Dec 1, 2010

Recognizing Handwritten Arabic Script through Efficient Skeleton-Based Grapheme Segmentation Algorithm

Dr. Gheith Ali Abandah
Fuad Jamour

The University of Jordan

Outline

- Introduction
- Limitations of previous algorithms
- Approach
  A. Sub-word separation
  B. Segmentation
  C. Recognition and post-processing
- Experiments and results
- Conclusions and future work

Introduction

- Arabic is a cursive language

عربية

- Holistic approaches are successful for limited vocabulary
- But there are 100,000s of Arabic words
- To support recognizing unconstrained handwritten Arabic script, we need an efficient segmentation solution

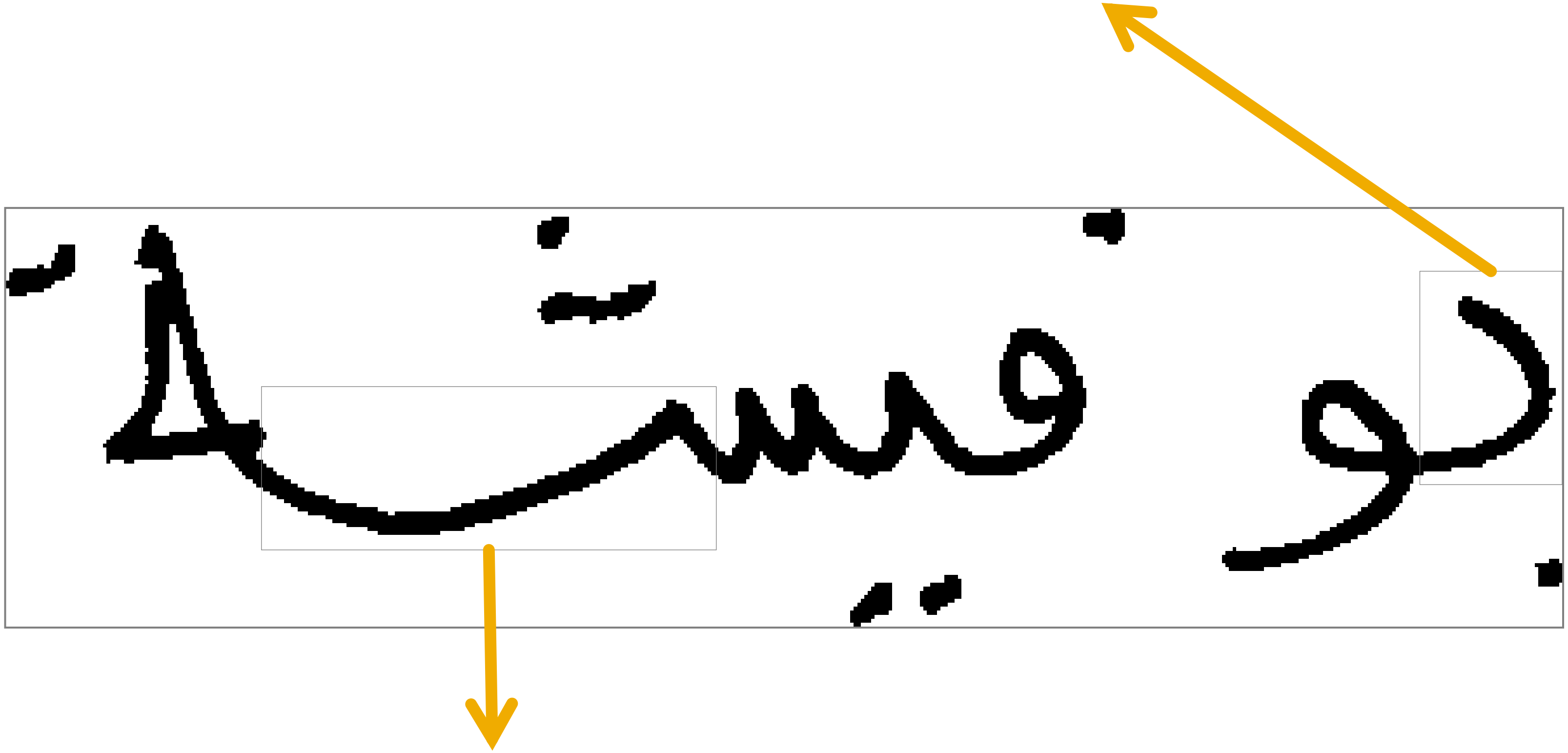
ع ر بـ يـ ة

Limitations of previous algorithms

- Previous segmentation approaches relied on detecting the following features to find the segmentation points:
  - Horizontal strokes near the base line
  - Changes in stroke width
  - Local minima
  - Etc.

Limitations of previous algorithms – examples

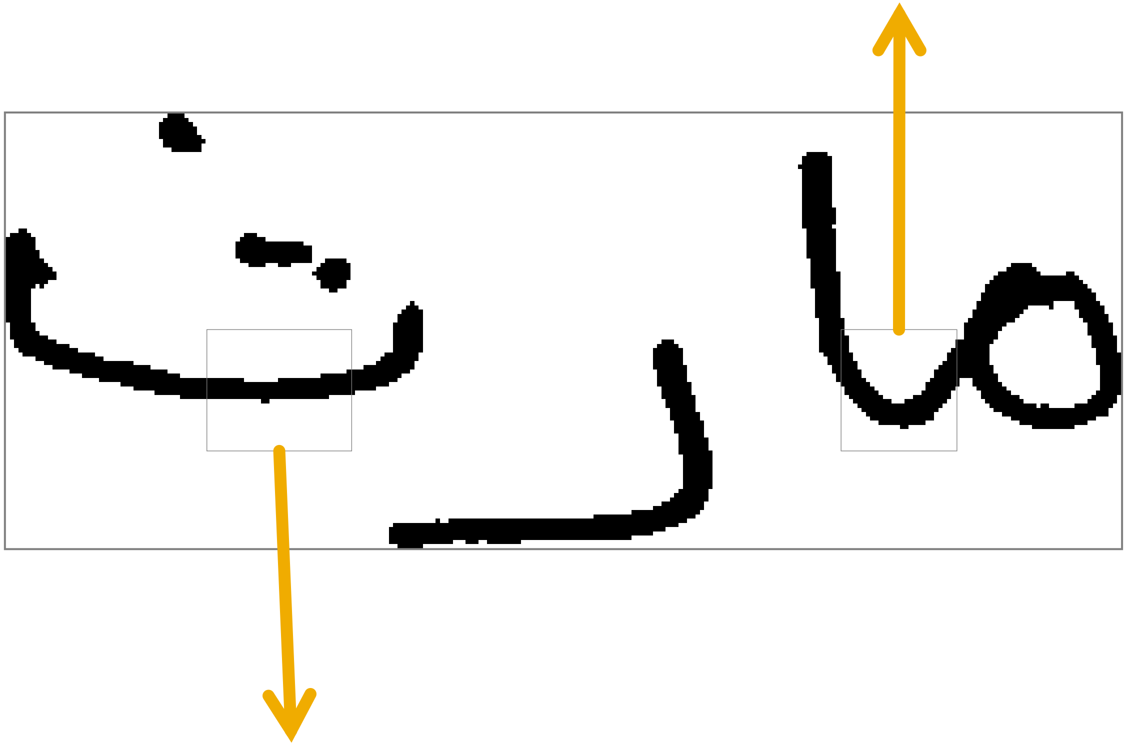
Not a horizontal stroke, missed



Long stroke, over-segmented

Limitations of previous algorithms – examples

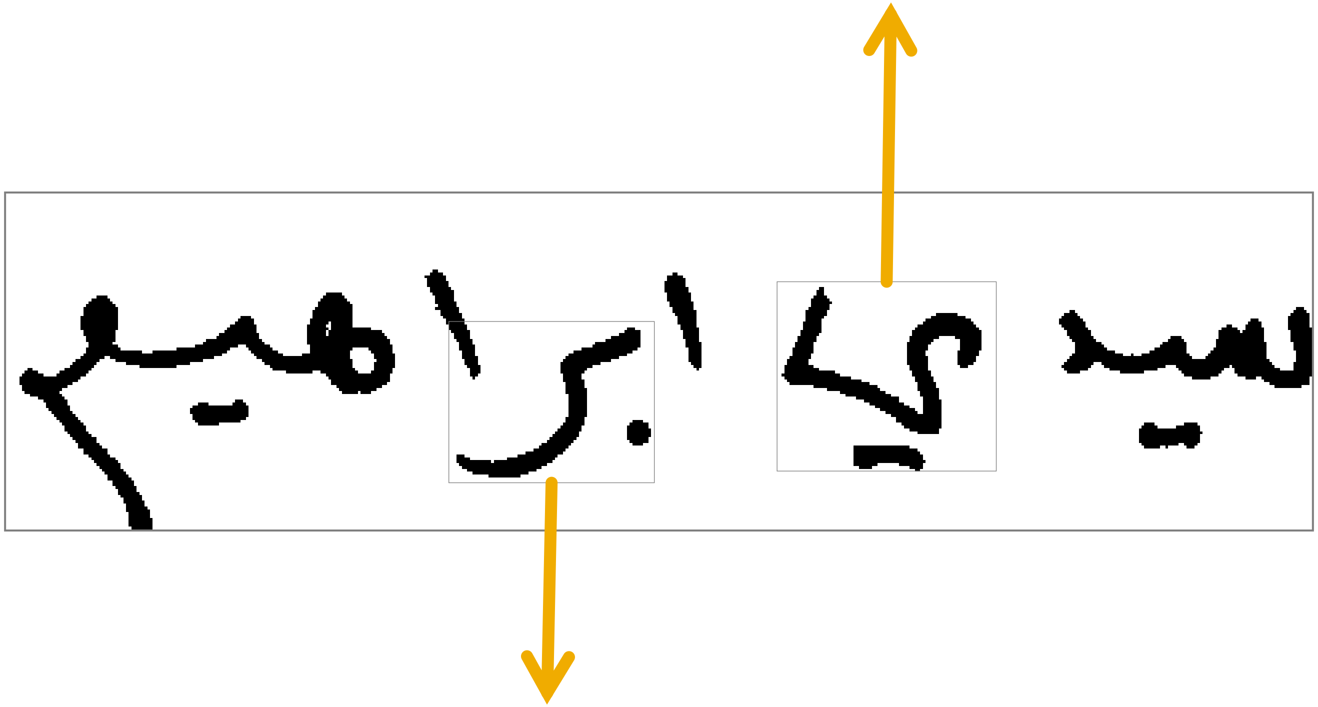
Constant stroke width, missed



Stroke with pit, over-segmented

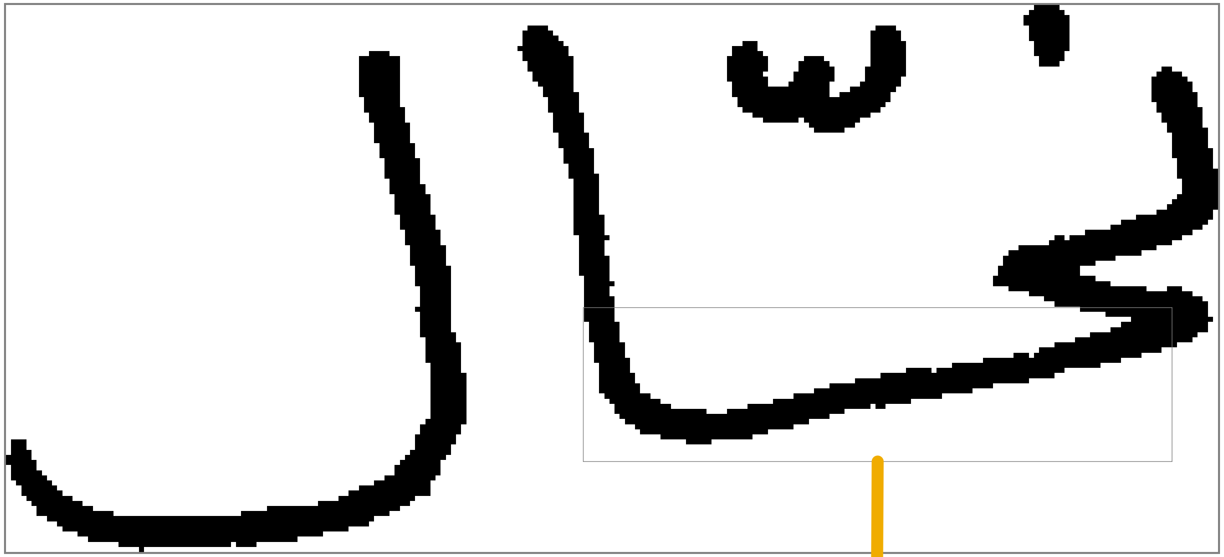
Limitations of previous algorithms – examples

Local min, over-segmented



No local min, missed

Limitations of previous algorithms – examples

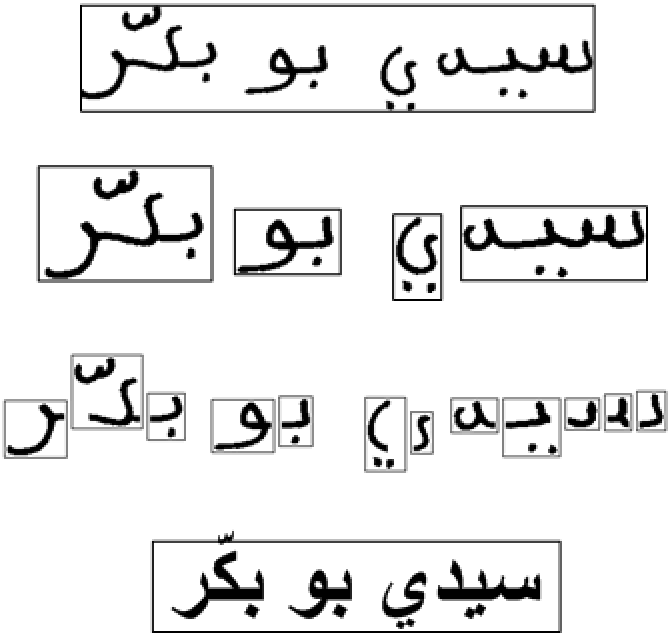


Baseline not horizontal, missed

Approach

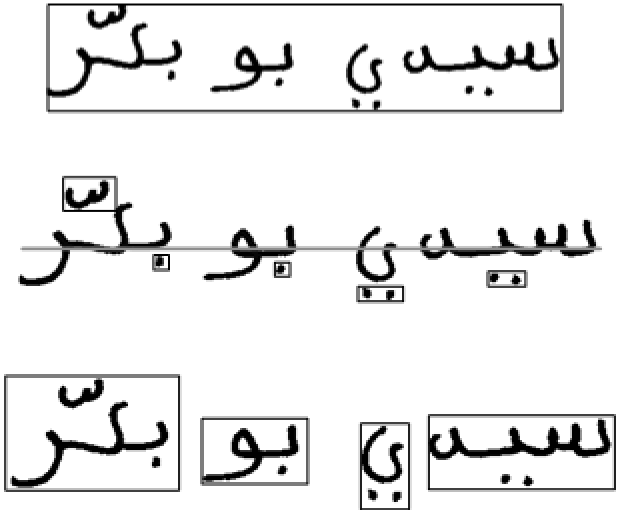
Skeleton-based grapheme segmentation algorithm.

A. Sub-word separation
B. Segmentation
C. Recognition and post-processing



A. Sub-word separation

1. Baseline estimation
2. Secondary bodies identification
3. Sub-word extraction and secondary bodies assignment



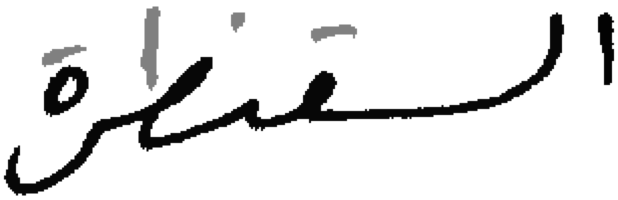
A. Sub-word separation – cont.

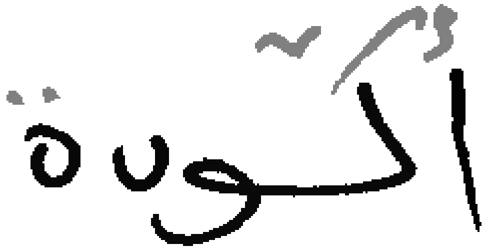
2. **Secondary bodies identification:**

a) Body is very small compared to other bodies in the same image

b) It is relatively small and far from the baseline

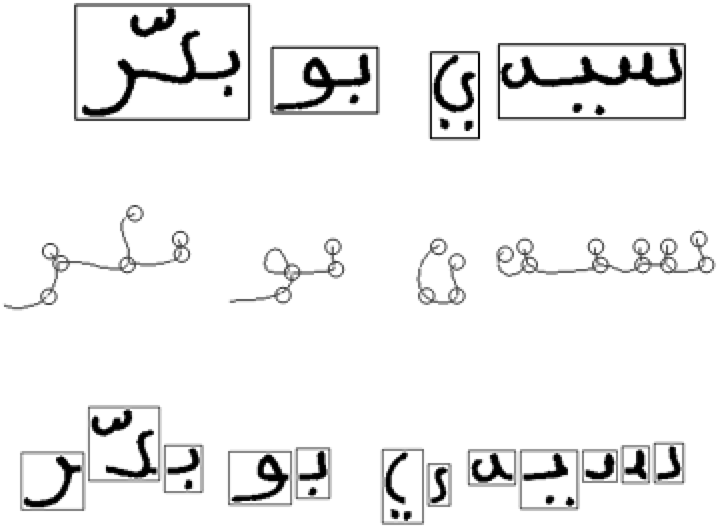
c) It is a vertical line and has a relatively large body below it





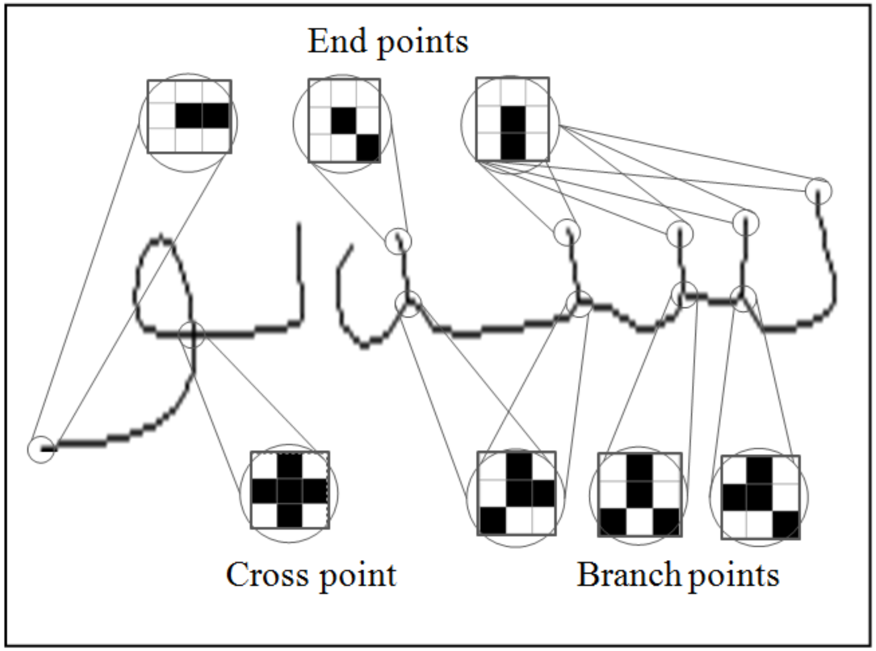
B. Segmentation

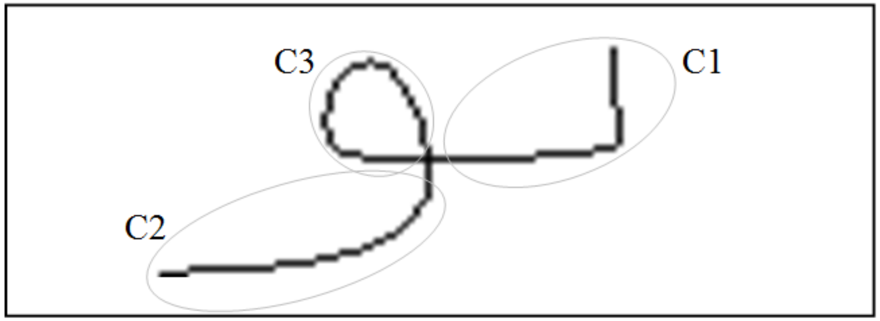
1. Thinning and feature points identification
2. Continuities identification
3. Subtle branch points and edge points detection
4. Rule-based segmentation
5. Grapheme separation



2. Segmentation – cont.

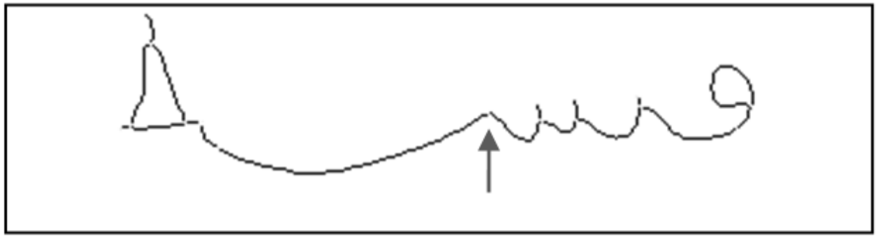
1. Thinning and feature points identification
   - End points
   - Branch points
   - Cross points
2. Continuities identification



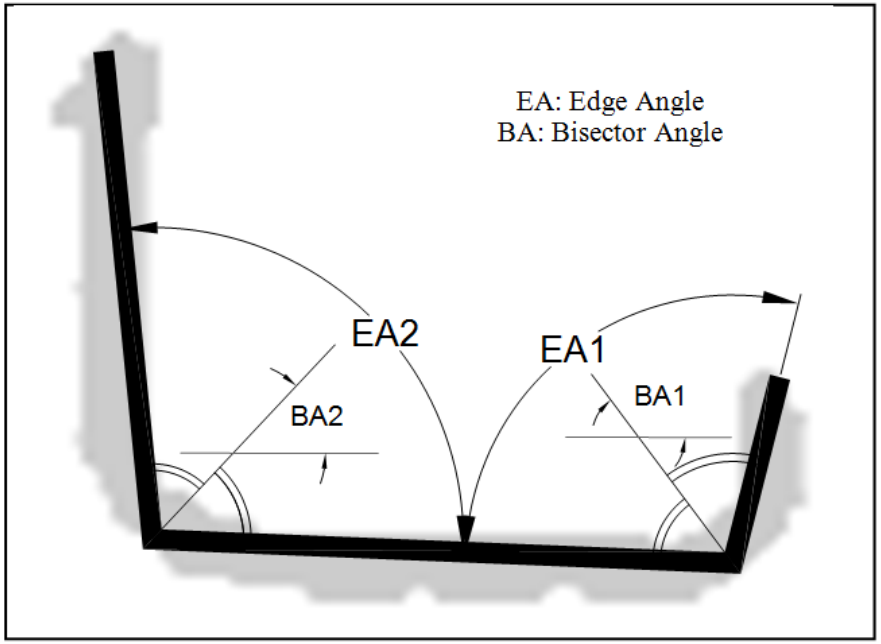


B. Segmentation – cont.

3. Subtle branch points and edge points detection

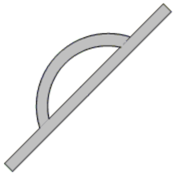


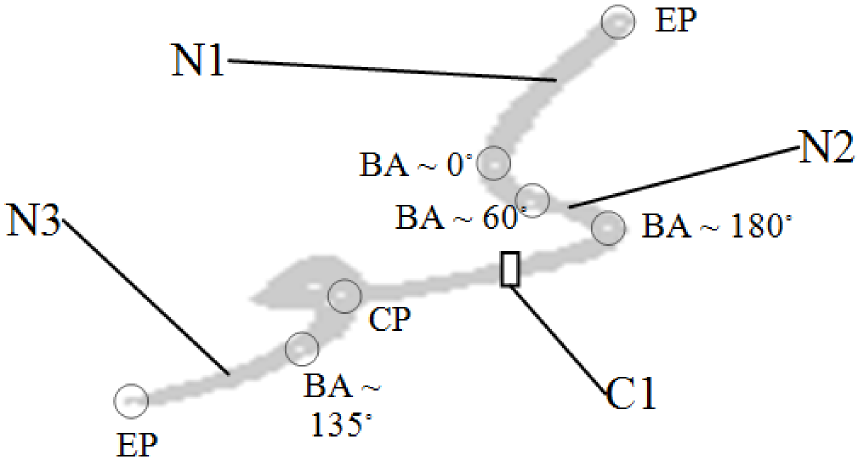
For each edge point, find *edge angle* and *bisector angle*



B. Segmentation – cont.

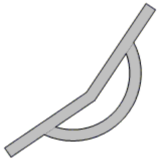
4. Rule-based segmentation
   a) Not vertical: the orientation of the continuity should be between -45° and +45°
   b) If the right end is an edge, its bisector angle should be between 45° and 225°
   c) The left end is not an end point

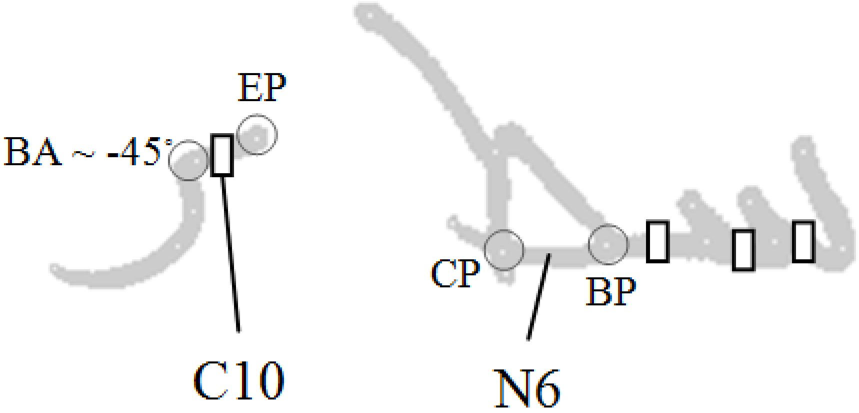




B. Segmentation – cont.

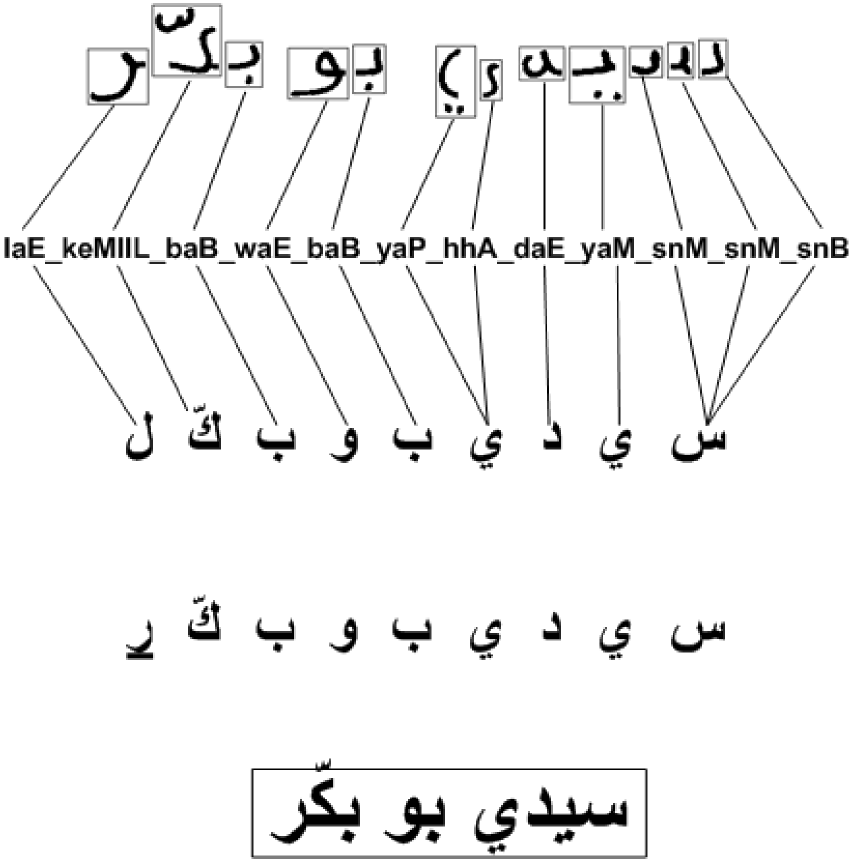
4. Rule-based segmentation
   - d) If the left end is an edge, its bisector angle should be between -155° and 65°
   - e) It is not totally covered from above or from below





C. Recognition and post-processing

1. Grapheme recognition (Tesseract)
2. Graphemes to characters (lookup table with weights)
3. Word matching



Experiments and results

Measure	Count	Percentage
Total words	107	100%
Under-segmented words	1	1%
Over-segmented words	3	3%
Total characters	882	100%
Characters correctly recognized	763	87%
Words correctly recognized	101	94%

Lower accuracies with more samples.

Conclusions

- Proposed algorithm solves problems found in other algorithms.
  - Does not depend on baseline estimation, thus it avoids baseline estimation problems
  - Does not assume that the segmentations points are always on horizontal continuities of specific lengths, thus avoids problems in segmenting slanted and long strokes
  - Does not depend on stroke width and local minima, thus avoids problems with pitted and constant-width strokes
  - It analyzes the edge points to avoid undesirable over and under segmentation

Future work

- We have dropped Tesseract as our recognition engine and we are using other feature extraction and grapheme classification techniques that are more suitable for handwritten Arabic script

Question?

Thank You

- **Contact information**
  - Email: abandah@ju.edu.jo
  - Homepage: http://www.abandah.com/gheith