ECE 5470 Lecture 13
Image Analysis, Curves, Splines and Polylines

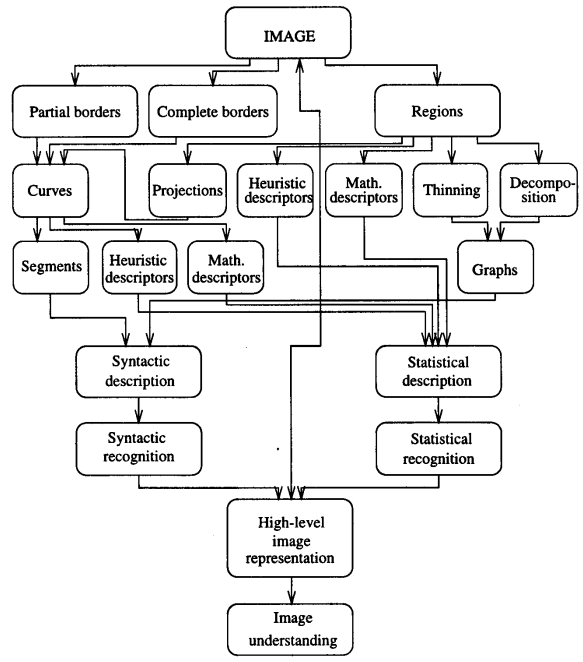
Topics

- Image Analysis
  - What is the big picture
  - How to characterize segments/features
- How to represent boundaries
  - Polylines
  - B-Splines
  - Cubic Splines

Image Analysis

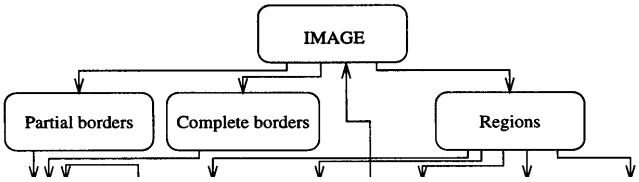
- Image Source
- Image Filtering
- Segmentation
- Feature Extraction
- Feature Classification
- Analysis Results

- Image Sensor Knowledge
- Problem Knowledge: Task Dependent Methods



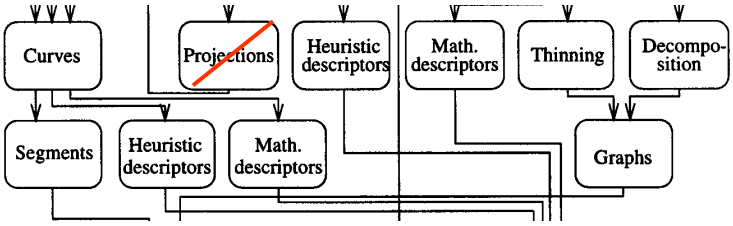
From Sonka, Hlavac and Boyle "Image Processing, Analysis and Machine Vision"

Segmentation



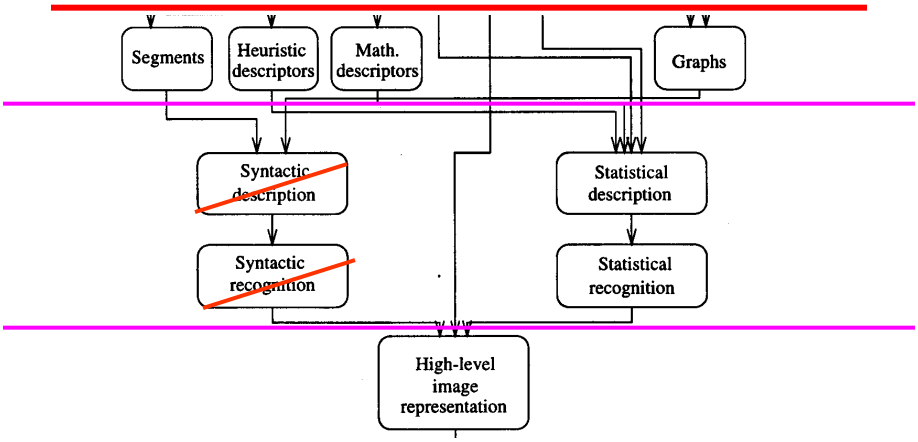
- Partial Borders: Edge operator
- Complete Borders: Boundary Detection
- Regions: Region Growing

Feature Extraction



- Edge segments
- Ad hoc feature descriptors
- Mathematical descriptors
- Graphs (with attributes)

Feature Classification

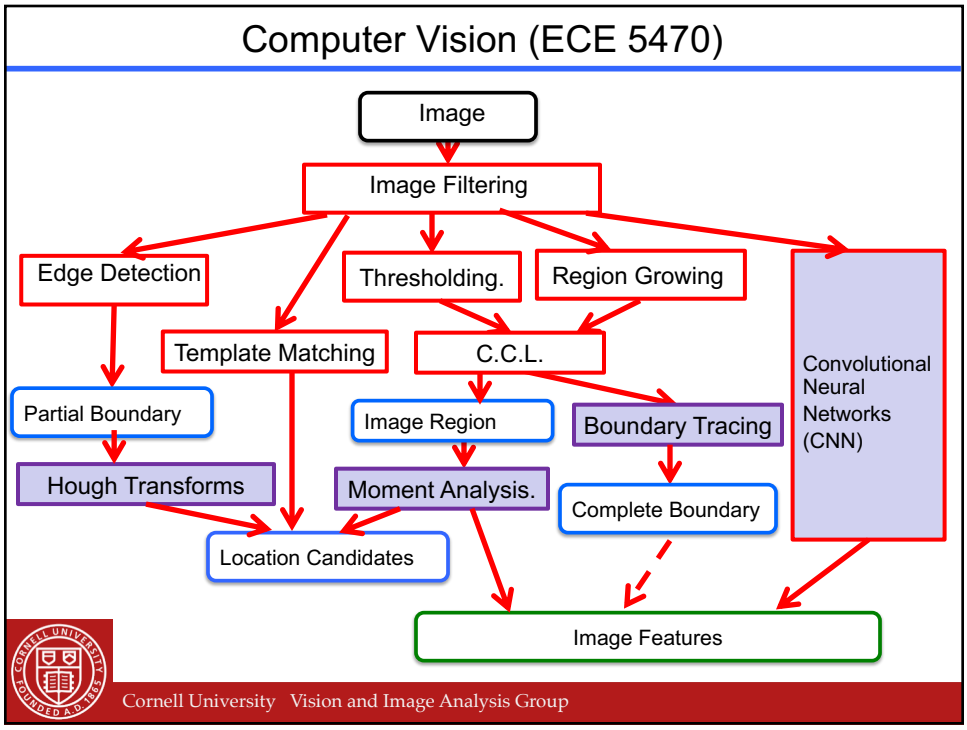


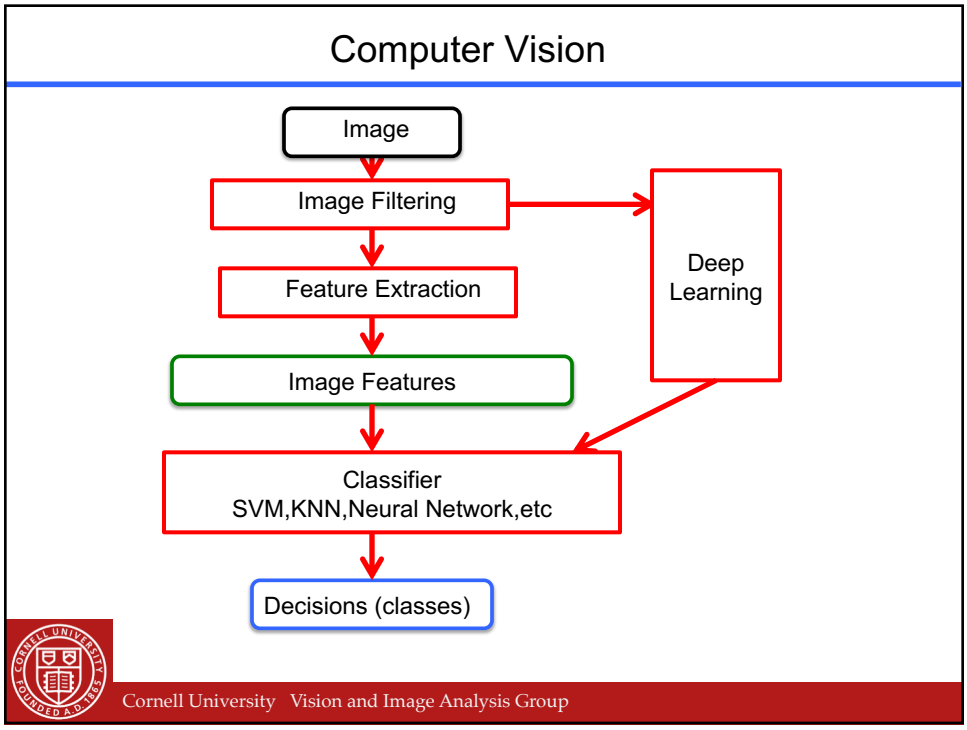
Feature Extraction: Shape Representation

Image Source

1. Image Filtering
2. Segmentation
3. Feature Extraction
4. Feature Classification

Analysis Results


Computer Vision (ECE 5470)
Image
Image Filtering
Edge Detection
Thresholding.
Region Growing
Convolutional Neural Networks (CNN)
Template Matching
C.C.L.
Partial Boundary
Image Region
Boundary Tracing
Hough Transforms
Moment Analysis.
Complete Boundary
Location Candidates
Image Features
Cornell University Vision and Image Analysis Group


Computer Vision
Image
Image Filtering
Deep Learning
Feature Extraction
Image Features
Classifier
SVM,KNN,Neural Network,etc
Decisions (classes)
Cornell University Vision and Image Analysis Group

Shape Representation and Description Taxonomy

- Input representation form:
  Contour based/(multiple) region based
- Object reconstruction ability: Can the original shape be reconstructed from the representation
- Incomplete shape recognition ability:, can the shape be recognized from partial information
- Local/Global description character
- Mathematical/heuristic techniques
- Statistical/Symbolic object recognition
- Robustness/Sensitivity to transformations
  translation rotation and scale

Boundary Representation

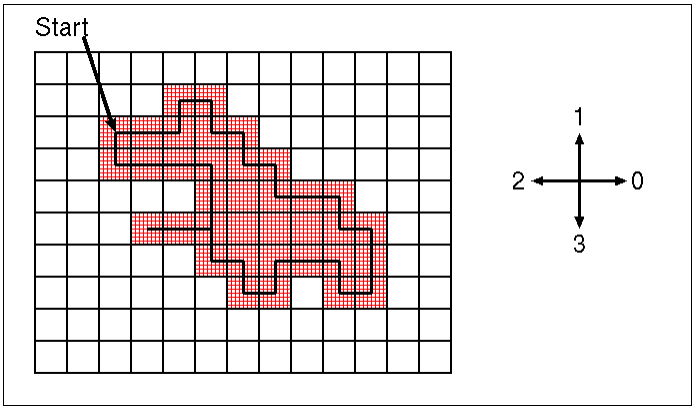
Boundary Representation

- Binary Image Array
- Chain Code
  - efficient boundary encoding
- Polyline
  - approximate boundary with straight line segments
- B-splines, Cubic Splines
  - approximate boundary by parametric curves
- (Fourier Descriptors)

Chain Code

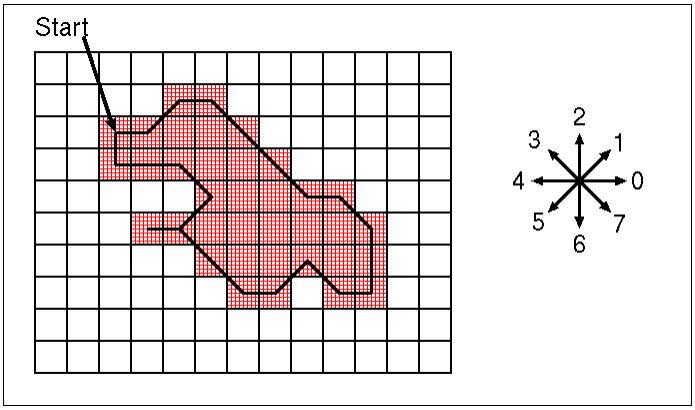
- Trace the boundary of a region using a boundary following algorithm and record the trajectory from pixel to pixel with an efficient direction code.
- Some limited features of the region may be computed directly from the chain code (e.g. area, perimeter length).

Four-Direction Chain code



0,0,1,0,3,0,3,0,3,0,0,3,0,3,3,2,1,2,2,3,2,1,2,1,2,2,0,0
,1,1,2,2,2,1

Eight-Direction Chain Code



0,1,0,7,7,7,0,7,6,6,4,3,5,4,3,3,4,0,1,3,4,4,2

Chain Code

- Use a direction code to specify the sequence of boundary pixels
- Efficient coding of the boundary
  - Requires start location plus code for each step
  - 4-direction code ⇒ 2-bits per code
  - 8-direction code ⇒ 3-bits per code
- For the example image

  4Dir. 34x2 = 68 bits 8Dir. 23x3 = 69 bits

Chain Code

- Input form: pixel boundary
- Object reconstruction ability: exact
- Incomplete shape recognition ability: no
- Local/Global: local
- Mathematical/heuristic techniques: math?
- Statistical/Symbolic: ?
- Transformations translation (S) rotation(S) and scale(S)…Recoding required except for integer pixel translations

! More compact representation than pixel array

Polylines

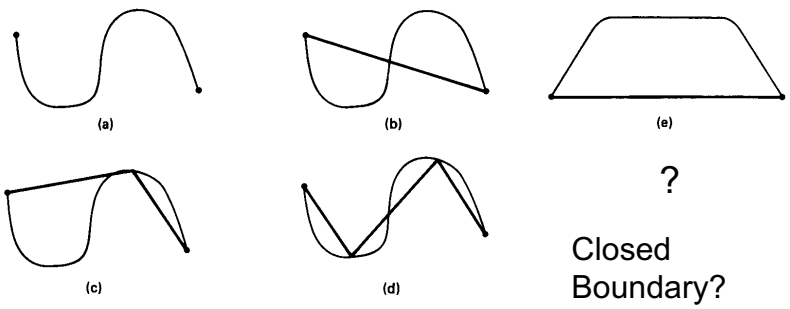
Concept: Approximate a curve or closed contour by a sequence of straight line segments

Problem: How to locate break points to give the best lines? There is no unique solution. Many algorithms have been proposed

- Tolerance Band
- Minimum Length [elastic thread]
- Split or merge algorithms

Split Polyline Algorithm

1. Draw a straight line between the end points
2. Pick the point on the curve farthest from the line. If not within tolerance, use a breakpoint.
3. Repeat 2. Until all curve points are within tolerance.



Polylines

- Input form: pixel boundary
- Object reconstruction ability: partial
- Incomplete shape recognition ability: yes
- Local/Global: local
- Mathematical/heuristic techniques: math?
- Statistical/Symbolic: ?
- Transformations

  translation (S) rotation(S) and scale(S)…

  solves problems of pixel representation

Boundary Representation

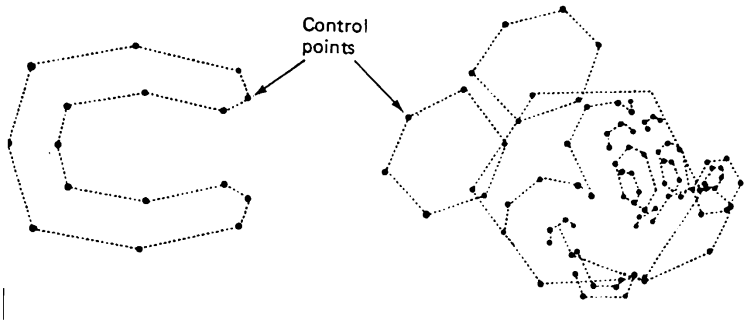
- Straight line approximation - polylines
- Curved line approximation
  - (a) B-Splines
  - (b) Cubic Splines (Pass through knot points)

Curves

Mathematical expressions

- Explicit $y = f(x)$ - not used
- Implicit $f(x,y) = 0$
- Parametric $(x(u)\ y(u))$ n for some parameter u

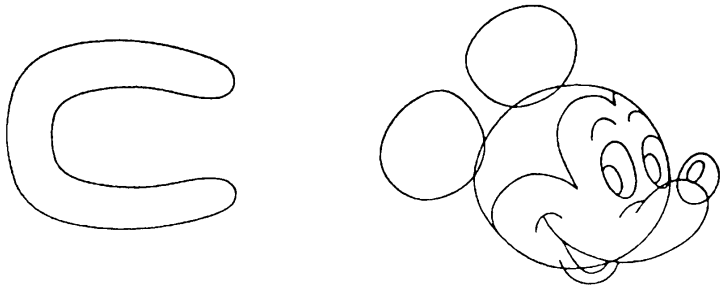
Polyline Representation



16 knot points

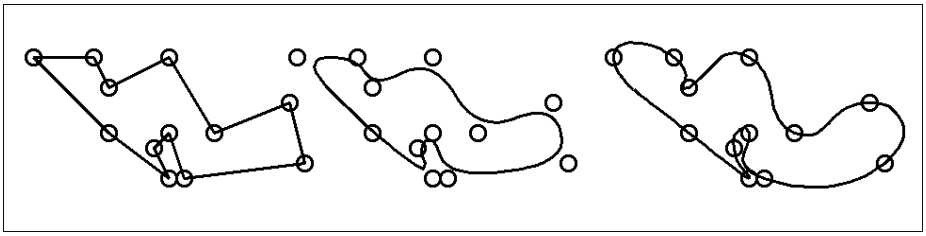
99 knot points

B-Spline Representation

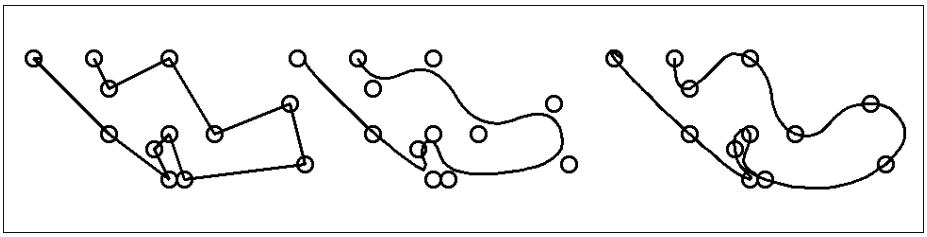


- 1038 points 10536 points
- compression ratio more than 100:1

Splines



Polygon B-Splines Interpolated Splines



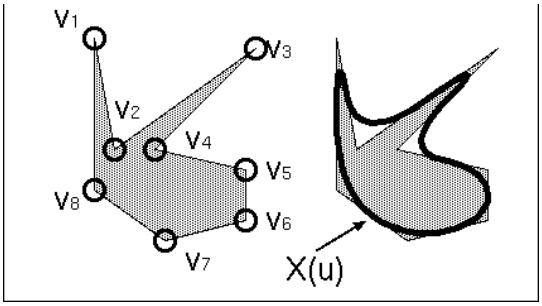
B-Splines

Concept: A curved boundary segment or a closed contour can be represented by a small number of points

B-splines: piecewise polynomial curves that are related to a guiding polygon. Very useful for computer graphics applications

$$x(u) = \sum_{i=1}^{n} v_i B_i(u) \qquad P(u) = (x(u), y(u))$$

B-Splines



$$p(u) = (x(u), y(u))$$

$$x(u) = \sum_{i=1}^{n} v_i B_i(u)$$

B_i are the spline basis funtions

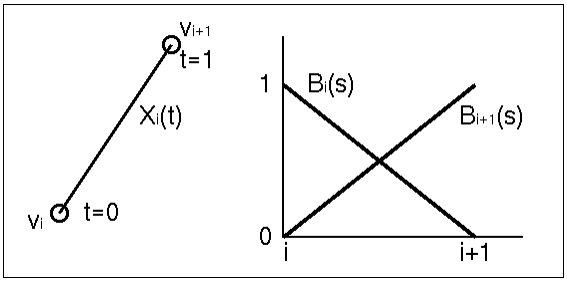
v_i are the vertices of the guiding polygon

x(u) is the b-spline curve

n is the degree of the spline interpolation

Linear Interpolation (n=1)

A line segment



Basis Functions

$$B_i(t) = 1 - t$$
$$B_{i+1}(t) = t$$

- In the span (*i*,*i*+1) $X(s) = v_i B_i(s-i) + v_{i+1} B_{i+1}(s-i)$
- in matrix form

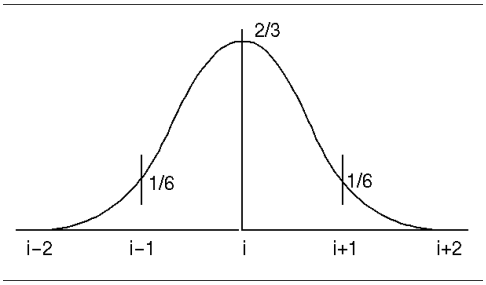
$$X_i(t) = [t \quad 1] L [v_i \quad v_{i+1}]^T \qquad L = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

Cubic B-Splines

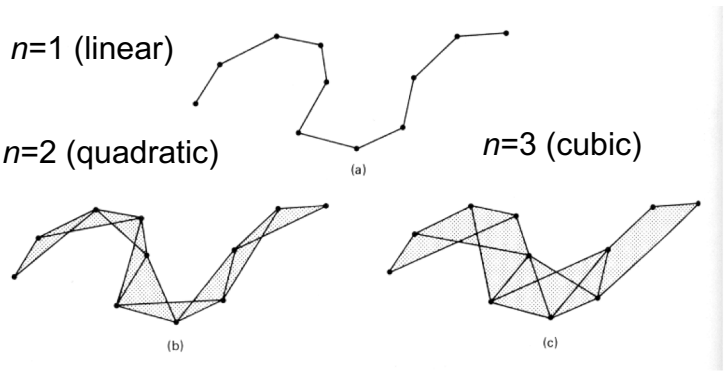
- A Cubic is the lowest order for which the curve can change sign between two points

$$X_i(t) = [t^3 \quad t^2 \quad t \quad 1][C][v_{i-1} \quad v_i \quad v_{i+1} \quad v_{i+2}]^T$$

$$C = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}$$

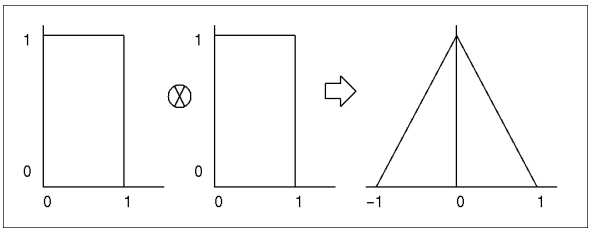


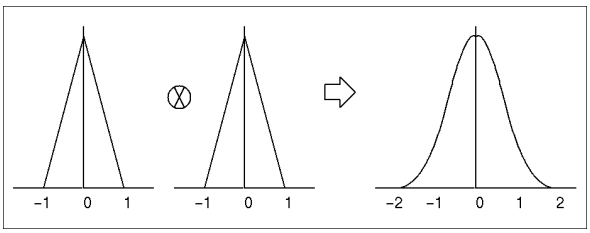
B-Spline Bounds



- The spline of degree *n* must lie in the convex hull formed by consecutive groups of *n* + 1 points.

Spline Functions





Cubic Splines for Image Boundaries

Problem: Polylines are computed from vertices (knot points) that are on the original boundary and B-splines do not pass through the vertices.

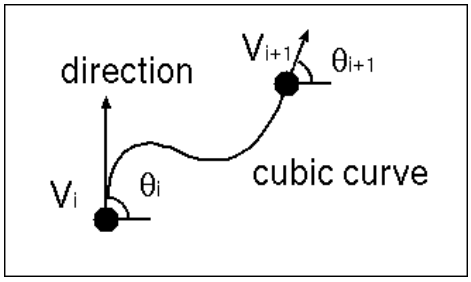
Solution: Define cubic splines that pass through the vertices.

$$P(u) = (x(u), y(u)) = a_0 + a_1 u + a_2 u^2 + a_3 u^3$$

Parameters for Cubic Spline

- Line must pass through V_i and V_{i+1} (4 parameters)
- Line must pass through V_i with direction θ_i and through V_{i+1} with direction θ_{i+1} (2 parameters)

- Two more parameters are required to specify the cubic function.



Curvature

Given a line $p(u) = (x(u), y(u))$

The unit tangent vector is $t(u) = \frac{p'(u)}{|p'(u)|}$

The curvature of the curve is the derivative of the tangent

$$n(u) = p''(u)$$

Consider three points along a curve $p(u + \Delta)$, $p(u)$ and $p(u - \Delta)$. Consider a circle passing through these points. In the limit as $\Delta \rightarrow 0$ the curvature is the inverse of the radius of this osculating circle.

Two more parameters

- The second derivative is not continuous (as is the case for B-splines)
- Minimize for the whole boundary the sum of the square magnitude of difference in the second derivative at the n-1 knots

$$\chi^2 = \sum_{i=1}^{n-1} (\Delta \ddot{P})$$

- where $\Delta \ddot{P} = \ddot{P}_{i-1}(1) - \ddot{P}_i(0)$

Splines

Other Issues: how to deal with abrupt direction changes (straight lines) and the ends of incomplete boundaries.

- Abrupt direction changes (corners) may be achieved by placing more than one knot point at the same location.
- Line ends are treated with modified weights

Splines

- Input form: Knot points
- Object reconstruction ability: partial
- Incomplete shape recognition ability:?
- Local/Global: local
- Mathematical/heuristic techniques: math
- Statistical/Symbolic: ?
- Transformations

  translation (S) rotation(S) and scale(S)…

  solves problems of pixel representation

Polyline and Spline Summary

- Complete and incomplete boundaries can be efficiently represented by polylines and splines
- The polyline and spline representations are not unique.
- How to compare two spline representations?