

Word Matching of handwritten scripts



MS 206
Hebrew square book script, Iraq, 1st half of 11th c.

Seminar about ancient document analysis

Introduction

- Contour extraction
- Contour matching
- Other methods
- Conclusion
- Questions

Problem

Text recognition in handwritten historical documents:

- Traditional handwriting recognizers based on OCR do not perform well
 - Problem of noise
 - faded ink
 - weak strokes

Idea

- Instead of single character matching, the whole word is matched
 - word patterns are stored in a data base
 - training set needed (or interactive teaching)
 - recognized words are indexed by a matching process
- The matching is done on contour features
 - amount of concavity and convexity at different scale levels.

Contour extraction

Preprocessing step of the matching process

- binarization of the image
- Position estimation
- Labeling of components
- Creating the word contour

Preprocess: Binarization

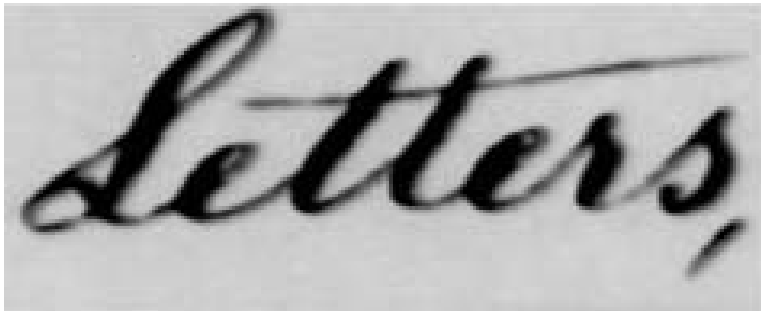
- Local threshold using Sauvola's method

$$T = \mu \left(1 - k \left(1 - \frac{\sigma}{R} \right) \right)$$

where as μ is the mean and σ is the standard deviation of a local area centered by the pixel ($k = 0.02$ and $R = 128$)

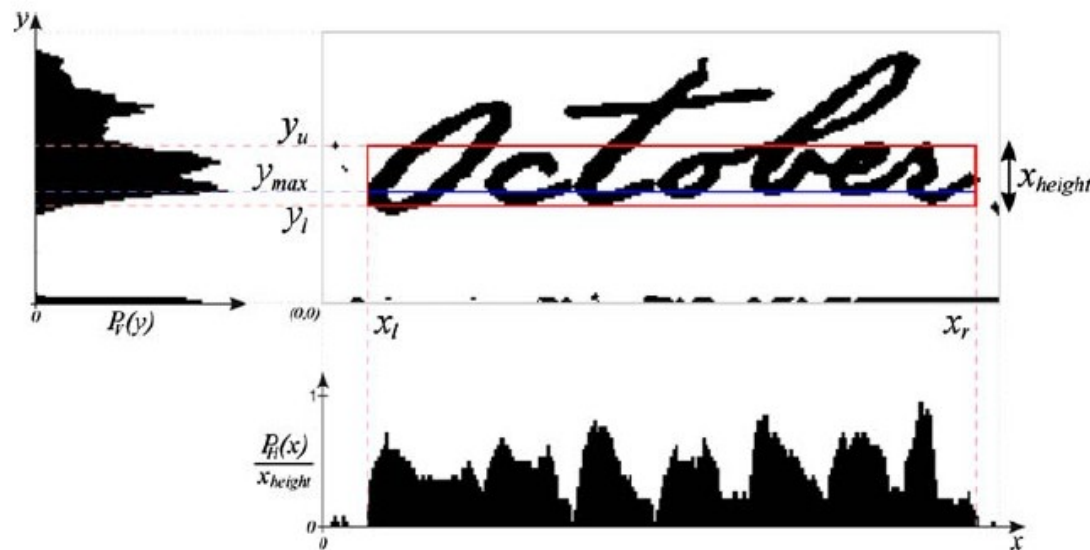
Preprocess: Binarization

- Other possibility using morphological operators
 - **Dynamically thresholding** is done on an opened image, where as the parameter are estimated on the closed image
 - weak strokes and connections between letters are detected and dilated.



Preprocessing: Position estimation

- Position estimation is important to connect disconnected parts later
- Interested part is the main body of the lower letters



Preprocessing: Connecting

Connecting the components is done in 2 steps

- Labeling

- 8-neighborhood connectivity

- ignoring the components
with $\beta = 0.1$

$$\beta x_{height}^2 < P_A(c)$$

- This will remove punctuations and diacritic signs,
therefore not usable for languages with lots of
diacritic signs

Preprocessing: Connecting

- **Connecting components**
(done by adding synthetic lines, since most disconnections occur between letters)
 - Sorting the components by the horizontal position of center of gravity
 - generating possible links between the closest pixels of the two components
 - choosing the best valid link.

Preprocessing: Connecting

- Valid links are:
 - Both ends of the link are strictly inside the main body of the lowercase letters. They must meet following condition:

$$y_1 < y < (y_u - \alpha_c x_{height}) \quad \text{where } \alpha_c \text{ is set empirically to } 0.2$$

- Both ends of the link are considerably above the main body of lowercase letters. Therefore it has to meet following condition:

$$y > 2y_u - y_1$$

Contour Matching

Following techniques are proposed:

- MCC (multiscale convexity concavity)
- MCC-DCT (MCC using concrete cosine transf.)
- Chaincode
- Feature based methods

Contour Matching: MCC

MCC stores the amount of convexity/concavity on different scale levels for each point.

Representation as Matrix $M(\sigma, u)$, where σ is the scale and u is the contour point

Concave and convex parts are distinguished by the sign.

Contour Matching: MCC

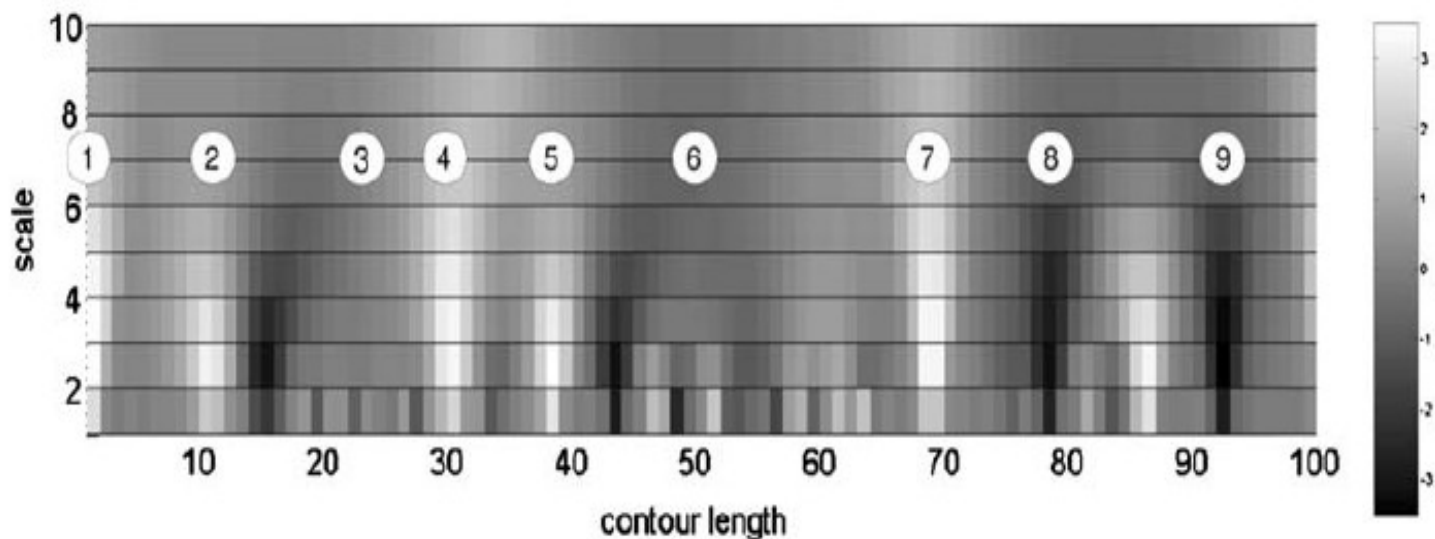
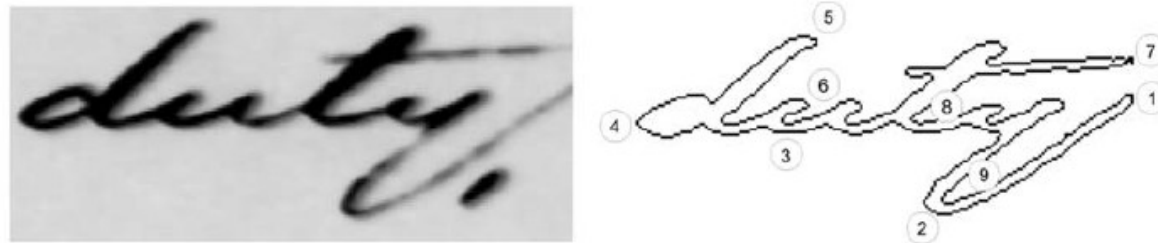
The evaluation process generates the matrix M.

- Running on the curve C with N contour points
- parameterized by arc length u: $C(u) = (x(u), y(u))$, where $u \in [0, N]$.
- convolution of Contour C with Gaussian kernel Φ_σ of width $\sigma \in \{1, 2, \dots, \sigma_{\max}\}$.

$$x_\sigma(u) = \int x(t) \phi_\sigma(u-t) dt$$
$$y_\sigma(u) = \int y(t) \phi_\sigma(u-t) dt$$

$$\phi_\sigma(u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{u^2}{2\sigma^2}}$$

Contour Matching: MCC

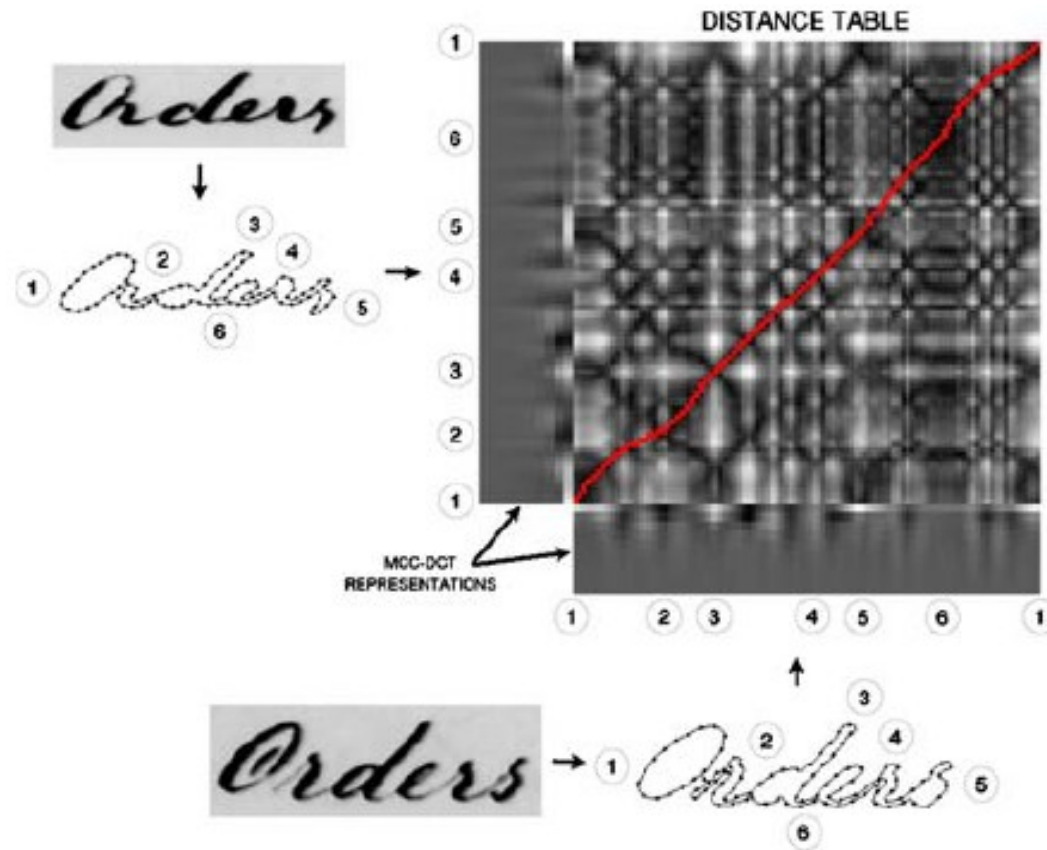


Matching Process

- Generation of a distance table
 - size $N \times N$
 - each cell $d_{j,i}$ stores the distance between the point i of the first contour and point j of the second contour
 - $f_{\sigma u}$ is the matrix for one contour ($\sigma_{\max} \times N$)
 - P_{σ} is a weight to control the relative proportions between distances computing different scales

$$d(u_A, u_B) = \sum_{\sigma=1}^{\sigma_{\max}} P_{\sigma} \frac{|f_{\sigma u_A}^A - f_{\sigma u_B}^B|}{r_{\sigma}^A - r_{\sigma}^B} \quad r_{\sigma} = \max_u \{f_{\sigma u}\} - \min_u \{f_{\sigma u}\}$$

Matching Process



Other approaches

- Contour matching with MCC-DCT
 - MCC-DCT uses the 1D cosine transformation
 - Closely related to the MCC method
 - The adaptation of the weights P_σ is much simpler
- Other holistic approaches
 - Feature based (Scalar and profile features)

Conclusions

Advantages

- The proposed method can be used for other problems with multishaped objects
- easier to match handwritten words than single characters

Disadvantages

- large data base needed for words
- not optimal approach for languages using lots of diacritic signs.

Questions?



Thank you for your interest.