Chapter 11 Arc Extraction and Segmentation

11.1 Introduction

edge detection: labels each pixel as edge or no edge
additional properties of edge: direction, gradient magnitude, contrast
edge grouping: edge pixels in same region boundary grouped together
edge detector: typically produces short linear disjointed edge segments
each edge segment: has an orientation and a position
edge segments: of little use until aggregated into extended edges
global edge aggregation: upon proximity, relative orientation, contrast, ...
local edge aggregation: extends edges by seeking compatible neighbor edges
global methods: can incorporate domain knowledge into cost functions
global methods: more robust to noisy edge data
global methods: bought at expense of relatively high computational cost
aggregation into extended edges and parametric description: new research area
extended edges: like other curves, can be described at multiple scales
Fig. 3.18(a) 64×64 image of an industrial part
Fig. 3.18(b) edges of (a), large dot: edge center, medium dot: edge extension
Fig. 3.18(c) thinning of (b) leads to the minimally connected graph
Fig. 3.18(d) simple contour following based on orientation compatibility
Fig. 3.18(e) polygonal approximation with vertices highlighted

Fig. 3.18(f) tangent versus arc-length plot
Fig. 3.18(g) same as (d), except letters indicating initial knots
Fig. 3.18(h) fitted straight lines and conics and final set of knots
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 3.18=====

11.2 Extracting Boundary Pixels from a Segmented Image

after segmentation/connected components: region boundary may be extracted
border algorithm: to extract all region boundaries in left-right, top-bottom

11.2.1 Concepts and Data Structures

input image: symbolic image whose pixel values denote region labels
current regions: borders partially scanned but not yet output
past regions: completely scanned and their borders output
future regions: not yet been reached by the scan
at most $2 * number\_of\_columns$ region labels may be active at a time
hash table: used to allow rapid access to chains of a region
each chain: linked list of pixel positions grow-able from beginning to end

11.2.2 Border-Tracking Algorithm

symbolic image and its output from the border procedure

=====Fig. 11.1=====

11.3 Linking One-Pixel-Wide Edges or Lines

segments: may consist of line, endpoint, corner, junctions
symbolic edge image containing junction (3,3) and potential corner (5,3)
=====Fig. 11.2=====
algorithm to track segments like these has to be concerned with

- starting a new segment
- adding an interior pixel to a segment
- ending a segment
- finding a junction
- finding a corner

segments: lists of edge points representing straight or curved lines
pixeltype: isolated, starting, interior, ending, junction, corner
output of the edge-track procedure on the image of Fig. 11.2
=====Fig. 11.3=====

11.4 Edge and Line Linking Using Directional Information

edge linking: labeled pixels with similar enough directions form chains

edge linking: chains identified as arc segment with good fit to curvelike line
=====Garfield 17:11=====

11.5 Segmentation of Arcs into Simple Segments

extracted digital arc: sequence of row-column pairs 4-neighbor or 8-neighbor
arc segmentation: partitions extracted digital arc to fit straight/curved line
endpoints of subsequences: corner points or dominant points
basis for partitioning process: identification of all locations with

- sufficiently high curvature (high change in tangent angle to length)
- enclosed by subsequences fitting different straight lines or curves

simple arc segment: straight-line or curved-arc segment
techniques: from iterative endpoint fitting, splitting to using tangent angle deflection, prominence, or high curvature as basis of segmentation

11.5.1 Iterative Endpoint Fit and Split

one distance threshold d^*
pixel with farthest distance to line AC is B and distance $\geq d^*$ then split
=====Fig. 11.4=====

11.5.2 Tangential Angle Deflection

another approach: identify locations where two line segments meet
exterior angle between two line segments: change in angular orientation
=====Fig. 11.5=====
caution: spatial quantization at small distances can completely mask direction

11.5.3 Uniform Bounded-Error Approximation

segment arc sequence into maximal pieces whose points deviate $\leq$ given amount
optimal algorithms: excessive computational complexity
caption of Fig. 11.6 explains Tomek's segmentation technique
=====Fig. 11.6=====

11.5.4 Breakpoint Optimization

after initial segmentation: shift breakpoints to produce better segmentation
first: shift odd final point i.e. even beginning point
then: shift even final point i.e. odd beginning point

11.5.5 Split and Merge

first: split arc into segments with error sufficiently small
then: merge successive segments if resulting segment sufficiently small error
third: try to adjust breakpoints to obtain better segmentation

repeat: until all three steps produce no further change

11.5.6 Isodata Segmentation

iterative isodata line-fit clustering procedure: determines line-fit parameter
then: each point assigned to cluster whose line fit closest to the point

11.5.7 Curvature

geometry of circular arc segment defined by (r_n, c_n) and (r_{n+1}, c_{n+1})
=====Fig. 11.7=====
$[r(t), c(t)]$: curve represented parametrically, $a < t < b$
$s(t)$: arc length going from $[r(a), c(a)]$ to $[r(t), c(t)]$

$$s(t) = \int_a^t \sqrt{\left(\frac{dr(u)}{du}\right)^2 + \left(\frac{dc(u)}{du}\right)^2} du$$

$$\frac{ds(t)}{dt} = \sqrt{\left(\frac{dr(u)}{du}\right)^2 + \left(\frac{dc(u)}{du}\right)^2}$$

T: unit length tangent vector at $[r(t), c(t)]$ measured clockwise from column axis

$$T = \begin{pmatrix} \frac{dr(t)}{dt} \\ \frac{dc(t)}{dt} \end{pmatrix} \frac{1}{\frac{ds(t)}{dt}}$$

N: unit normal vector at $[r(t), c(t)]$

$$N = \begin{pmatrix} -\frac{dc(t)}{dt} \\ \frac{dr(t)}{dt} \end{pmatrix} \frac{1}{\frac{ds(t)}{dt}}$$

For a planar curve, the curvature at a point on the curve is the limit, as arc length change goes to zero, of the change in tangent angle divided by the change in arc length.
κ: curvature defined at point of arc length s along curve
Δs: change of arc length
$\Delta\theta$: change in tangent angle

$$\kappa(s) = \lim_{\Delta s \to 0} \frac{\Delta\theta}{\Delta s}$$

natural curve breaks: curvature maxima and minima
curvature passes through zero: local shape changes from convex to concave
surface elliptic: when limb in line drawing is convex
surface hyperbolic: when its limb is concave
surface parabolic: wherever curvature of limb zero
cusp singularities of projection: occur only within hyperbolic surface
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 4.14=====

11.6 Hough Transform

Hough transform: method for detecting straight lines and curves on images

Hough transform: template matching

11.6.1 Hough Transform Technique

The Hough transform algorithm requires an accumulator array whose dimension corresponds to the number of unknown parameters in the equation of the family of curves being sought.

Finding Straight-Line Segments

$y = mx + b$: for straight lines does not work for vertical lines
m: slope, b: intercept
$d = r \sin\theta + c \cos\theta$: where row ($r$) and column ($c$) used
d: perpendicular distance from the line to the origin
θ: the angle the perpendicular makes with the x-axis
=====Fig. 11.8=====
use ρ_{min}, θ_{min} and ρ_{max}, θ_{max} to determine if point falls into cell
an accumulator array quantized in this fashion
=====Fig. 11.9=====
Fig. 7.18(a) image with five labeled points
Fig. 7.18(b) each point mapped onto $\rho\theta$ plane as a curved line
Fig. 7.18(c) A: intersection of points 1,3,5; B: intersection of points 2,3,4

Fig. 7.18(d) $A, B, C \Rightarrow C, B, A$ since θ, ρ change sign at $\pm 90°$

=====Gonzalez and Woods, *Digital Image Processing*, Fig. 7.18=====

Fig. 7.19(a) infrared image containing two hangars and a runway

Fig. 7.19(b) thresholded gradient image using Sobel operators

Fig. 7.19(c) linear Hough transform of (b)

Fig. 7.19(d) three accumulator cells with highest count, no gaps ≥ 5 pixels

=====Gonzalez and Woods, *Digital Image Processing*, Fig. 7.19=====

Finding Circles

r: row

c: column

r_0: row-coordinate of the center

c_0: column-coordinate of the center

d: radius

$(r - r_0)^2 + (c - c_0)^2 = d^2$: implicit equation for a circle

circles represented by equations:

$$r = r_0 + d \sin \theta$$

$$c = c_0 + d \cos \theta$$

direction of the gradient at the boundary points of a circle

inward pointing gradients will accumulate evidence for the circle center

=====Fig. 11.11=====

Extensions

generalized Hough transform: to arbitrary shapes specified by boundary points

Variations

Burns line finder: to find straight lines in complex images of outdoor scenes

Burns line finder: uses Hough transform and connected components algorithm

11.6.2 A Bayesian Approach to the Hough Transform

δ: some fixed number related to function of resolution cell size

set of pixels potentially participating in line $r \sin\theta + c\cos\theta = d$:

$$E(\theta, d) = \{(r, c) \in R \times C || r \sin\theta + c\cos\theta - d| < \delta\}$$

=====joke=====

11.7 Line Fitting

procedure for least-squares fitting of line to observed noisy values

(r_n, c_n): unknown values before noise perturbation

$(\hat{r}_n, \hat{c}_n)$: noisy observed values

$\alpha r_n + \beta c_n + \gamma = 0$: line where points (r_n, c_n) lie model for $(\hat{r}_n, \hat{c}_n)$:

$$\hat{r}_n = r_n + \xi_n$$

$$\hat{c}_n = c_n + \eta_n$$

ξ_n, η_n: independent and identically distributed with mean 0 and variance σ^2

$$E[\xi_n] = E[\eta_n] = 0$$

$$V[\xi_n] = V[\eta_n] = \sigma^2$$

$$E[\xi_n \xi_j] = \begin{cases} \sigma^2 & n = j \\ 0 & \text{otherwise} \end{cases}$$

$$E[\eta_n \eta_j] = \begin{cases} \sigma^2 & n = j \\ 0 & \text{otherwise} \end{cases}$$

$$E[\xi_n \eta_j] = 0$$

11.7.1 Variance of the Fitted Parameters

randomness of observed data points in noise: leads to parameter randomness

11.7.2 Principal-Axis Curve Fit

principal-axis curve fit: generalization of line-fitting idea

11.8 Region-of-Support Determination

region of support too large: fine features smoothed out
region of support too small: many corner points or dominant points produced

11.9 Robust Line Fitting

robust line fitting: fit insensitive to a few outlier points
least-squares formulation first, then modify it to make it robust

11.10 Least-Squares Curve Fitting

different approaches and solving methods as follows

11.10.1 Gradient Descent

11.10.2 Newton Method

11.10.3 Second-Order Approximation to Curve Fitting

oblique view of a torus exhibiting folds, cusps, and T-junctions
folds, cusps, and T-junctions: canonical singularities of surface projections
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 4.15=====

11.10.4 Fitting to a Circle

r: row
c: column
(a, b): center of circle
R: radius
circle represented by:

$$f(r, c, a, b, R) = (r - a)^2 + (c - b)^2 - R^2 = 0$$

11.10.5 Variance of the Fitted Parameters

11.10.6 Fitting to a Conic

conic sections: by intersecting a cone with a plane
conic sections: can be circle, ellipse, parabola, hyperbola
=====Jain, *Machine Vision,* Fig. 6.8=====
r: row
c: column
a, b, A, B, C: coefficients
conic represented by:

$$f(r, c, a, b, A, B, C) = A(r - a)^2 + 2B(r - a)(c - b) + C(c - b)^2 - 1 = 0$$

11.10.7 Fitting to an Ellipse
the most common conic fitting: fit to an ellipse

11.10.8 Bayesian Fitting
equal-weight least-squares fit: suitable when all observed points noisy
Bayesian fitting: some observed data points not noisy

11.10.9 Uniform Error Estimation
uniform error estimation appropriate: if all points in observed arc reliable

edge detection purpose: encode and describe information in economical form
======Nalwa, *A Guided Tour of Computer Vision*, Fig. 3.1======
======joke======