A LINE-BASED REPRESENTATION FOR MATCHING WORDS
CONTENTS

- Introduction
- Line-based Word Representation
- Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- Redif Extraction Task: Redif Extraction using Contour Segments (RECS)
- Experimental Results
- Conclusion
The excessive number of documents in digital environment has increased →
- Efficient access to historical documents
- Word Spotting techniques, alternative to OCR systems based systems
- Poor quality of historical documents, and variety of scripts
INTRODUCTION

RELATED STUDIES

- Manmatha et al. (1996, 2003a, 2003b)
  - Projection profiles, word profiles, and background/ink transitions
  - Dynamic time warping (DTW)
- Rath and Manmatha, 2007
  - Employ clustering to recognize words
INTRODUCTION

RELATED STUDIES

- Adamek et al., 2007
  - Contour-based approach to match the words
  - Contours are extracted after several processes, including binarization, and removing artifacts.
  - Multi-scale convexity concavity (MCC) representation with DTW
INTRODUCTION

RELATED STUDIES

- Ataer and Duygulu, 2006
  - Extract interest points from word images using scale invariant feature transform (SIFT) operator (Lowe, 2004)
  - A codebook obtained by the vector quantization of SIFT descriptors is then used to represent and match the words.
  - The method is tested on Ottoman documents
INTRODUCTION

APPROACH

- Line-based representation
- Two matching criteria
  - Word matching task
    - Word Image matching using Line Descriptors (WILD)
  - Redif Extraction task
    - Redif Extraction using Contour Segments (RECS)
INTRODUCTION

CONTRIBUTION

- Effective and efficient representation of words images based on line descriptors
- Two approaches for word matching and *redif* extraction task (WILD and RECS)
- Tested on different languages; English and Ottoman,
- Provides promising results without the need of pre-processing steps in word matching task
- Multi-scale analysis in word matching task
- A pioneering image-based automatic *redif* extraction method, first in the literature (RECS)
Introduction

Line-based Word Representation
- Binarization
- Extraction of Contour Segments
- Line Approximation
- Line Description

Word Matching Task: Word Image Matching using Line Descriptors (WILD)

Redif Extraction Task: Redif Extraction using Contour Segments (RECS)

Experimental Results

Conclusion
LINE-BASED WORD REPRESENTATION

 Motivation
  • Words consist of lines and curves
  • Success of using line segments as descriptors for object recognition (Ferrari et al., 2008)

 Words are described using line segments extracted from the contours of images
LINE-BASED WORD REPRESENTATION

- BINARIZATION

![Image of binarization example]
LINE-BASED WORD REPRESENTATION

- EXTRACTION of CONTOUR SEGMENTS
  - Connected components → 8-neighbors
LINE-BASED WORD REPRESENTATION

- LINE APPROXIMATION
  - Fit lines to the points of the contour segments, instead of using the contour itself.
  - Line approximation is performed using Douglas-Peucker\(^1,2\) algorithm
  - The parameter \(\tau\), as approximation accuracy

\(^1\) Douglas and Peucker, 1973,
\(^2\) Hershberger and Snoeyink, 1992
LINE-BASED WORD REPRESENTATION

- LINE APPROXIMATION

[Image showing line-based word representation with examples]
LINE-BASED WORD REPRESENTATION

LINE DESCRIPTION

- A line is described using the position, orientation, and length information as in (Ferrari et al., 2008)
- \( l = \{p_s, p_m, p_e, \theta, \rho\} \)
Introduction

Line-based Word Representation

Word Matching Task: Word Image Matching using Line Descriptors (WILD)
- Line Matching
- Word Matching

Redif Extraction Task: Redif Extraction using Contour Segments (RECS)

Experimental Results

Conclusion
Word Matching Task: Word Image matching using Line Descriptors (WILD)

- Each word image $I$ is represented as a set of line descriptors as:
  - $I = \{l_1, l_2, \ldots, l_N\}$, $N$ is the number of line descriptors.
Word Matching Task

- Representative points of each line descriptor are re-arranged as:

\[
X_l = \frac{\sum x^i_m}{N}, \quad Y_l = \frac{\sum y^i_m}{N}, \quad i = 1 \ldots N,
\]

- We prefer to use only the mid-point and refer to it as \( r \).
Word Matching Task

- **LINE MATCHING**
  - The distance between two line descriptors \( \ell_a \) and \( \ell_b \) are computed as:
  - \[ d(\ell_a, \ell_b) = 4d_r + 2d_\theta + d_\ell \]
  - where \( d_r = |r_a - r_b| \), \( d_\theta = |\theta_a - \theta_b| \),
  - and \( d_\ell = |\log(\rho_a, \rho_b)| \)
  - \( r_a \) and \( r_b \) → mid-points of the line descriptors
  - \( \theta_a \) and \( \theta_b \) → orientations of the line descriptors [0,\( \Pi \)]
  - \( \rho_a \) and \( \rho_b \) → lengths of the lines
WORD MATCHING

- Based on the distances between line descriptors
- $I_a$ and $I_b$ are the word images having $N_a$ and $N_b$ line descriptors
- For each line descriptor in $I_a$, we search for the best matching line in $I_b$.

That is; $(\ell^a_i, \ell^b_j)$ is a matching pair, if $d(\ell^a_i, \ell^b_j) < d(\ell^a_i, \ell^b_k) \forall k, j \neq k, k = 1, 2, ..., N_b.$
WORD MATCHING

If two or more line descriptors in $I_a$ match to a single line in $I_b$ then we choose the one with the minimum distance and eliminate the others.

For example; $I^a = \{\ell^a_1, \ell^a_2, \ell^a_3\}$ $I^b = \{\ell^b_1, \ell^b_2, \ell^b_3, \ell^b_4\}$

the minimum matches are $\{(\ell^a_1, \ell^b_3), (\ell^a_2, \ell^b_2), (\ell^a_3, \ell^b_2)\}$,

$$D_{a,b} = d(\ell^a_1, \ell^b_3) + \min(d(\ell^a_2, \ell^b_2), d(\ell^a_3, \ell^b_2)).$$
Word Matching Task

- WORD MATCHING
  - To compute the final score
    - In addition to distance of matched pairs of line descriptors
    - The number of hits $h_{a,b}$ as the number of matches between two images
    - The number of line descriptors in the images $N_a$ and $N_b$

$$f(I^a, I^b) = (D_{a,b}) \left( \frac{(N_a - h_{a,b})^2 + (N_b - h_{a,b})^2}{\sqrt{[(N_a)^2 + (h_{a,b})^2][(N_b)^2 + (h_{a,b})^2]}} \right)$$
Word Matching Task

- WORD MATCHING
  - a global distance matrix $F(Q \times Q)$
  - $F(a,b) = f(I_a, I_b)$, $Q$ is the number of images in test bed
  - $F(1,3)$ is the dissimilarity value between the first and third image in the data set.
Introduction

Line-based Word Representation

Word Matching Task: Word Image Matching using Line Descriptors (WILD)

Redif Extraction Task: Redif Extraction using Contour Segments (RECS)

Experimental Results

Conclusion
In Ottoman (Divan) poetry, most of the poems are based on a pair of lines, couplet or distich. Distich: two hemistichs (lines) Hemistichs of the same distich completes each other The rhyme and redif are used to provide the integrity of the distichs of a poem and provide a melody to its voice. The redif can be explained as the repeated patterns following the rhyme in a poem.
<table>
<thead>
<tr>
<th>1.1</th>
<th>Seni seyr itmek için reh-güzer-i gülşende</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>İki cânibde durur serv-i hîrâmân saf saf</td>
</tr>
<tr>
<td>2.1</td>
<td>Mescid içre göre tâ kimlere hemzânûsîn</td>
</tr>
<tr>
<td>2.2</td>
<td>Şekl-i sakkada gezer dide-i gîryân saf saf</td>
</tr>
<tr>
<td>3.1</td>
<td>Gökde efgân iderek sanma geçer hayl-i küleng</td>
</tr>
<tr>
<td>3.2</td>
<td>Çekilûr kûyune mürgân-î dil ü cân saf saf</td>
</tr>
</tbody>
</table>
Redif Extraction Task

- In the method, we automatically extract the *redifs* in the handwritten literary Ottoman text images.
- Un-segmented images are used.
- Unlike word spotting studies, the most basic unit is not the word image but contour segment \( C \) which may represent a word, a character, and a sequence of characters.
Redif Extraction Task

- Method can be summarized as:
  - Normalization of line descriptors with reference line

\[
X_C = \frac{\sum x_m^i}{n}, \quad Y_C = \frac{\sum y_m^i}{n}, \quad i = 1, 2, \ldots, n
\]

\[
\ell^r = (p_m^r, \theta^r, \rho^r)
\]

\[
\ell' = (p_m', \theta', \rho') \quad x_m' = x_m - x_m^r
\]

\[
y_m' = y_m - y_m^r
\]

\[
\theta' = \theta - \theta^r
\]

\[
\rho' = \rho / \rho^r
\]
Redif Extraction Task

- Contour segment descriptors are defined as:
  \[ C' = \{ \ell'_1, \ell'_2, \ldots, \ell'_n \} \]

- Construction of codebook \( B \):
  \[ B = \{ b_1, b_2, \ldots, b_k \} \]

- Represent contour segment descriptors as a sequence of elements of codebook:
  \[ C'' = \{ b_{\ell'_1}, b_{\ell'_2}, \ldots, b_{\ell'_n} \} \]
  where \( b_{\ell'_i} \in B \), \( b_{\ell'_i} \) is the code of \( \ell'_i \), and \( i = 1, 2, \ldots, n \).
Redif Extraction Task

- We compute the distance between two contour segment descriptors each consisting of elements of the codebook.
  - The difference is the sum of insertions, deletions, and substitutions.

\[ C'_i = \{ b_1, b_2, b_3, b_4 \} \]
\[ C'_j = \{ b_1, b_3, b_3 \} \]

- Second code of the \( C'_i \) should be deleted
- \( b_4 \) should be substituted with \( b_3 \)
Redif Extraction Task

- REDIF EXTRACTION
- Rules of the redif
- A redif must appear
  - At the end of the second hemistich –line- of a distich –couple- (constraint 1)
  - In every distich (constraint 2)
Redif Extraction Task

- REDIF EXTRACTION
  - According to constraint 1
    - The x positions of the redifs should roughly be the same and they should be close to the left border (end of the last hemistich)
    - We eliminate the contour segments that do not appear in the left (last) part of the distichs
      - $\alpha_1(w) < X$, $w$ is the width of the image $\alpha_1$ in $[0,1]$
    - Among the remaining ones, a contour segment and its matches are need to be vertically aligned to be counted as a redif
    - For a contour segment, we check each of its matches whether they are vertically aligned. We ignore the rest of the matches for the segment if not.
Redif Extraction Task

- **REDIF EXTRACTION**
  - According to constraint 1
    - Two contour segments are vertically aligned;
      - If the distance in x positions $\alpha_2(w)$, $\alpha_2$ in [0,1]
    - $\alpha_1$ and $\alpha_2$ are empirically determined, 0.25 and 0.15
  - We check the remaining contour segments and their matches, we check the number of matches for each remaining contour segment to satisfy constraint 2.
  - Minimum number of matches should be 5

12/24/2009
Redif Extraction Task

- We search for the contour segments that have one or more common matches and take the union of the matches of those contour segments, and we perform this operation until any pair of contour segments has a common match.

- Having combined the common matches not to extract the same contour segment as *redif*, we check whether the contour segment extracted as *redif* has more than five matches, and if so we count that contour segment as *redif*
Redif Extraction Task

\[ C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}, C_{50}, C_{36}, \ldots \} \]
\[ C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}, C_9, C_{17}, \ldots \} \]
\[ C_3 \rightarrow \{ C_3, C_{12}, C_{20}, C_{30}, C_{35}, C_{38}, C_{46}, \ldots \} \]
\[ C_4 \rightarrow \{ C_4, C_5, C_{12}, C_{41}, C_{42}, C_{43}, \ldots \} \]
\[ C_7 \rightarrow \{ C_7, C_{15}, C_{25}, C_{34}, C_{42}, C_{49}, C_{24}, C_{32}, \ldots \} \]

\[ \ldots \]
\[ C_{18} \rightarrow \{ C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}, C_{50}, \ldots \} \]
\[ C_{19} \rightarrow \{ C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}, C_{17}, \ldots \} \]

\[ \ldots \]
Redif Extraction Task

According to constraint 1

- We do not consider the contour segment descriptors like \( C_{50}, C_{26}, C_9, \ldots \) as a redif candidate. Since they do not satisfy
  - \( X < \alpha_1(w) \)
- After checking contour segments should be vertically aligned
  - \( \alpha_2(w) \)
The candidate list turns out to be

\[ C_1 \rightarrow \{ C_1, C_{10}, C_{18}, C_{28}, C_{44}, C_{50}, C_{56}, \ldots \} \]
\[ C_2 \rightarrow \{ C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}, C_9, C_{17}, \ldots \} \]
\[ C_3 \rightarrow \{ C_3, C_{12}, C_{20}, C_{30}, C_{35}, C_{38}, C_{46}, \ldots \} \]
\[ C_4 \rightarrow \{ C_4, C_5, C_{12}, C_{41}, C_{42}, C_{43}, \ldots \} \]
... 
\[ C_7 \rightarrow \{ C_7, C_{15}, C_{25}, C_{34}, C_{42}, C_{49}, C_{24}, C_{32}, \ldots \} \]
... 
\[ C_{18} \rightarrow \{ C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}, C_{50}, \ldots \} \]
\[ C_{19} \rightarrow \{ C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}, C_{17}, \ldots \} \]
...
Then we search for the contour segment descriptors having common matches, and take the union of them.

\[ C_1 \rightarrow \{C_1, C_{10}, C_{19}, C_{28}, C_{44}\} \text{ and } C_{18} \rightarrow \{C_{18}, C_{10}, C_1, C_{28}, C_{36}, C_{44}\} \]

\[ C_2 \rightarrow \{C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}\} \text{ and } C_{19} \rightarrow \{C_{19}, C_{37}, C_{45}, C_{29}, C_2, C_{11}\} \]

then we have

\[ C_1 \rightarrow \{C_1, C_{10}, C_{19}, C_{28}, C_{44}, C_{36}\} \]

\[ C_2 \rightarrow \{C_2, C_{11}, C_{19}, C_{29}, C_{37}, C_{45}\} \]

\[ C_3 \rightarrow \{C_3, C_{12}, C_{20}, C_{30}\} \]

\[ C_4 \rightarrow \{C_4, C_5, C_{12}, C_{41}\} \]
Redif Extraction Task

- According to constraint 2
  - Each candidate should have at least 5 or more matches
    - Then we only have $C_1$ and $C_2$
    - $R = \{C_1, C_2\}$
Introduction

Line-based Word Representation

Word Matching Task: Word Image Matching using Line Descriptors (WILD)

Redif Extraction Task: Redif Extraction using Contour Segments (RECS)

Experimental Results

- Data sets
- Evaluation Criteria
- Results

Conclusion
EXPERIMENTAL RESULTS

- Word Matching
  - DATA SET
    - Images from George Washington (GW) Collection in Library of Congress, which is used as a benchmark data set in word spotting studies
      - Ten pages of GW (GW10)
        - 2381 words
      - Twenty pages of GW (GW20)
        - 4860 words
EXPERIMENTAL RESULTS

- **Word Matching**
  - **DATA SET**
    - Ottoman sets used in Ataer and Duygulu, 2006 - 2007
      - Three pages – 257 words OTM1
      - Six pages – 823 words OTM2
      - Combination of OTM1 and OTM2 (OTM1+2)
EXPERIMENTAL RESULTS

- **Word Matching**
  - **EVALUATION CRITERIA**
    - Retrieval
      - trec_eval package – precision-recall
    - Recognition
      - Word Error Rate

\[
WER = 1 - \left( \frac{\text{#correct matches in test page}}{\text{#words in test page}} \right)
\]
EXPERIMENTAL RESULTS

- Redif Extraction
  - DATA SET
    - 100 poems from
      - 15th to 19th centuries
    - Turkey Manuscripts¹
    - Ottoman Text Archive Project (OTAP)²

¹ www.yazmalar.gov.tr
² courses.washington.edu/otap/
EXPERIMENTAL RESULTS

- Redif Extraction
  
  EVALUATION CRITERIA

\[
R : \text{extracted redifs} \\
R_{gt} : \text{ground truth for redifs} \\
ER : \text{extraction rate } \in [0, 1] \\
ER = \frac{\# \text{correct extractions in } R}{\max(\text{sizeof}(R_{gt}), \text{sizeof}(R))}
\]
**EXPERIMENTAL RESULTS**

- **Word Matching**

(GW data sets)

<table>
<thead>
<tr>
<th>December</th>
<th>Instruction</th>
<th>should</th>
<th>1755</th>
</tr>
</thead>
<tbody>
<tr>
<td>December</td>
<td>Instruction</td>
<td>would</td>
<td>1755</td>
</tr>
<tr>
<td>Dec. Decem.</td>
<td>Instructions</td>
<td>I could</td>
<td>1755</td>
</tr>
<tr>
<td>December</td>
<td>Instructions</td>
<td>should</td>
<td>1755</td>
</tr>
<tr>
<td>December</td>
<td>Instructions</td>
<td>I have</td>
<td>1755</td>
</tr>
<tr>
<td>December</td>
<td>Instructions</td>
<td>would</td>
<td>1755</td>
</tr>
<tr>
<td>December</td>
<td>Alexandria</td>
<td>would</td>
<td>1755</td>
</tr>
<tr>
<td>Recru.</td>
<td>Instructions</td>
<td>I should</td>
<td>3, 1755</td>
</tr>
<tr>
<td>Buckner</td>
<td>Alexandria</td>
<td>would</td>
<td>1755</td>
</tr>
<tr>
<td>Decembe</td>
<td>Alexandria</td>
<td>I should</td>
<td>1755</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS

- Word Matching
  (OTM data sets)
EXPERIMENTAL RESULTS

- **Word Matching**
  - Considering results of different $\tau$ values
  - Simply adding the distance matrices of different $\tau$ values
## EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Method</th>
<th>Data set</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>our approach</td>
<td>GW10</td>
<td>0.688</td>
<td>1.000</td>
</tr>
<tr>
<td>our approach</td>
<td>GW10</td>
<td>0.774</td>
<td>0.770</td>
</tr>
<tr>
<td>DTW (Rath and Manmatha [23])</td>
<td>GW10</td>
<td>0.653</td>
<td>0.711</td>
</tr>
<tr>
<td>DTW (Rath and Manmatha [22])</td>
<td>GW10</td>
<td>0.726</td>
<td>0.652</td>
</tr>
<tr>
<td>our approach</td>
<td>GW20</td>
<td>0.566</td>
<td>1.000</td>
</tr>
<tr>
<td>our approach</td>
<td>GW20</td>
<td>0.667</td>
<td>0.673</td>
</tr>
<tr>
<td>DTW (Rath and Manmatha [22])</td>
<td>GW20</td>
<td>0.518</td>
<td>0.550</td>
</tr>
<tr>
<td>our approach</td>
<td>OTM1</td>
<td>0.987</td>
<td>1.000</td>
</tr>
<tr>
<td>bag-of-words (Ataer and Duygulu [4])</td>
<td>OTM1</td>
<td>0.910</td>
<td>1.000</td>
</tr>
<tr>
<td>DTW (Ataer and Duygulu [4][1])</td>
<td>OTM1</td>
<td>0.940</td>
<td>1.000</td>
</tr>
<tr>
<td>our approach</td>
<td>OTM2</td>
<td>0.944</td>
<td>1.000</td>
</tr>
<tr>
<td>bag-of-words (Ataer and Duygulu [4])</td>
<td>OTM2</td>
<td>0.840</td>
<td>1.000</td>
</tr>
<tr>
<td>our approach</td>
<td>OTM1+2</td>
<td>0.957</td>
<td>1.000</td>
</tr>
<tr>
<td>bag-of-words (Ataer and Duygulu [4])</td>
<td>OTM1+2</td>
<td>0.810</td>
<td>1.000</td>
</tr>
</tbody>
</table>
## Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>WER</th>
<th>WER w/o OOV words</th>
<th>Language model post-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>our approach</td>
<td>0.303</td>
<td>0.189</td>
<td>-</td>
</tr>
<tr>
<td>Adamek et al. [1]</td>
<td>0.306</td>
<td>0.174</td>
<td>-</td>
</tr>
<tr>
<td>Lavrenko et al. [14]</td>
<td>0.449</td>
<td>0.349</td>
<td>+</td>
</tr>
</tbody>
</table>
EXPERIMENTAL RESULTS

- *Redif* Extraction
  - In the collection of 100 poems
  - We obtain a score of 0.682, when $k$ is set to 45
EXPERIMENTAL RESULTS

- **Redif Extraction**
  - For large values of k,
    - our method is able to extract more complicated redifs
    - it misses more number of redifs
  - If we decrease the minimum number of matches to be counted as redif, our method extracts
    - more number of correct contour segments as redifs; however,
    - the number of false matches also increases
  - For higher values of the same parameter,
    - the number of false matches decreases;
    - the number of misses increases.
Introduction

Line-based Word Representation

Word Matching Task: Word Image Matching using Line Descriptors (WILD)

Redif Extraction Task: Redif Extraction using Contour Segments (RECS)

Experimental Results

Conclusion
CONCLUSION

- Line-based representation schema
  - Two matching criteria using the proposed representation schema; WILD, and RECS
- In most of the cases, WILD outperforms the results of the existing studies
- RECS provides promising results and motivation for future studies.