Segmentation Free Spotting of Cuneiform using Part Structured Models

Heidelberg University, **Bartosz Bogacz**
Smith College, **Nicholas Howe**
Heidelberg University, **Hubert Mara**

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

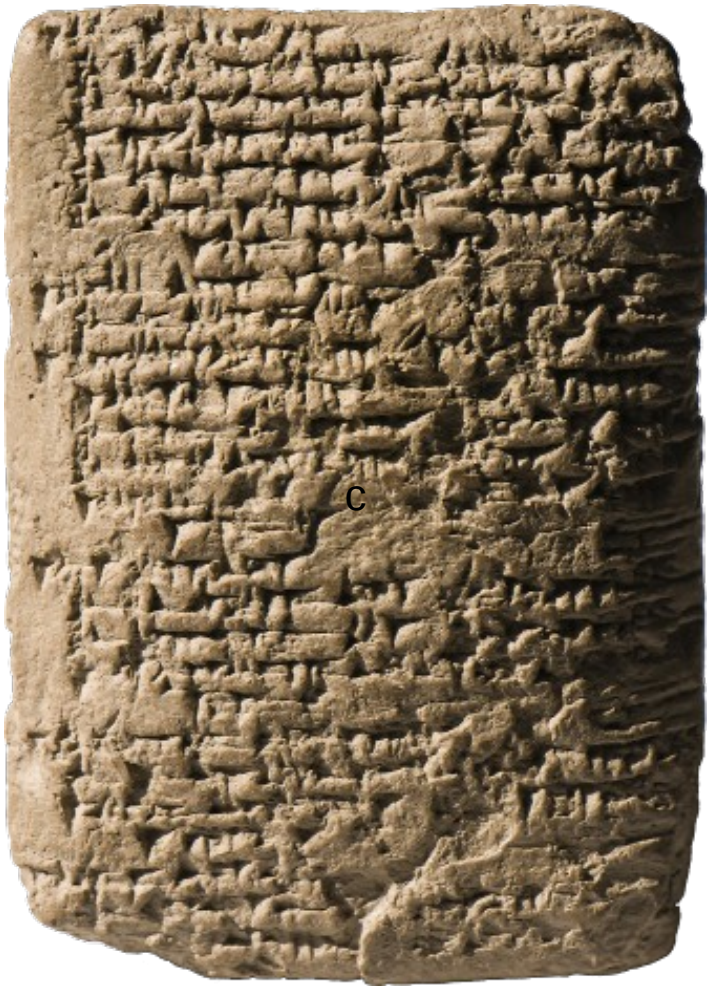
IWR Interdisciplinary Center for Scientific Computing

HGS MathComp

SMITH COLLEGE

Cuneiform Script

- More than 3,000 years of history
- Evolved from a pictographic to a syllabic script
- More than 500,000 clay tablets
- Only few Assyriologists

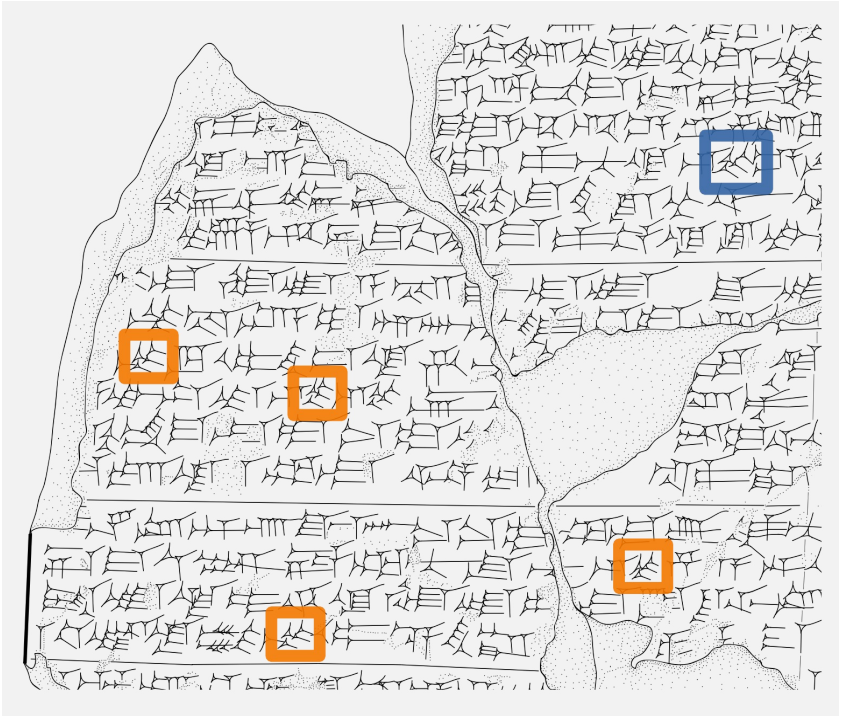


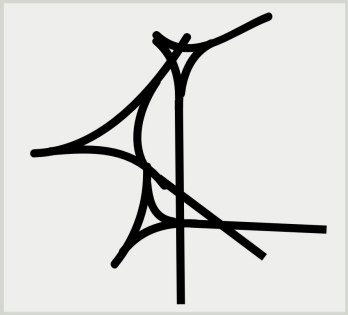
Cuneiform Script

- Cuneiform is a writing system used by at least 7 different languages
- Written by impressing a rectangular stylus in wet clay
- Our approach models geometric patterns instead of language

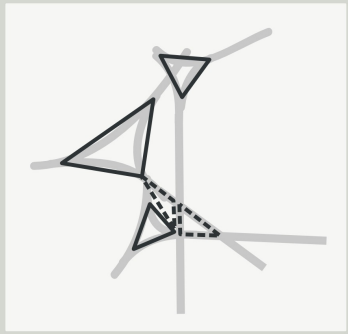
Goal

- Only few tablets are transliterated
- Transliterations can be incomplete and subjective
- Provide a mechanism for searching by graphical query

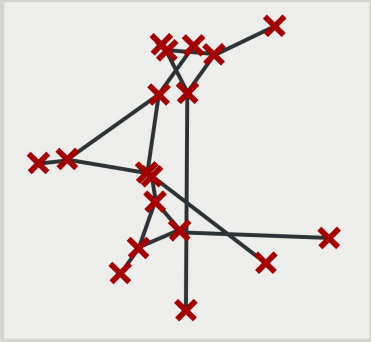




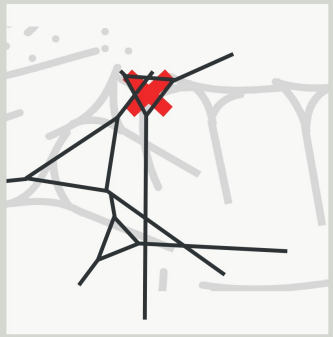
Cuneiform script



Optimal assignment

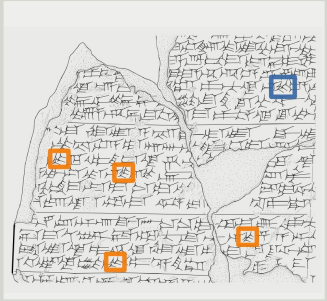


Wedge modeling

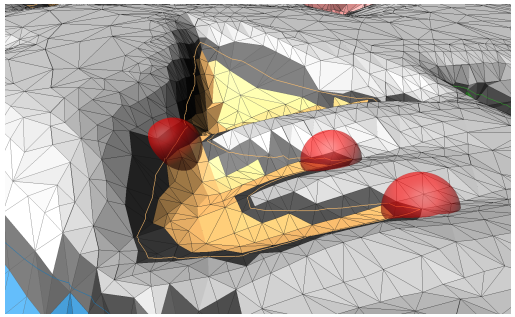


Part-structured spotting

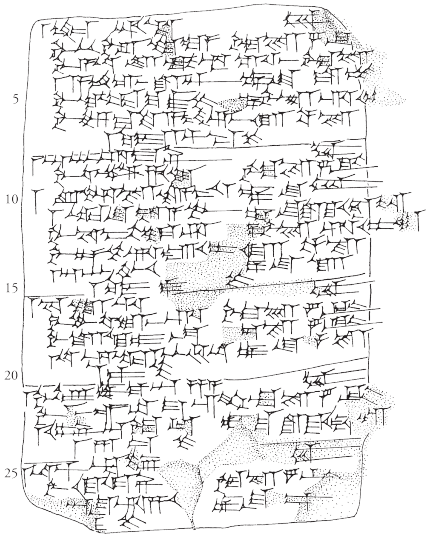
Word spotting



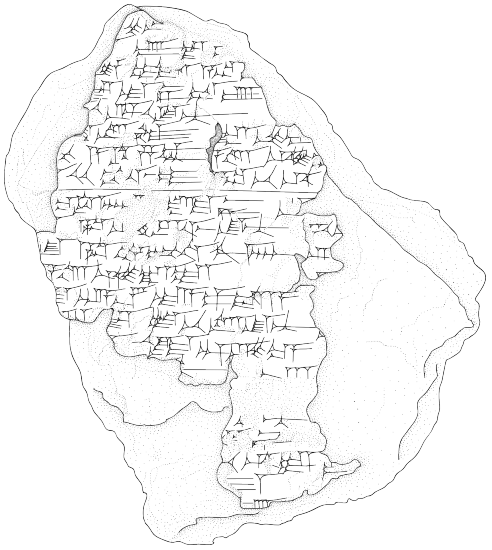
Different Sources



3D Scans



Retro-digitized



Born-digital

Unification of sources requires a common geometrical representation

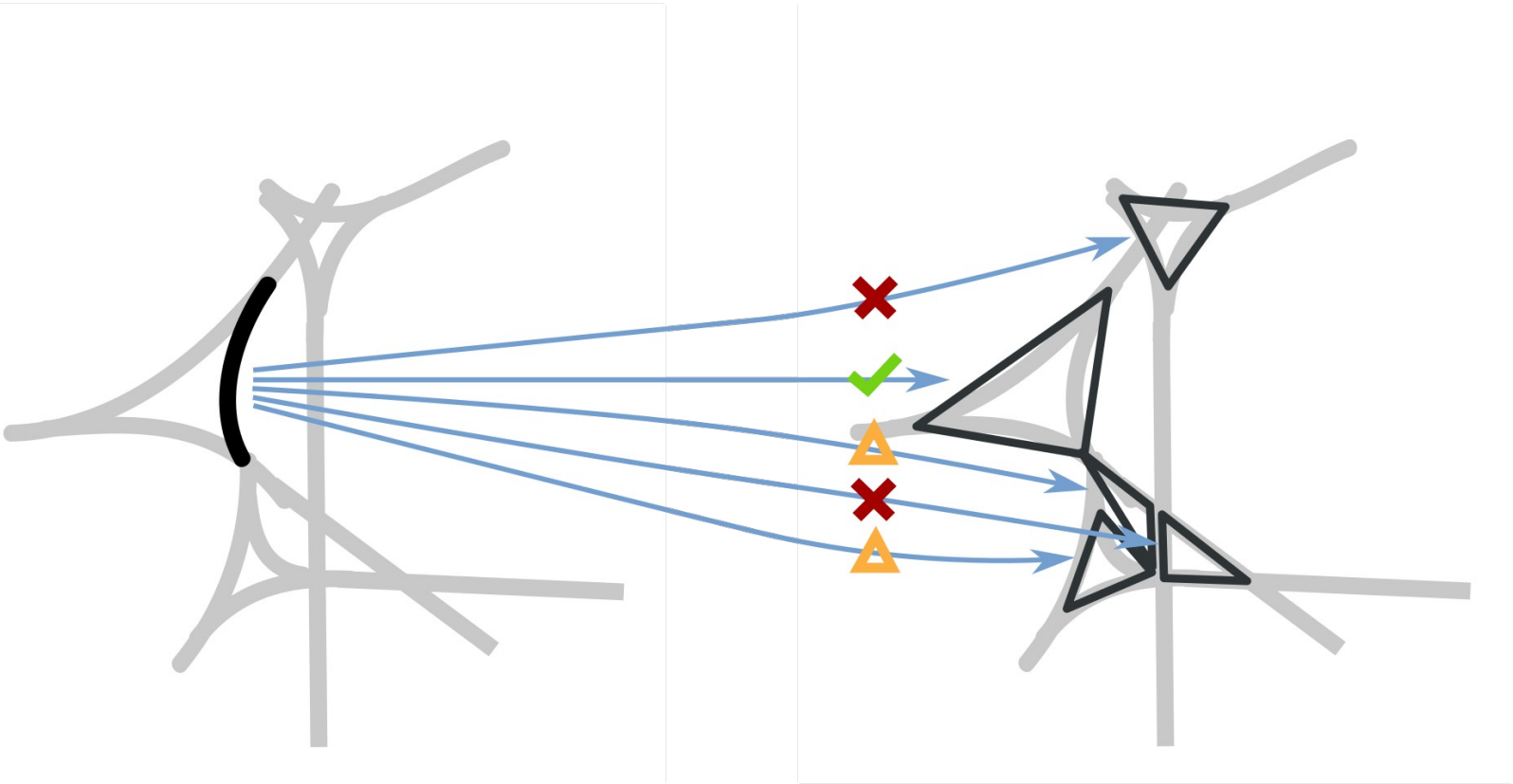
Extracting Wedges

- We model wedges as triangles with arms
- Find possible candidate wedges by finding cycles
- Prune this set of candidates using modeling constraints
  - No overlapping wedges
  - Sizes and angles are within sane bounds
  - Prioritize bigger wedges

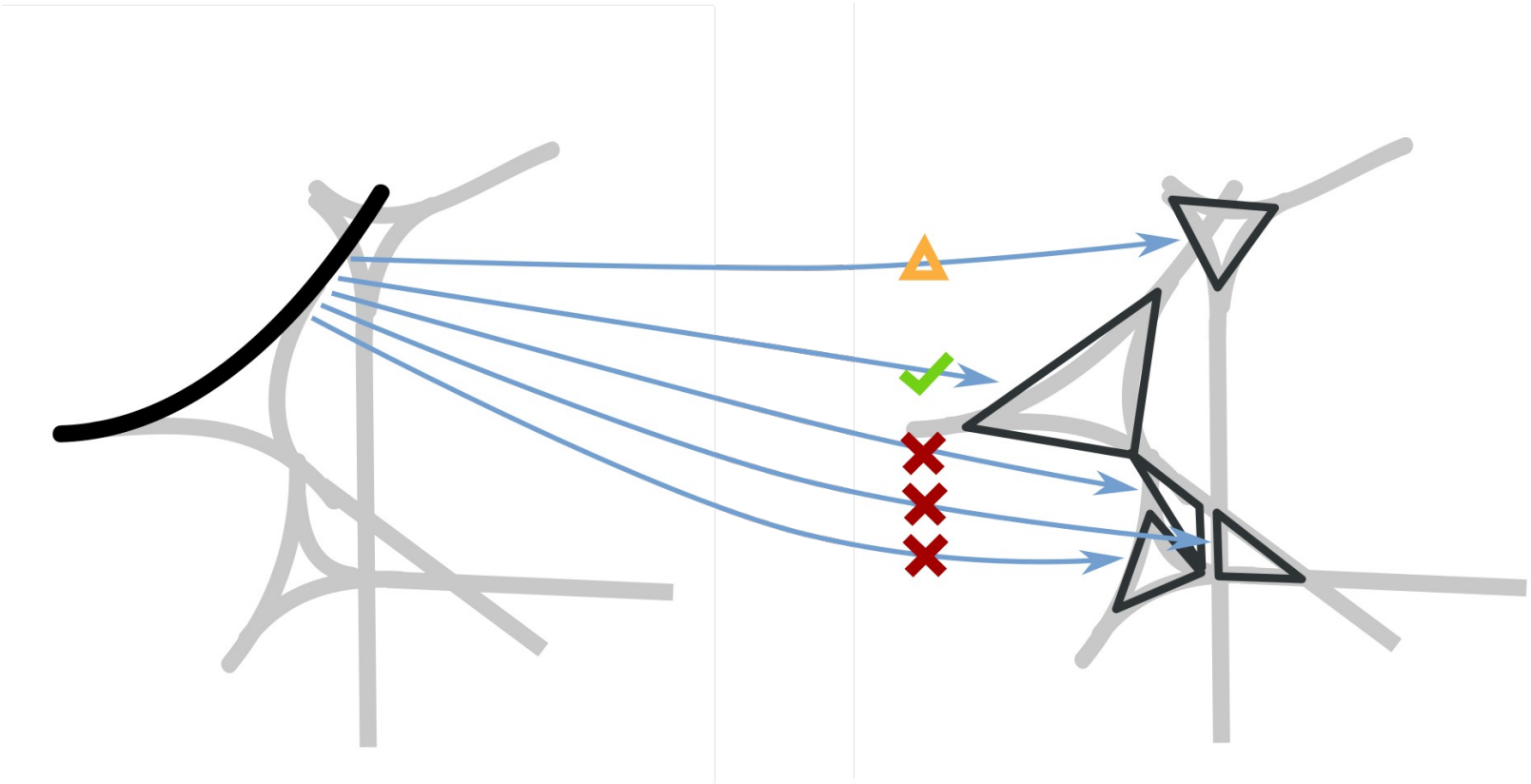
Extracting Wedges

- We re-formulate this constraint satisfaction task as an optimizing assignment task
- This enables us an efficient *O(n^3)* solution
- The set of strokes is being assigned to a set of candidate wedges

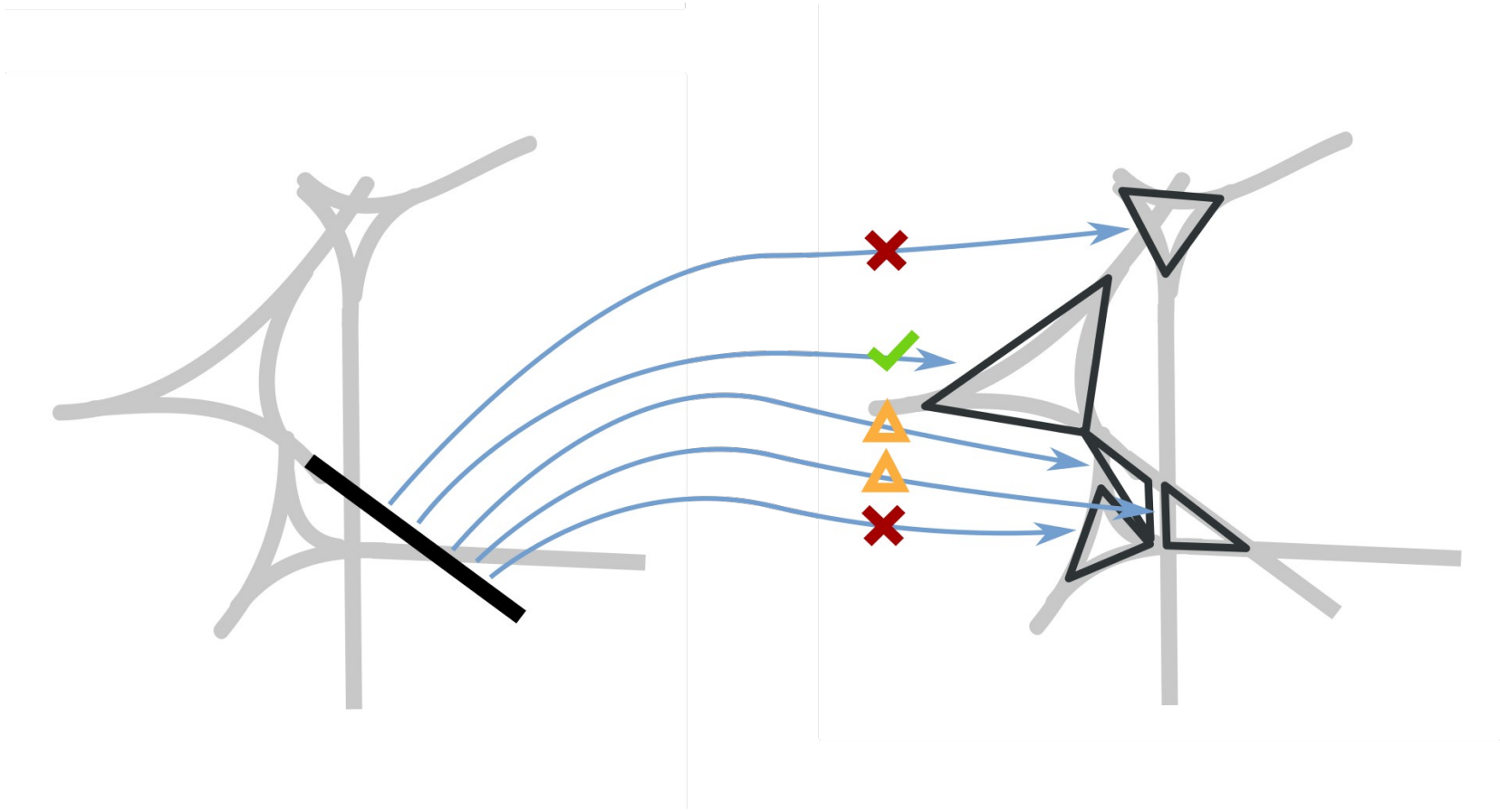
Optimal Assignment



Optimal Assignment



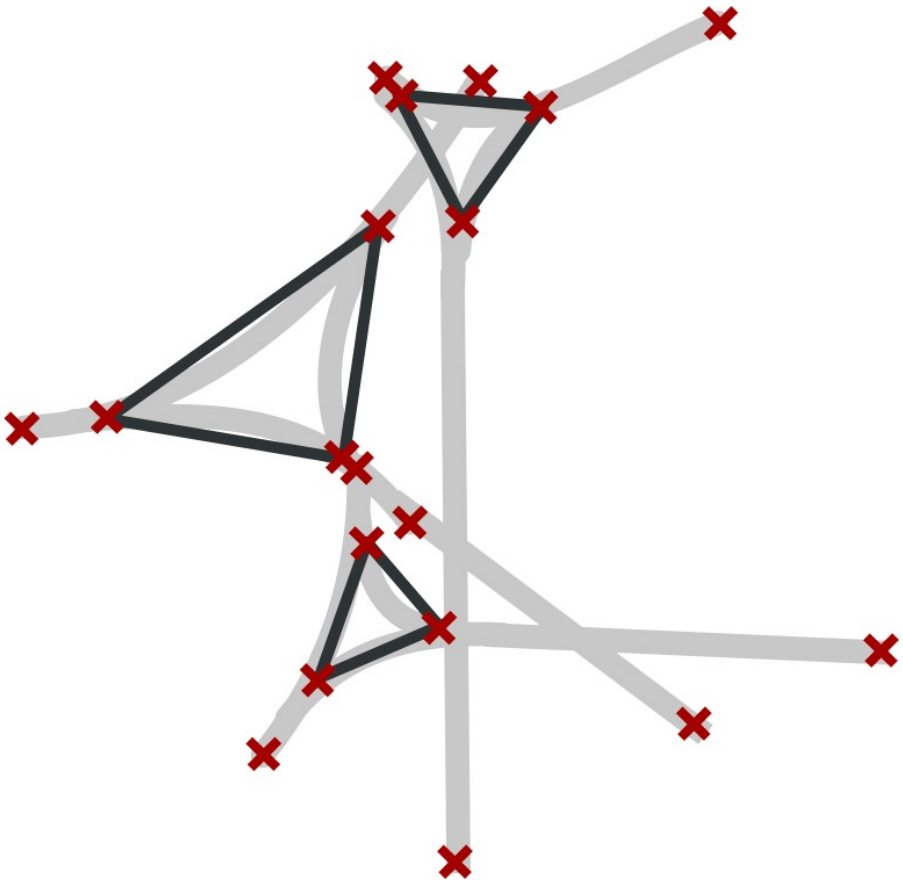
Optimal Assignment



Optimal Assignment

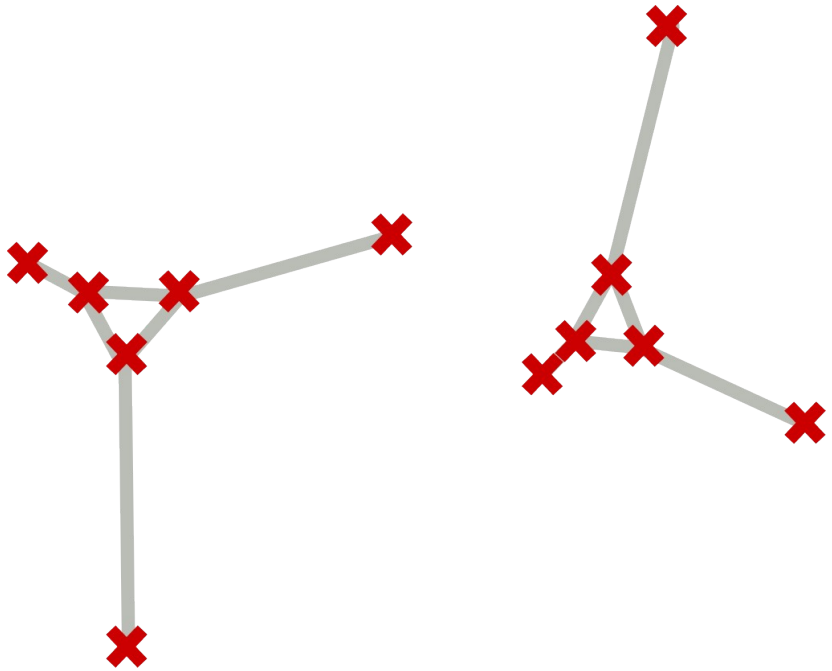
Wedge Features

- We want to represent extracted wedges as feature vectors
- Intersections and endpoints are most salient points in wedges
- Model wedges using these keypoints



Keypoint Model

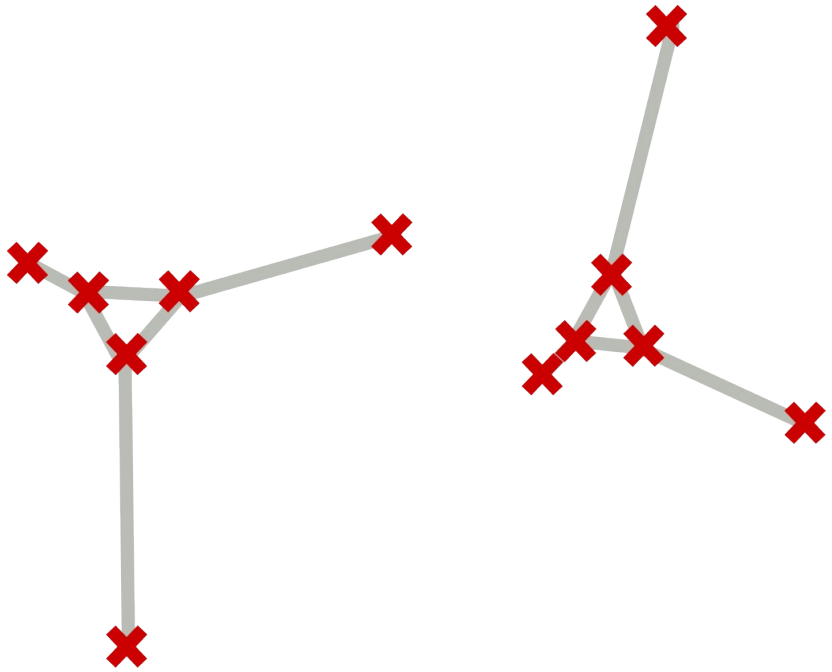
- Feature vector is a concatenation of the keypoints in our wedge model
  - Wedge-head intersections
  - Wedge-arm endpoints



$$f^k = (\overbrace{x_1y_1\ x_2y_2\ x_3y_3}^{\text{Wedge-head}}\ \overbrace{x_4y_4\ x_5y_5\ x_6y_6}^{\text{Wedge-arms}})$$

Keypoint Model

- Features are compared by Euclidean distance

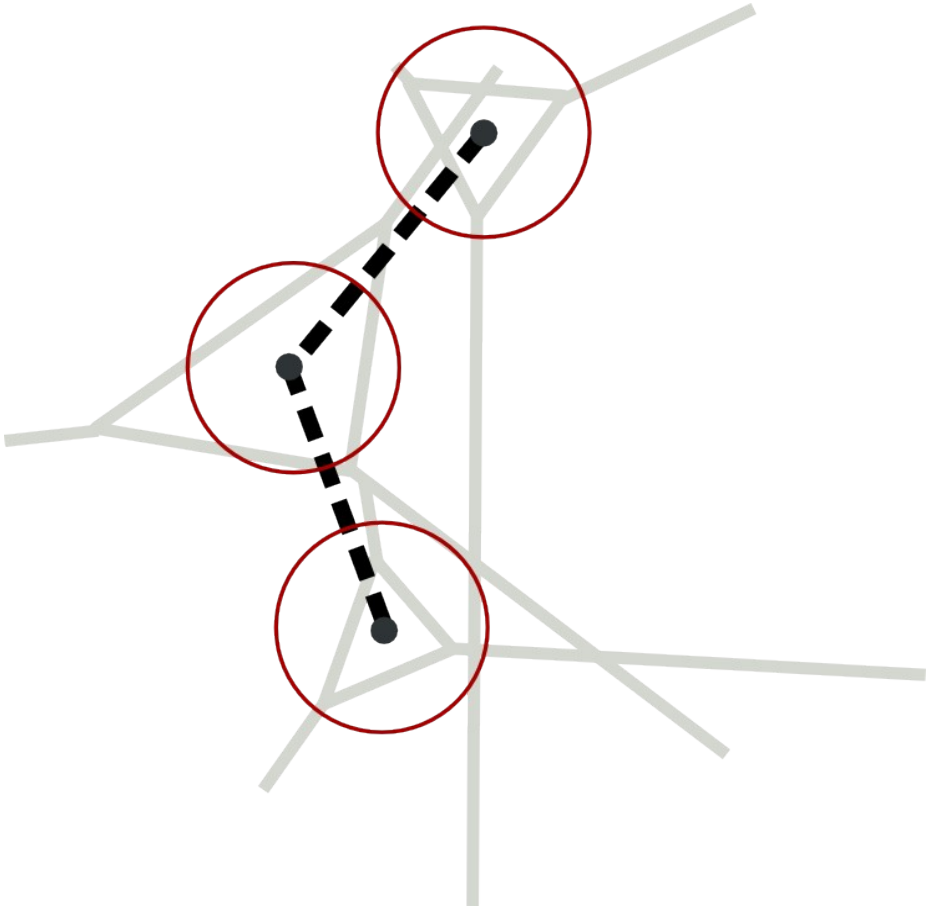


- Our new approach reorders points using optimal assignment

$$f^k = (\overbrace{x_1 y_1 \; x_2 y_2 \; x_3 y_3}^{\text{Wedge-head}} \; \overbrace{x_4 y_4 \; x_5 y_5 \; x_6 y_6}^{\text{Wedge-arms}})$$

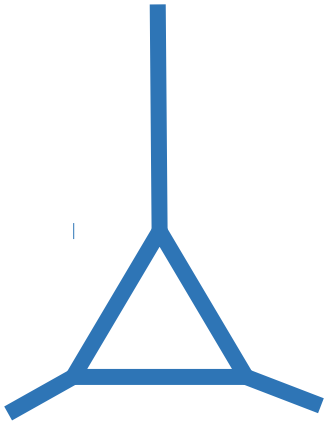
Part-structured Spotting

- Model characters as wedges connected by tree of flexible links
- Align query to candidates by deforming links
- Probability of a match is wedge similarities plus amount of link deformation

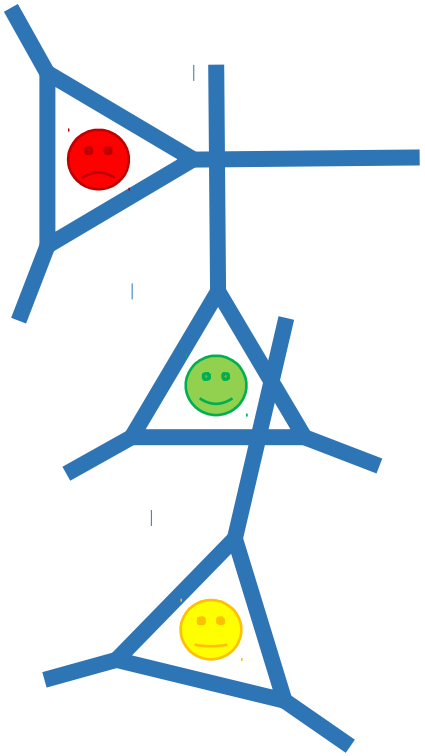


Generalized Distance Transform

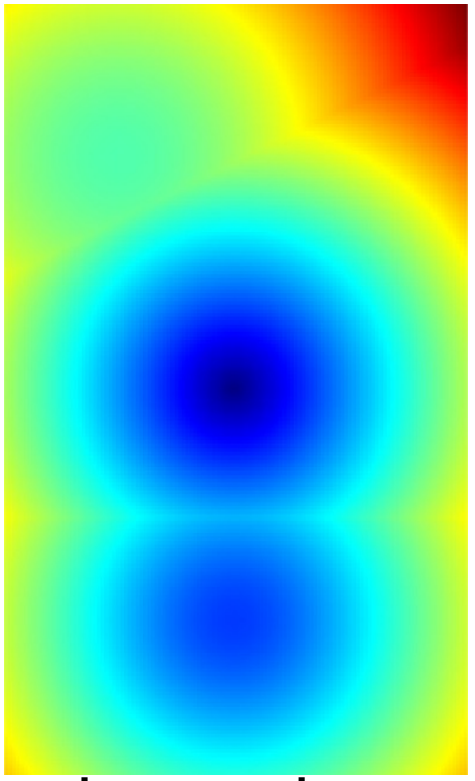
Query



Target



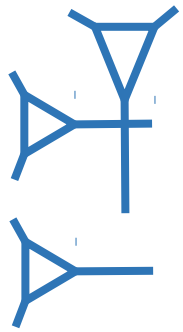
GDT



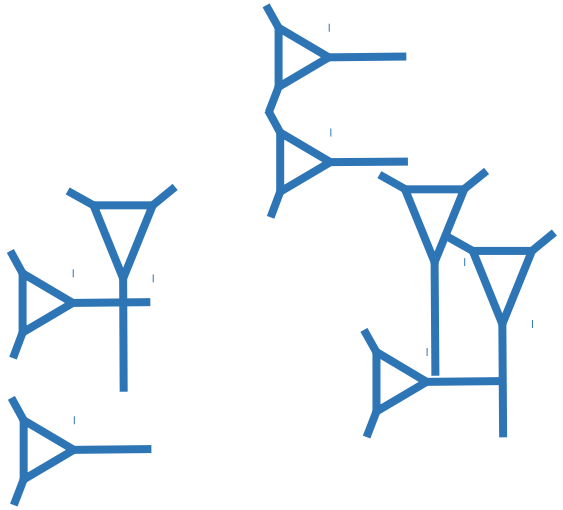
- Trades off between wedge similarity and distance

Part Structured Match Demo

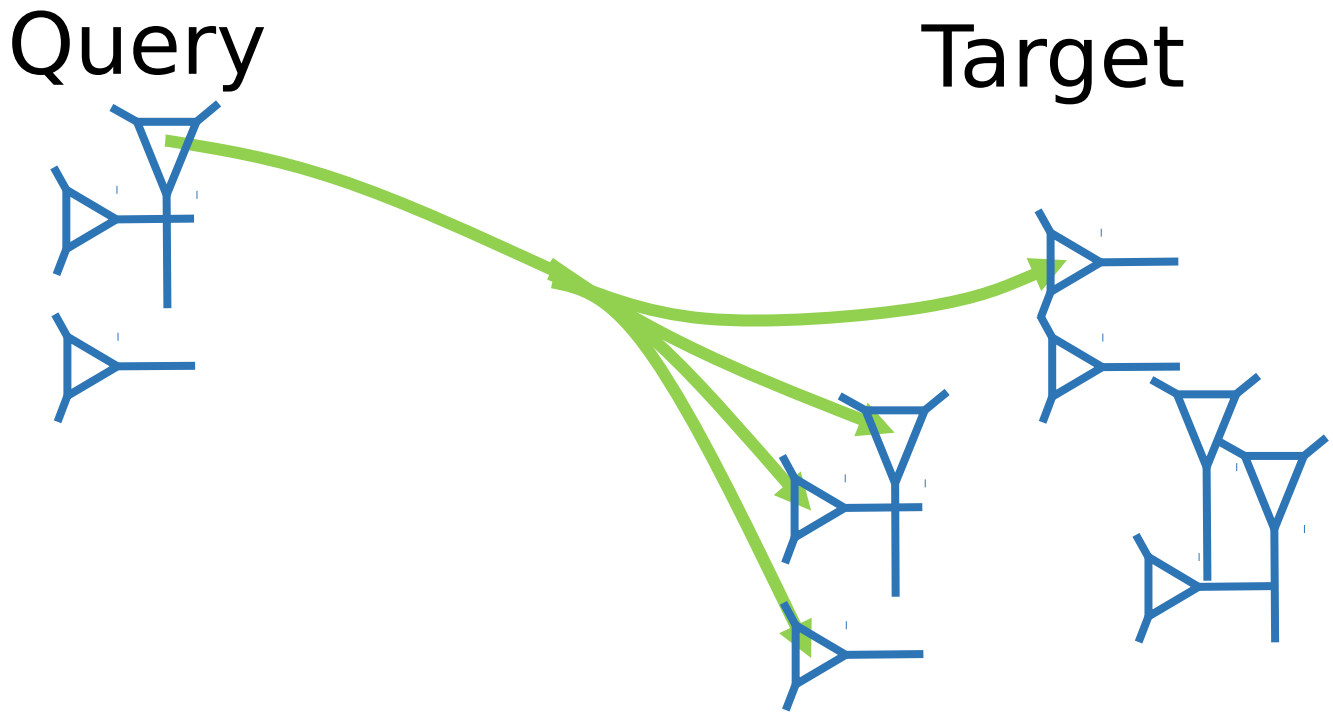
Query



Target

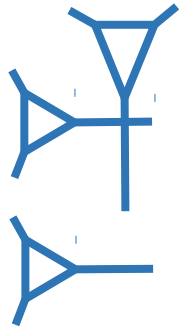


Part Structured Match Demo

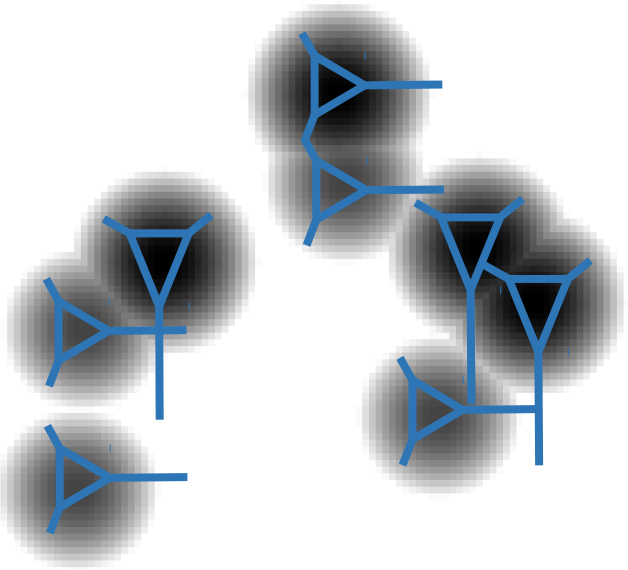


Part Structured Match Demo

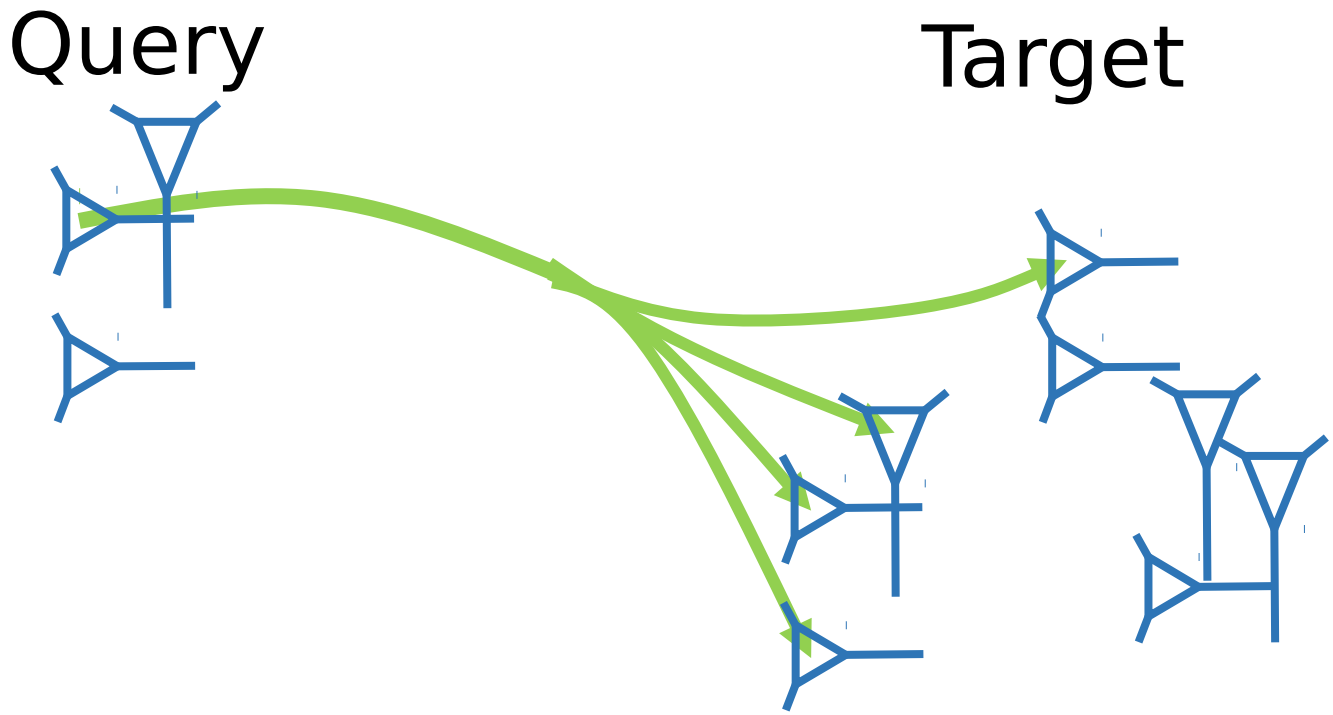
Query



Target

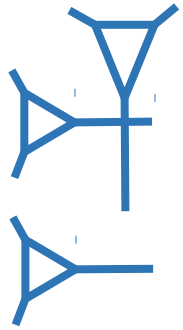


Part Structured Match Demo

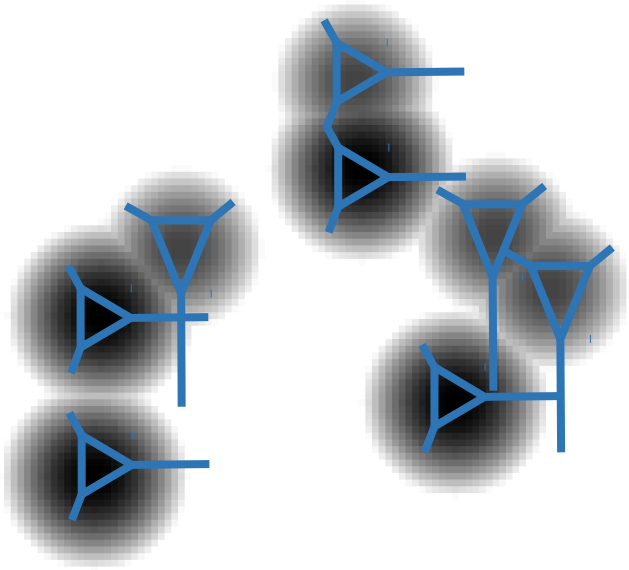


Part Structured Match Demo

Query

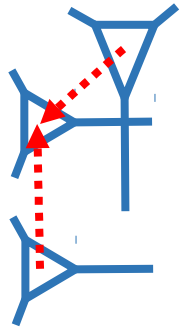


Target

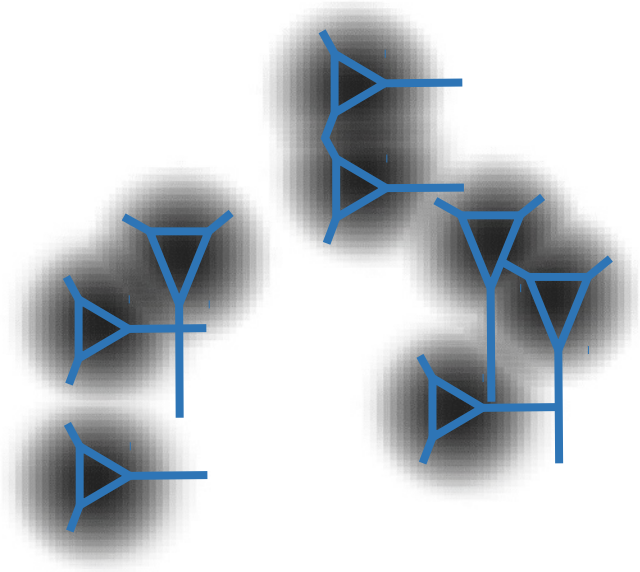


Part Structured Match Demo

Query

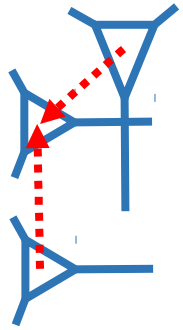


Target

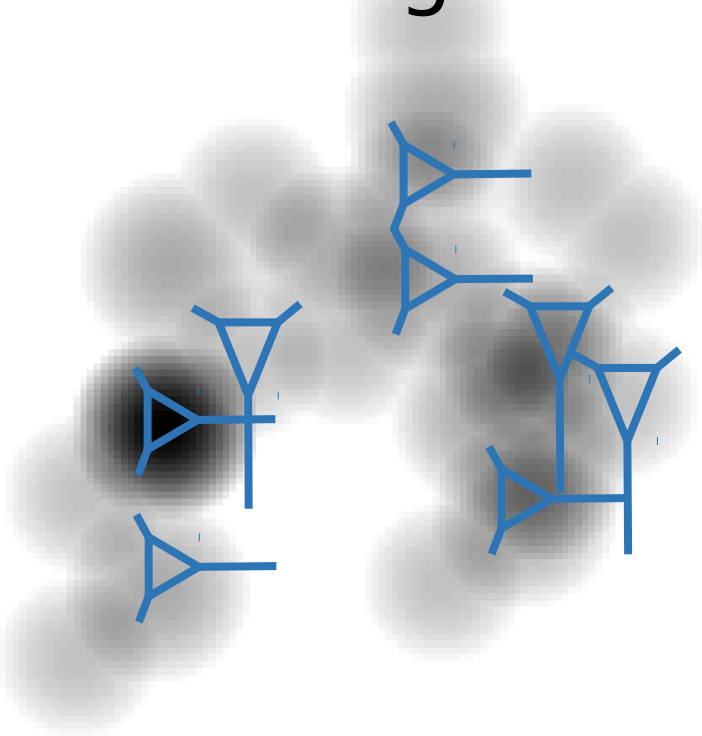


Part Structured Match Demo

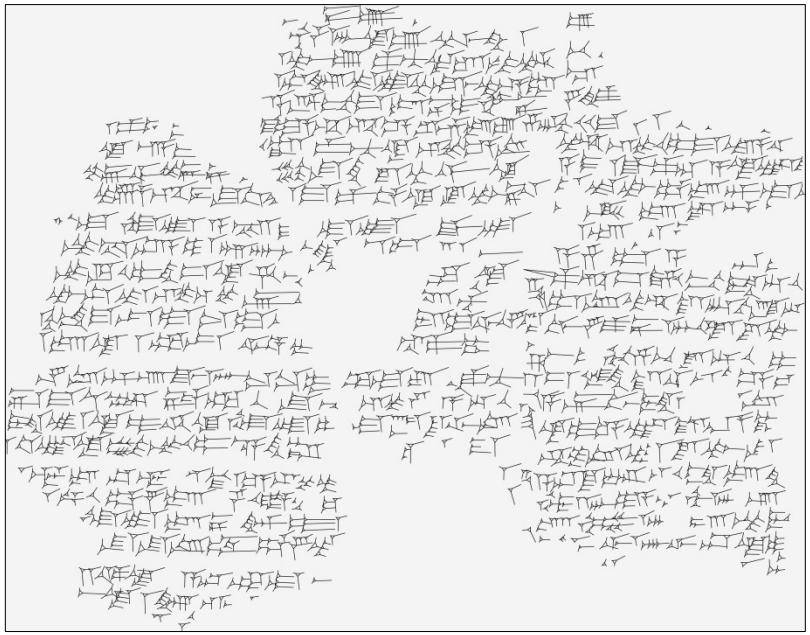
Query

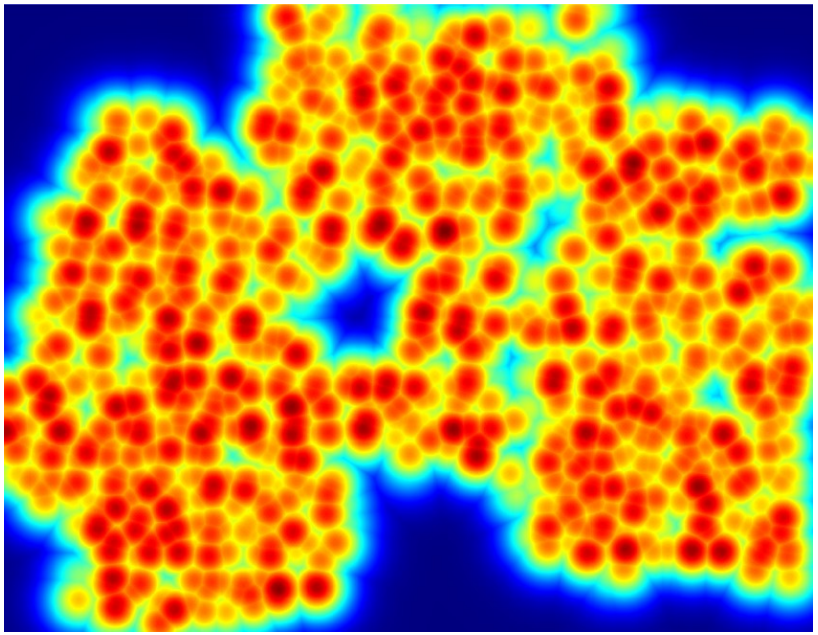


Target



Sample Results





Evaluation

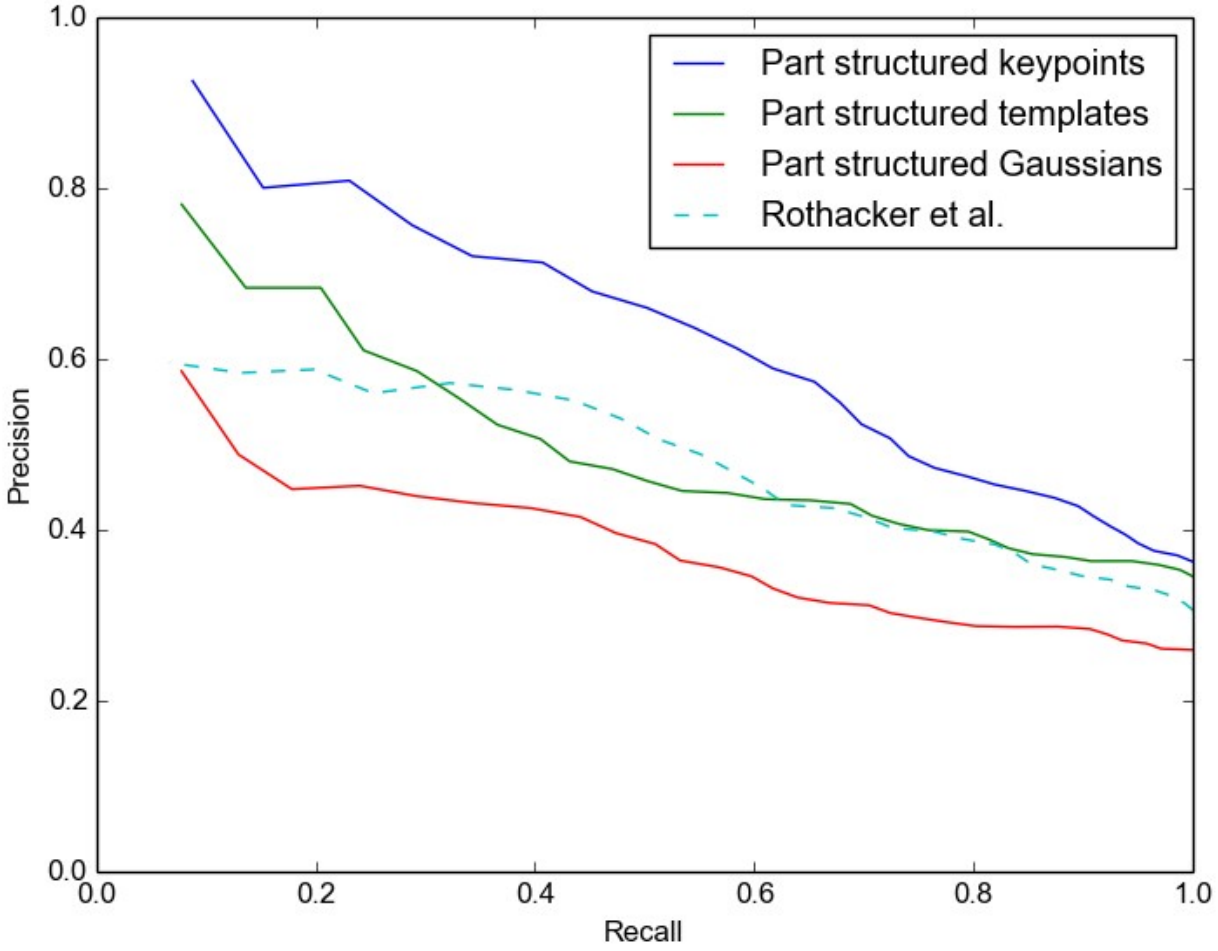
- Symbol spotting task with 40 query symbols of various lengths

- We compare against Rothacker et al. HMM Latin word spotting
  - No elevation data to evaluate their approach for cuneiform spotting
  - We rasterize our data to make it available for their method

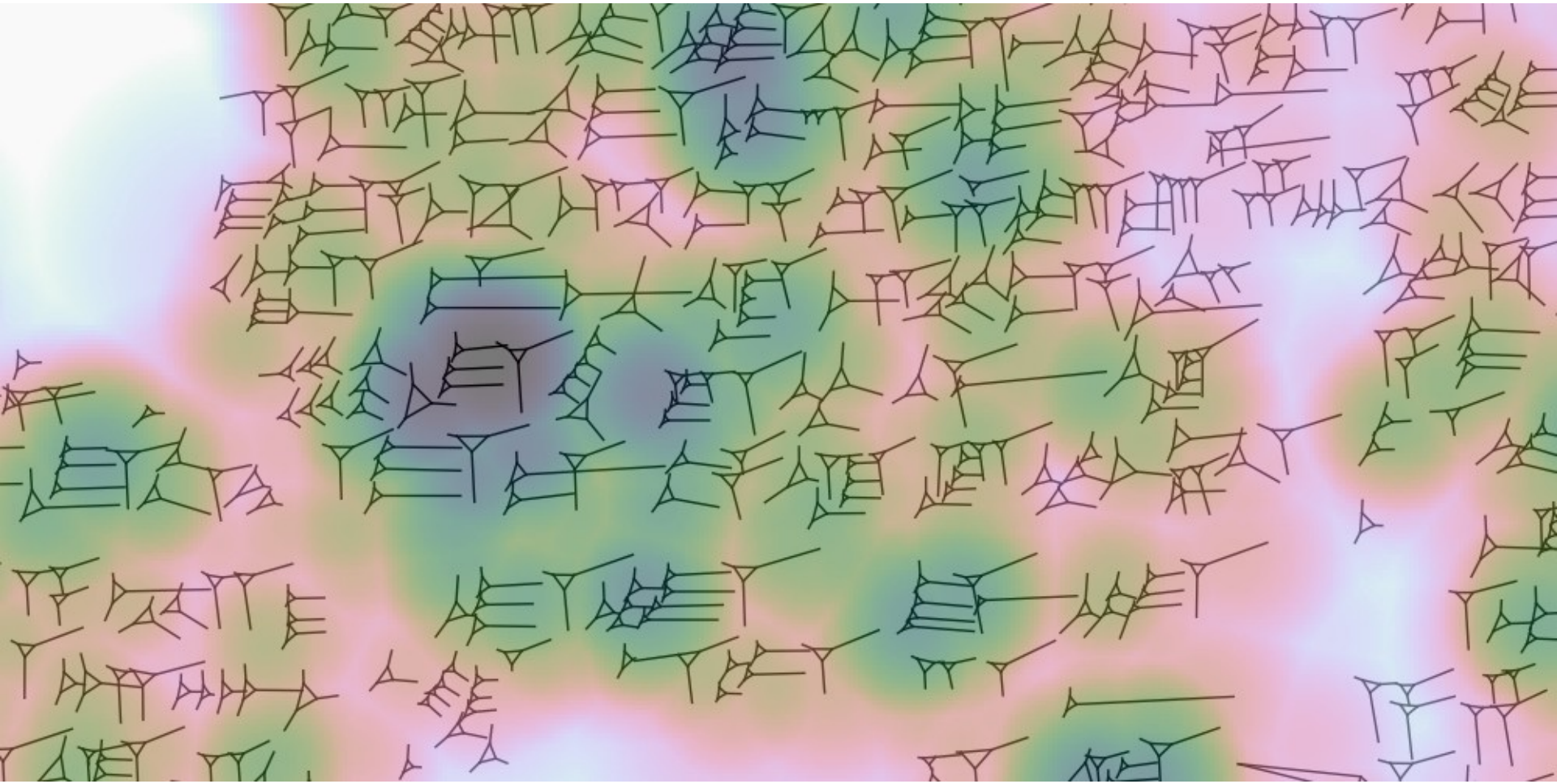
Evaluation

- Dataset are two cuneiform tablets with 500 identifiable characters
- Tablets are only incompletely labeled, precluding an automated evaluation
- Retrieval results are checked by an expert for false positives

Evaluation


Part structured keypoints
Part structured templates
Part structured Gaussians
Rothacker et al.
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

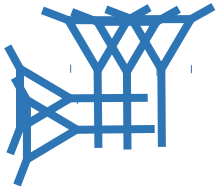
Query Results



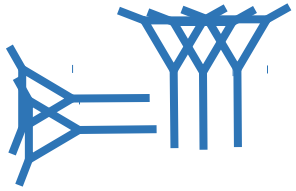
Summary

- Fast and optimizing method for cuneiform wedge detection
- Native and accurate feature representation of cuneiform wedges
- Fast symbol spotting of cuneiform characters

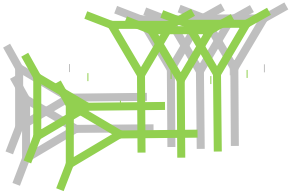
Part-Structured Spotting vs. Template Matching



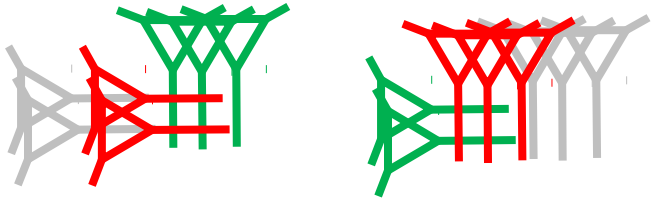
Query



Target



Part-structured:
Approximate match
everywhere



Template:
Matches only part