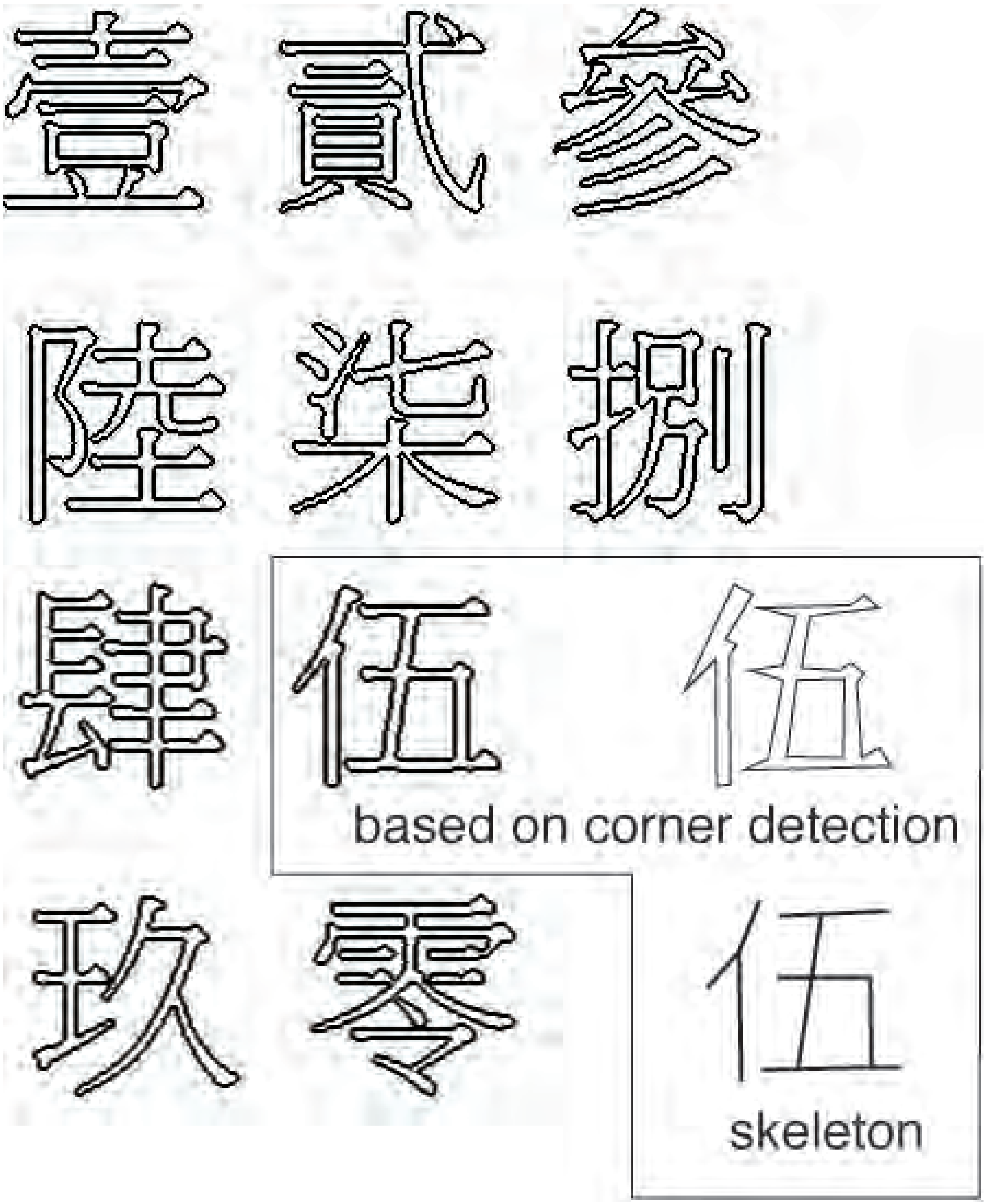
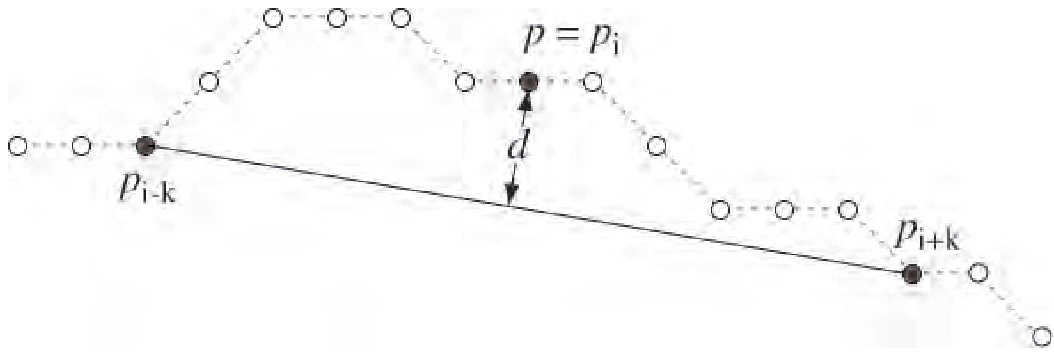
Corners for Pattern Analysis



In the previous lecture we discussed that curvature analysis also allows detecting high-curvature pixels ("corners").

However, many "corner detectors" are not based on curvature estimation but on some type of geometric analysis of curves. For example, the *arc-chord distance measure* is defined by the distance d between a point p on the curve and a symmetric chord defined by a parameter k (e.g., $k = 6$ in the figure).

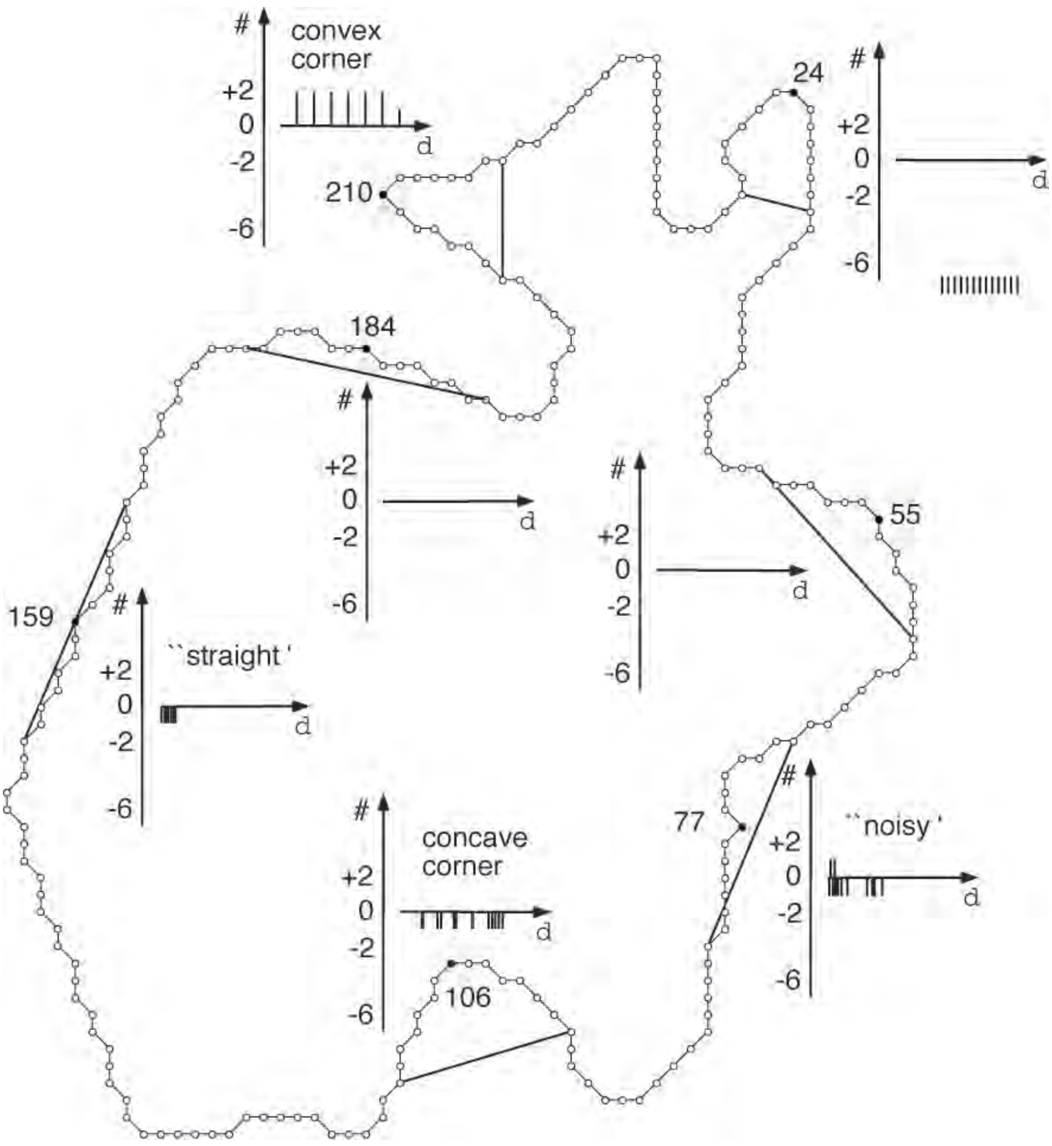


This measure can be calculated by a linear online algorithm (i.e., during tracing the curve), and can be used for some heuristic definitions of "corners".

Note that we do not attempt to define "corner" in a general way; applications and their picture data will define the necessary context.

Histograms of Arc-Chord Distances

Option 1: "insert $2k+1$ bars of height 1" at d if a pixel has arc-chord-distance d. Example for $k=6$:



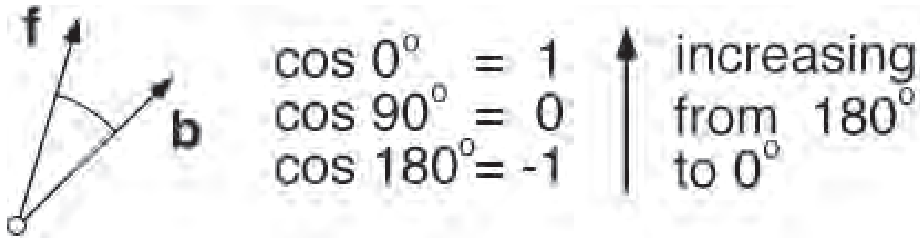
Option 2: have "before-bars" and "after-bars", and a "central-bar" for p_i itself, all again of length 1.

Option 3: have bars b_h, $h = -k, \ldots, 0, \ldots, k$, of varying length d_h, and histograms with axes for h and d

Algorithm RJ1973

[A. Rosenfeld and E. Johnston, 1973]

This algorithm detects a corner at p_i based on analyzing the cosines $c_{i,k}$ of the angles between the forward vector $\mathbf{f}_{i,k}$ and the backward vector $\mathbf{b}_{i,k}$; $c_{i,k}$ is called the *k-cosine angle measure*.



Let $0 < a < 1$ (e.g. $a = 0{\cdot}05$) and $k_0 = \lfloor am \rfloor$, where m is the number of pixels on the digital curve ρ.

We start at $k = k_0$ and decrement k as long as $c_{i,k}$ increases (i.e., the angle decreases).

Suppose this occurs at $k = k_i$ so that $k = 1$ or $c_{i,k_i} \geq c_{i,k_i-1}$ for the first time (i.e., at pixel p_j we have k_j and final value c_{j,k_j}).

In a subsequence of ρ of length k_i centered at p_i

> a corner is detected at p_i iff $c_{i,k_i} > c_{j,k_j}$ for all $j \neq i$ such that $|i - j| \leq k_i/2$ (modulo m).

Note that the algorithm depends on only one parameter a; it is adaptive to the length of the 8-curve.

Algorithm RW1975

[A. Rosenfeld and J.S. Weszka, 1975]

This algorithm uses $k_0 = \lfloor am \rfloor$ and averaged cosine values

$$\bar{c}_{i,k_0} = \begin{cases} \frac{2}{k_0+2} \sum_{k=k_0/2}^{k_0} c_{i,k} & \text{if } k_0 \text{ is even} \\ \frac{2}{k_0+3} \sum_{k=(k_0-1)/2}^{k_0} c_{i,k} & \text{if } k \text{ is odd} \end{cases}$$

In a subsequence of ρ of length k_0 centered at p_i

a corner is detected at p_i iff $\bar{c}_{i,k_0} > \bar{c}_{j,k_0}$ for all $j \neq i$ such that $|i - j| \leq k_0/2$ (modulo m).

ROS

The k-cosine angle measure is an example of a *significance measure* defined in a sliding interval of width $2k_0 + 1$ and evaluated in a reduced interval of width $k_0 + 1$.

The sliding interval defines a *region of support* (ROS) for a detected corner, which is normally considered to be at the center of its ROS. Evaluation of the measure is limited to a subset of the ROS. In the above examples the size of the ROS is uniformly determined by $\lfloor am \rfloor$, i.e. it is the same at every point of the curve.

The PhD thesis by M. Marji (2003) reviews over 100 corner detection and curvature estimation algorithms for 2D digital curves. Curvature estimators based on

- **(C1.1)** the change in the slope angle of the tangent line,
- **(C1.2)** derivatives along the curve; or
- **(C1.3)** the radius of the osculating circle

have been discussed in the previous lecture. Other approaches are based on

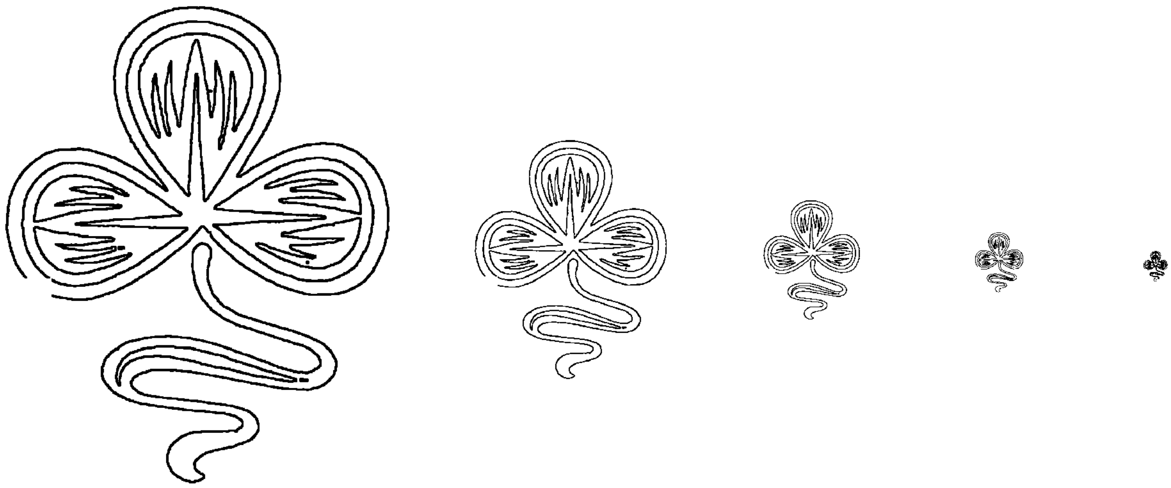
(C2) Estimation of curve properties in a uniformly determined ROS defined by input parameters. Examples of relevant properties are

- **(C2.1)** angles between forward and backward vectors (see above);
- **(C2.2)** approximation errors, e.g. defined by distances between arcs and chords (see below);
- **(C2.3)** directional changes between forward and backward ROSs.

(C3) Estimation of curve properties in an individual determined ROS.

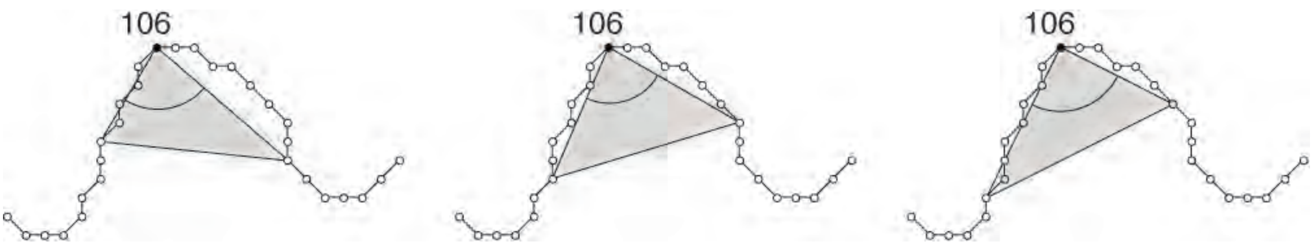
(C2.1)

[L.S. Davis 1977] uses approach **(C2.1)** at various scales to polygonally approximate the curve at different levels of detail.



[K. Deguchi 1988] defines *multiscale curvature* based on minima and maxima of the k-cosine at different scales, where k varies with the scale.

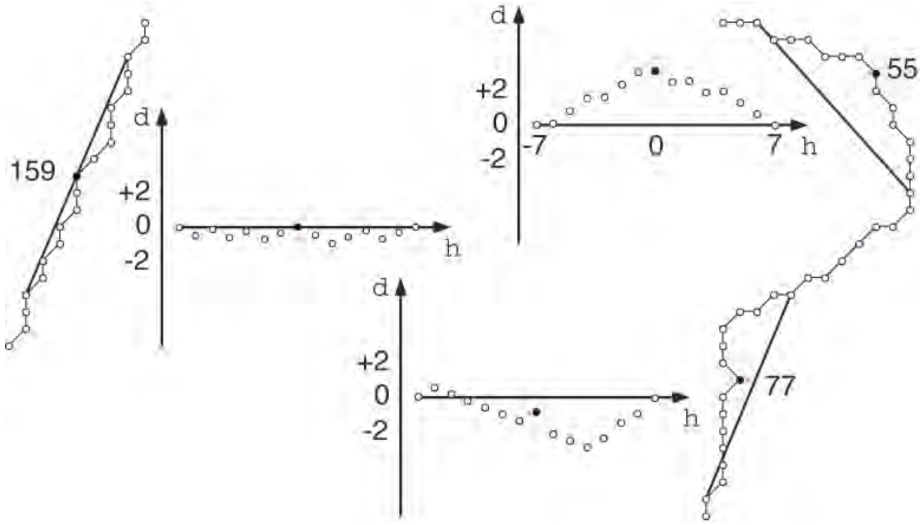
[D. Chetverikov and Z. Szabo, 1999] identify corners at locations at which a triangle of specified size and angle can be inscribed in the curve.



(angle increases from left to right)

(C2.2)

Two of the corner detectors proposed by [W.S. Rutkowski and A. Rosenfeld, 1978] are based on approach **(C2.2)**, specifically on distances between arcs and chords (see figures on pages 2 and 3).



In [M. Fischler and R. Bolles, 1986] points of the curve are classified into the categories "on smooth interval", "on noisy interval", and "corner" by analyzing deviations of arcs of the curve from their chords. (A more efficient implementation of this method was presented by [T.Y. Phillips and A. Rosenfeld, 1987]).

Arc-chord distance is combined with Gaussian scale-space techniques in [J.H. Han and T. Poston, 2001].

(C2.3)

[M.J. Eccles et al., 1977] define a *curvature chain* $c_i = \text{mod}_8(s_i - s_{i-1} + 11) - 3$ where the s_i's are chain codes. This curvature chain is "smoothed" by convolution with a filter.

[F. Arrebola et al., 1997] use comparisons of forward and backward chain-code histograms based on a correlation coefficient measure.

Chain-code histograms on a sliding arc are used in [F. Arrebola et al., 1999].

Other approaches based on curve properties include analysis of the chord lengths between pairs of points k steps apart along the curve [B. Kruse and C.V.K. Rao, 1978] or of the number of DSSs centered at a point of the curve [J. Koplowitz and S. Plante, 1995].

Corners can also be defined by local maxima of a measure of local symmetry [H. Ogawa, 1989].

[D.M. Tsai, H.T. Hou, and H.J. Sou, 1999] describe a corner detection method based on eigenvalues of covariance matrices of neighborhoods of curve points.

(C3)

As an example of approach **(C3)**, studies of the visual perception of shapes [D.J. Langridge, 1972] motivated [B. Rosenberg, 1974] to assign individually-sized neighborhoods to border points of convex regions to detect "dominant points".

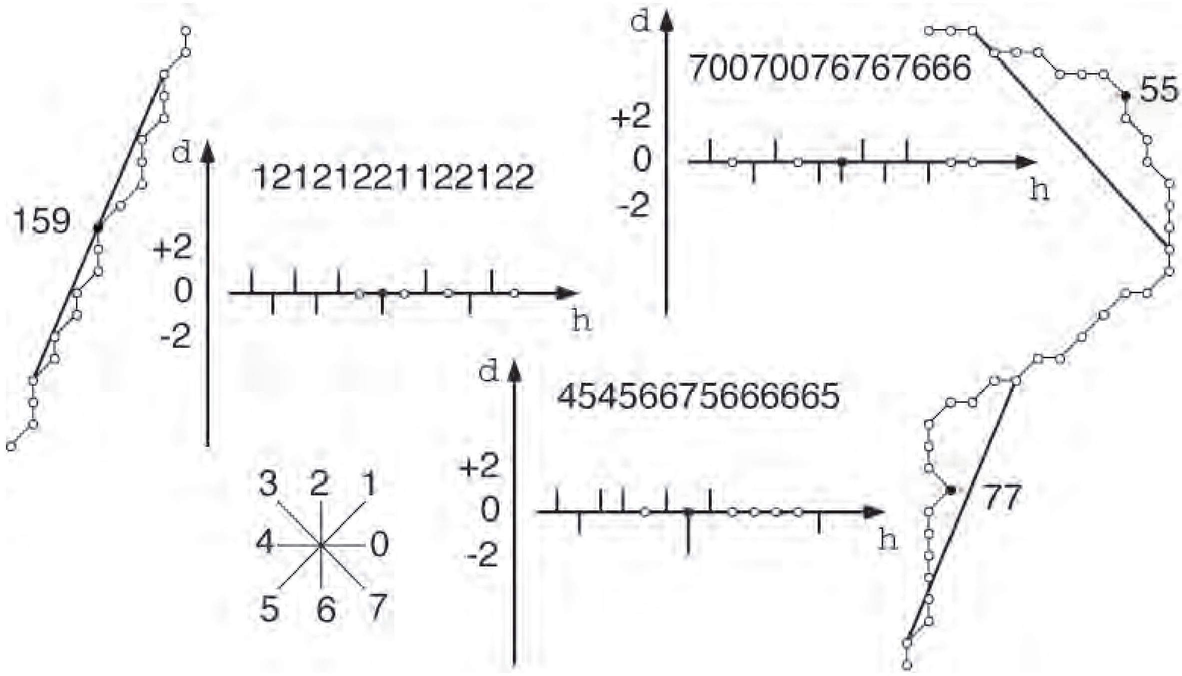
[B.K. Ray and K.S. Ray,1992] give an algorithm for calculation of backward and forward vectors and also allows asymmetric ROSs at a point. The backward and forward vectors point to p_{i-k} and p_{i+l} respectively, where k and l are determined by studying the angular changes that take place when moving away from p_i along the curve.

A (k, l)-cosine measure is used to detect corners; see [P. Reche et al., 2002] for further studies along these lines.

For a combination of multi-scale and individually sized ROSs see [R. Neumann and G. Teisseron, 2002].

(C4): Other Approaches

In [S.H.Y. Hung and T. Kasvand,1983] chain codes are transformed into differential chain codes $c_i = s_i - s_{i-1}$. The maximum lengths of non-zero sequences in the differential code, and sums of consecutive non-zero differential codes (starting with *pair sums* and generalizing to *group sums*), are used to detect “critical points”.



See [C. Arcelli and G. Ramella, 1993] for other applications of differential codes.

[A. Held, K. Abe, and C. Arcelli, 1994] derive a hierarchical polygonal approximation to a curve by detecting candidate “dominant points” and then iteratively removing “least significant points” until a stable polygonal approximation is reached.

Coursework

Related material in textbook: Section 10.4.1.

A.23. [4 marks] Design a corner detection algorithm which combines DSS segmentation with the arc-chord distance measure. Let $q_1, q_2, \ldots, q_h$ be the sequence of all endpixels of DSSs calculated by segmenting the given curve in subsequent maximum-length DSSs. Follow with your program the following heuristics (select $k \geq 1$, which could be related to the total number m of points on the curve):

(a) A corner is one of the endpixels $q_1, q_2, \ldots, q_h$.

(b) For identifying endpixel q_i to be a corner, consider the chord from q_{i-k} to q_{i+k} (modulo h) of length $l_{i,k}$ and calculate the arc-chord distance $d_{i,k}$ between q_i and the chord.

(c) Calculate the ratio $R = d_{i,k}/l_{i,k}$ for classifying q_i to be either no corner (i.e., R is close to zero), or a corner otherwise.

Run your algorithm on sets as suggested for **A.22** (but not using the digital circles, which do not make much sense here). Typically, k should be very small, say $k = 1, 2, 3$.