An Introduction to Handwritten Signature Recognition

Tyler Sheffield

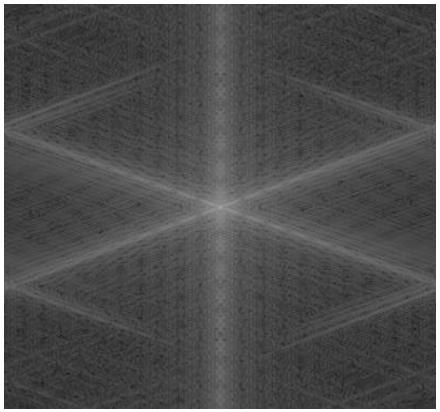
5/31/06

Basics First: Optical Character Recognition

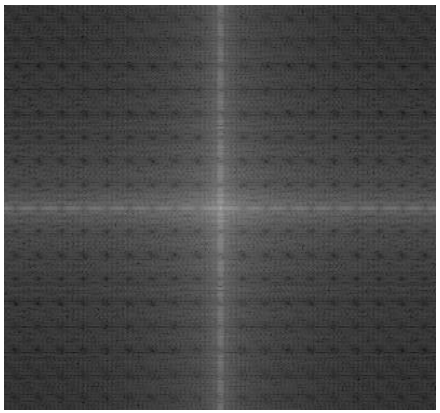
- Noise Reduction
- Binarization
- Segmentation
  - MATLAB has some nice tools to help with this (bwlabel, regionprops)

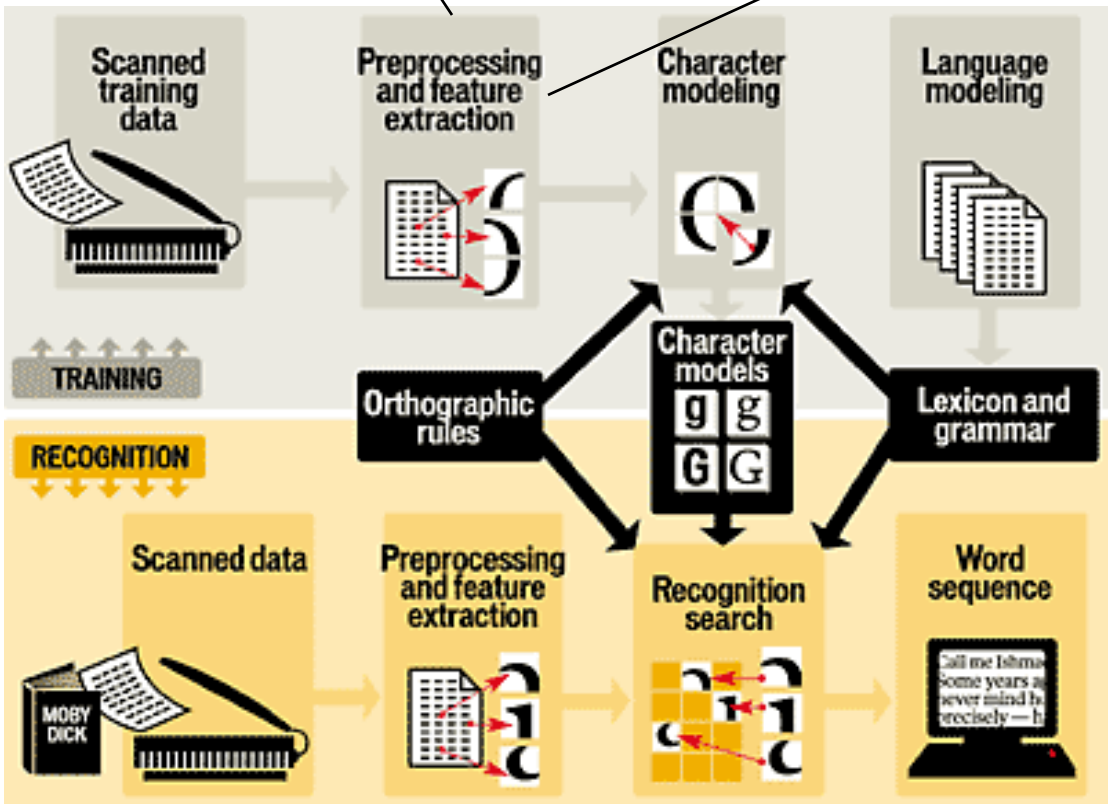
e.g. Fourier Transform

A



E





www.computerworld.com

Many Options for Geometric Feature Extraction

Centroid

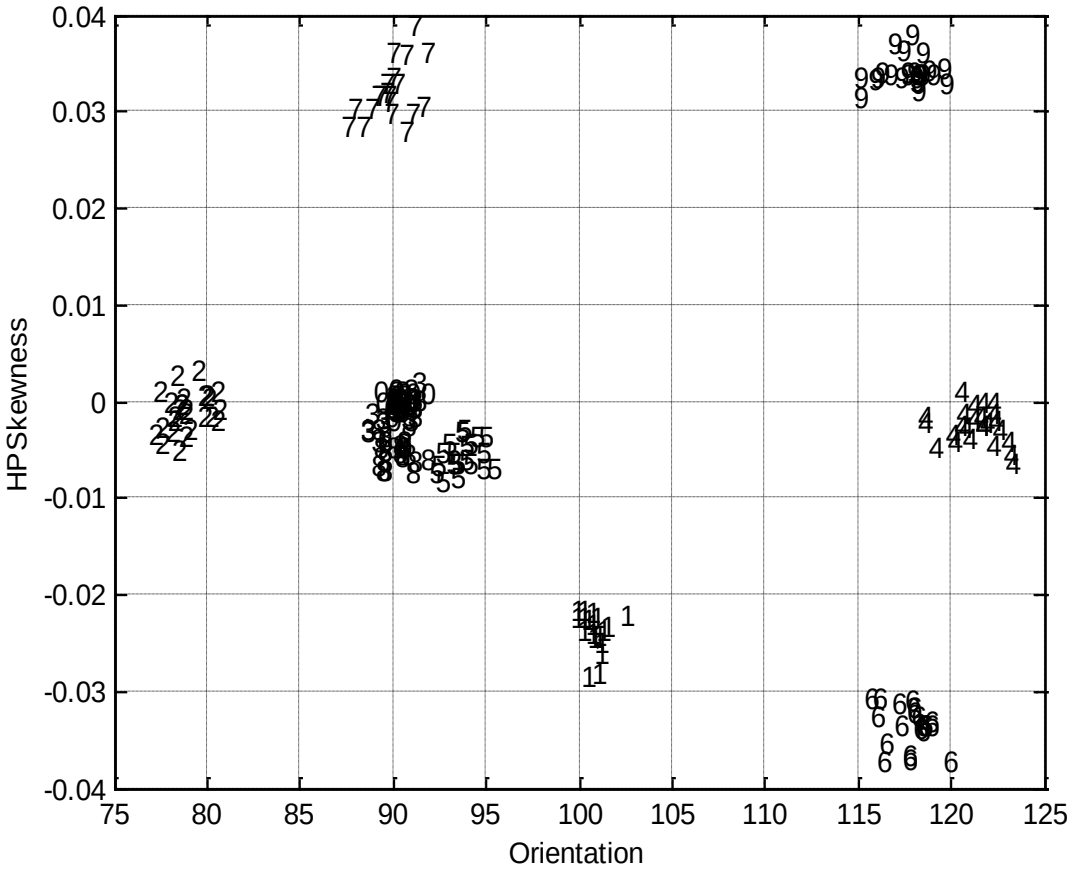
Perimeter → in pixels

Area → in pixels

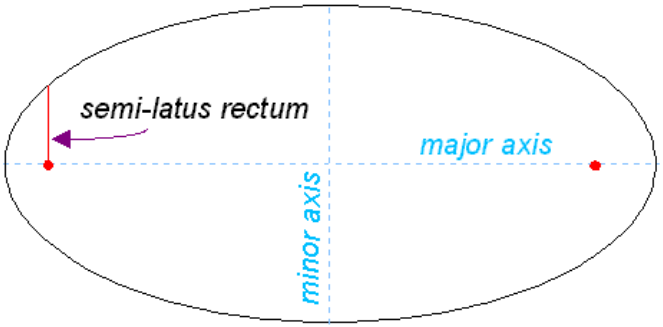
Extent → Area divided by the area of the bounding box

Skewness

$$\gamma_1 = \frac{\mu_3}{\sigma^3}$$



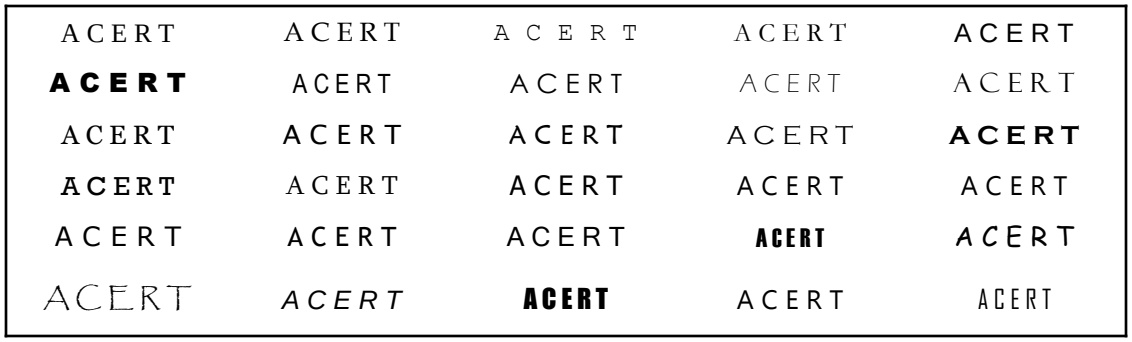
Orientation → angle between x-axis and major axis of ellipse sharing second moment

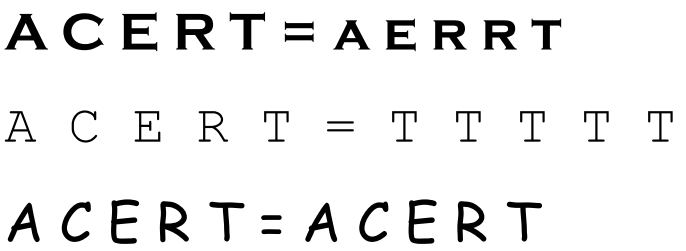


Eccentricity → distance between foci divided by major axis length

http://www.ele.uri.edu/~hansenj/projects/ele585/OCR/

OCR in Action





Text: THIS PAGE CONTAINS INFORMATION REGARDING THE SPECIFICATIONS

Output: THIG PAGG EQNTAING INZQRWAYIDN RSGARDING THG GPGCIFICATIOMG

37 out of 53 correct

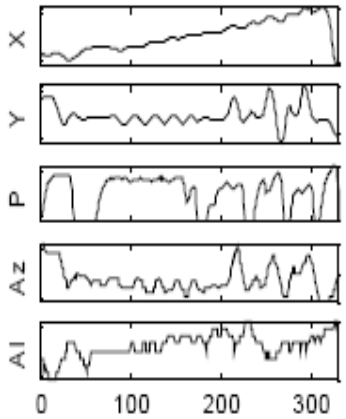
Noisy (crumpled) page results:

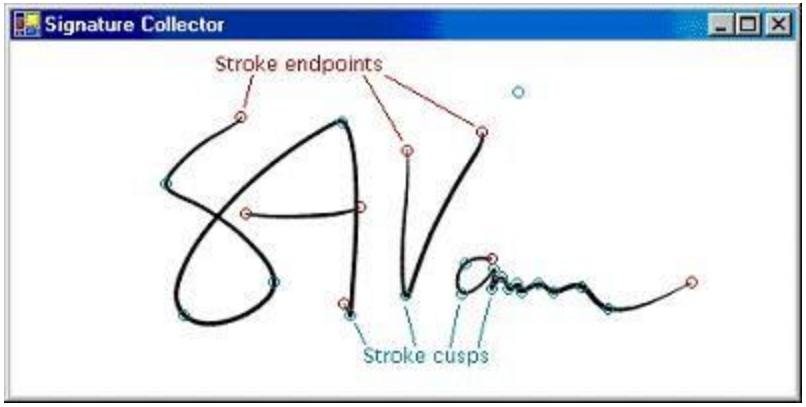
YHIG ORGG SRNTRIMG INFHNWZTIOM QEGDRQINF III IICGOGSIVIEATHIIINS

Signature Recognition as a Biometric

- **FAR**: False Acceptance Ratio
  - measure of how often a forgery passes for the real thing
- **FRR**: False Rejection Ratio
  - measure of how often a genuine signature is rejected by the system
- Human performance:
  - Expert (0.5%FAR @ 7%FRR)
  - Layperson (6.5%FAR @ 26%FRR)

On-Line (Dynamic) Recognition System:





Images and statistics courtesy of https://www.cse.msu.edu/~cse802/Papers/802_Signature_Verification.pdf#search='signature%20verification'

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dntablet/html/tbconinksig.asp

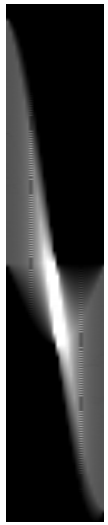
Various Methods of Off-Line Feature Extraction

- Line directionality
  - each pixel associated with vector describing connectivity to pixels around it

- Hough Transform
  - creates ρ by θ matrix from x cos θ + y sin θ = ρ by colinearity detection property
  - very useful in conjunction with tools like *houghpeaks* and *houghlines*

Original signature

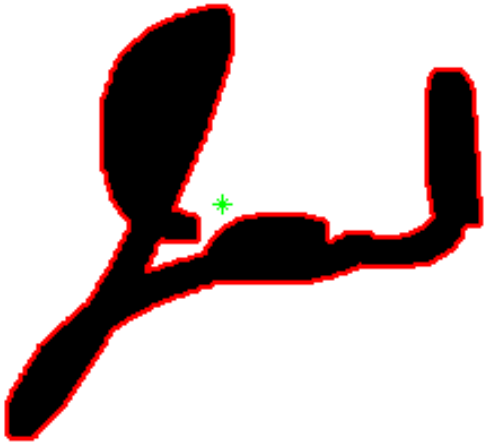
Result of skeletonization



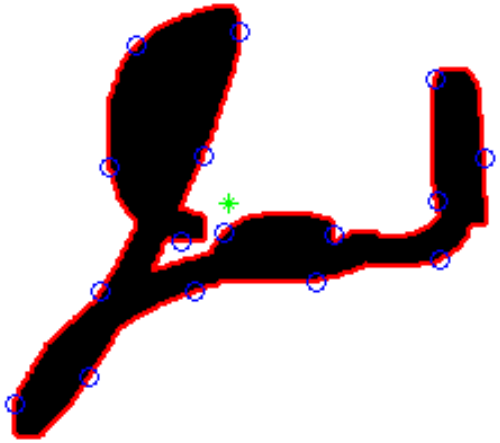
Finding Features of the Signature Outline*

Original signature

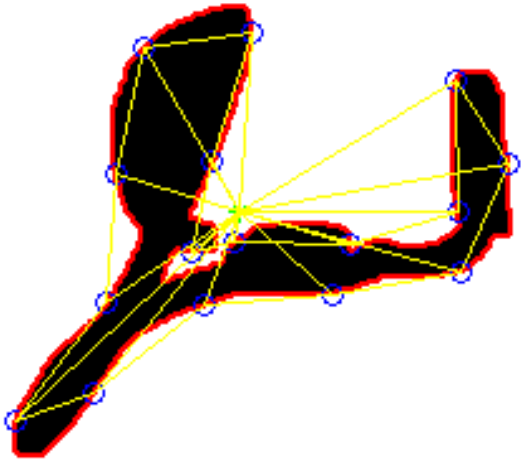
Perform dilation and fill in holes that may remain



Map the contour



Sample the contour points, record angle and distance from centroid



Find triangular masks for pixel counting

Feature vector sequence:

[dr(t) θ(t) A(t)]

* A variation on part of an approach proposed by Miguel A. Ferrer, Jesus B. Alonso, and Carlos M. Travieso in “Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic”

We Have Features, Now What?

- Build database
- Train the system
  - HMM, neural networks, **string matching**

Choose symbols to represent ranges of the feature vector values:

a: 0 < r <= .5 b: .5 < r <= 1 c: 1 < r <= 1.5

A signature's symbol representation might look like this:

444444455555555554444444555555 33
ehcbaaaaabbbbbcdgefbbbbbbbbbacch
szzzzzzzyyxwwwwwwvvvvvvuutsssss s

There are a variety of ways to compare strings, such as finite automata

- Acquire and classify sample data

Performance of Simple Signature Recognition System Using String Matching

Number of contour samples = 16
Number of symbols per feature = 8

Genuine Signatures

	Accept	Reject
Sig 10	4	4
Sig 11	5	3
Sig 13	6	2
Sig 20	6	2
Sig 21	6	2
Sig 23	5	3
Sig 24	4	4
Sig 30	6	2

Confusion matrix for classification of signatures

	Sig 10	Sig 11	Sig 13	Sig 20	Sig 21	Sig 23	Sig 24	Sig 30
Sig 10	**5**		**2**		**1**			
Sig 11		**8**						
Sig 13			**8**					
Sig 20				**6**	**2**			
Sig 21	**1**				**7**			
Sig 23						**6**	**2**	
Sig 24				**3**			**5**	
Sig 30						**1**		**7**

Results:

FAR = 27%
FRR = 34%

Classification values derived from function *strsimilarity*, provided in the textbook's DIPUM library