_0, p_0)$, go to Step 5.
 $\{\text{The border cycle is finished.}\}$
Otherwise, let $k := k + 1$, $q_k := q$,
 $\{\text{Go around in the local circular order at } p_i.\}$
and go to Step 2.
5. We are back at the original directed invalid edge
 (q_0, p_0) . The frontier circuit is $\langle p_0, p_1, \dots, p_i \rangle$.

Test: "Possible Step?"

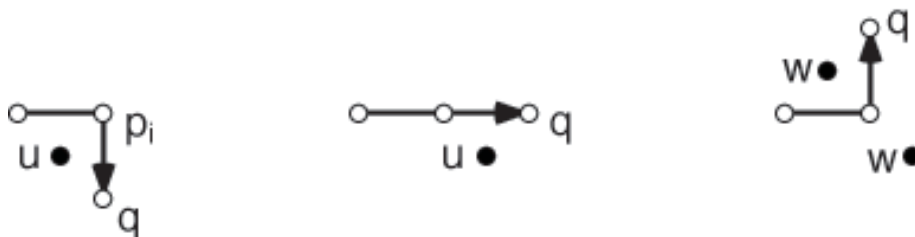
Assume the default order for the pixel values:

$u < v$ iff u less than v in the total order of picture values

In the flip-flop case of s-adjacency, adjacency between pixels that have value v is preferred over adjacency between pixels that have value u if $u < v$.

The frontier tracing algorithm is just a special case of the border tracing algorithm discussed in Lecture 10.

The important difference is that here we test whether the step sequences σ, σ' are possible at a grid point.



If a step sequence σ, σ' is not a left turn, it is possible iff the pixel $\tau_{\sigma'}(q)$ has value $u = P(x_0, y_0)$; if it is a left turn, it is impossible iff the two diagonal pixels other than $\tau_{\sigma}(p_i)$ and $\tau_{\sigma'}(q)$ have the same value $w > u$.

Euler Characteristic

Let M be a subset of the 2D cellular grid. An m -cell c ($0 \leq m \leq 2$) is *invalid* with respect to M iff $c \notin M$, but there is an m' -cell $c' \in M$ ($0 \leq m' \leq 2$) such that c' is incident with c .

Boundary counts for cells in M ($0 \leq i, j \leq 2$):

$$b_{ij}^M(c) = \begin{cases} \text{card}\{c' \in I(c) : \dim(c') = j \wedge c' \text{ is invalid}\} & \text{if } c \in M \text{ and } i = \dim(c) \\ 0 & \text{otherwise} \end{cases}$$

Total boundary counts for M :

$$b_{ij}^M = \sum_{c \in M} b_{ij}^M(c)$$

Note: these counts add values for all (outer and inner) border cycles of M .

Theorem 1 *Let M be a finite union of pairwise disjoint closed (or open) regions in the 2D cellular grid. Then the Euler characteristic of M is as follows:*

$$\chi(M) = \frac{1}{4}(b_{10} - b_{21}) \quad \text{for open regions}$$

and

$$\chi(M) = \frac{1}{4}(b_{12} - b_{01}) \quad \text{for closed regions}$$

Coursework

Related material in textbook: Sections 5.4.3 and (optional) 5.3.3.

A.14. [6 marks] Do Exercise 8 on page 190 (for binary pictures only). Extend the algorithm in such a way that

(i) at the end of tracing a border cycle all those total border counts are available (for this border cycle) which are required (see Theorem on page 11) to calculate the Euler characteristic for the region under consideration. (Note that in a binary picture we have open regions for one picture value corresponding to assumed 4-adjacency, and closed regions for the second picture value corresponding to assumed 8-adjacency.)

(ii) Demonstrate your algorithm on binary pictures of reasonable complexity (see, e.g., the Appendix for Lecture 13), illustrating the obtained (connected) regions and their total border counts together with the resulting Euler characteristic.