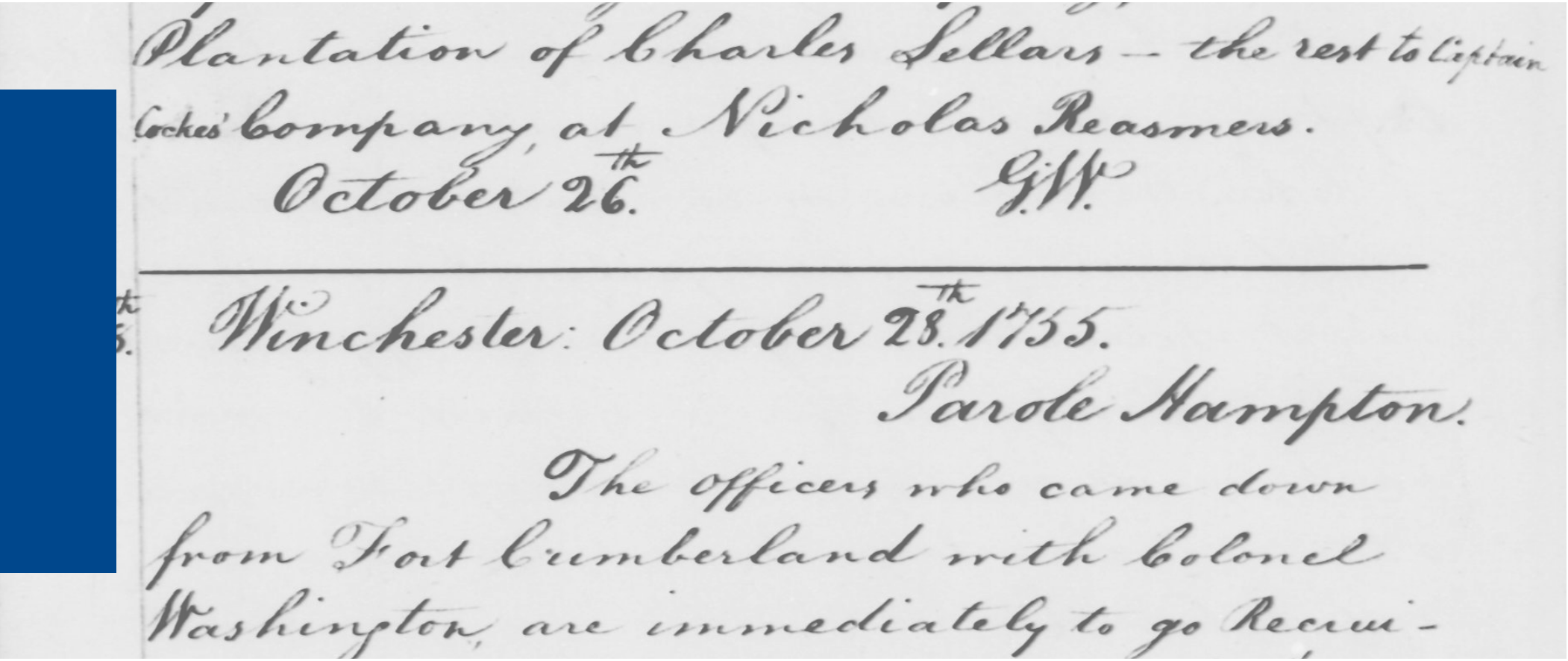
S+SSPR 2016

A Novel Graph Database for Handwritten Word Images &

Graph-based Keyword Spotting in Historical Handwritten Documents



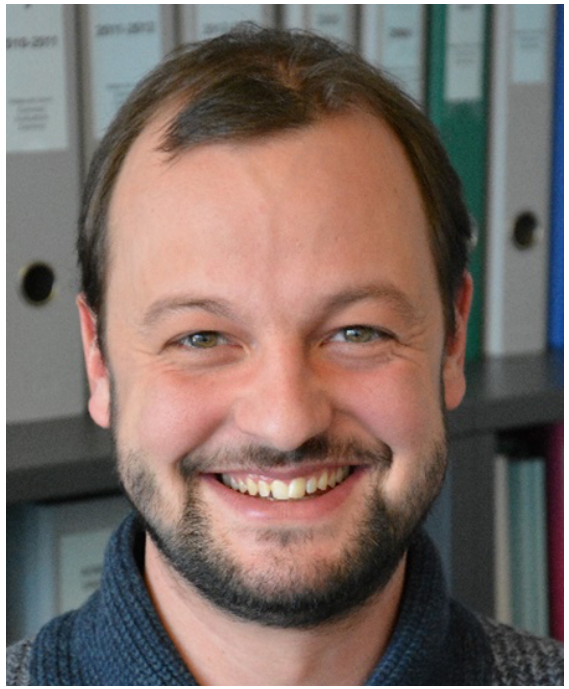
Team

Michael Stauffer

University of Pretoria

University of Applied Sciences and Arts Northwestern Switzerland

Andreas Fischer



University of Fribourg

University of Applied Sciences and Arts Western Switzerland

Kaspar Riesen

University of Applied Sciences and Arts Northwestern Switzerland

Content

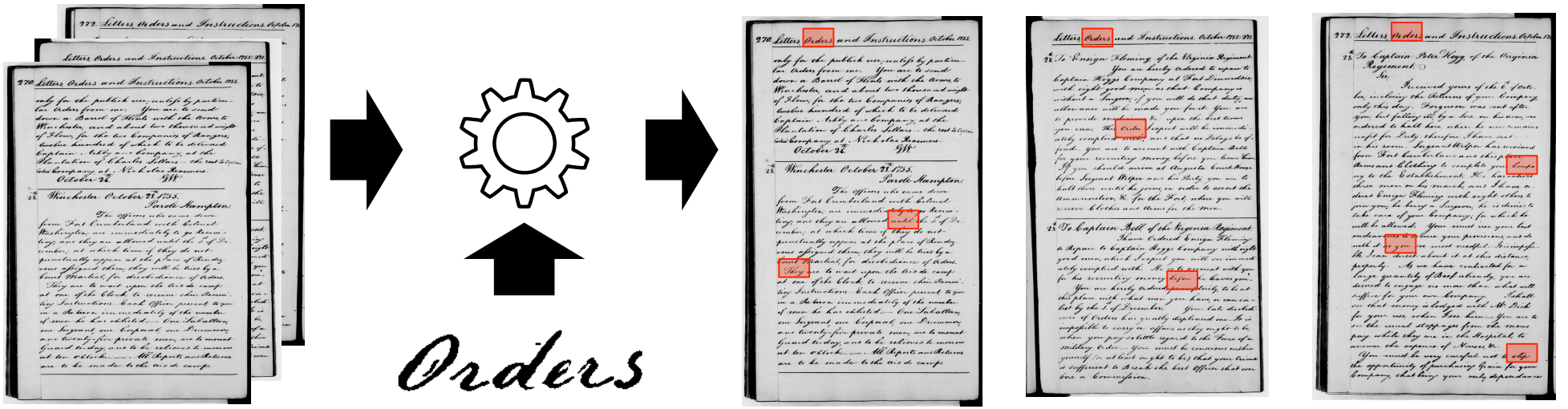
- Overview
- From Documents to Graphs
  - Image Preprocessing
  - Graph Representation
  - Graph Normalisation
  - Graph Matching
- Experiments
- Future Work
- Q+A

Overview – What is Keyword Spotting

Keyword Spotting (KWS) is the task of **retrieving any instance** of a given **query** word in **speech recordings** or **text images**.

➡ Focus on handwritten, historical documents

Document | Query | Word Spotting

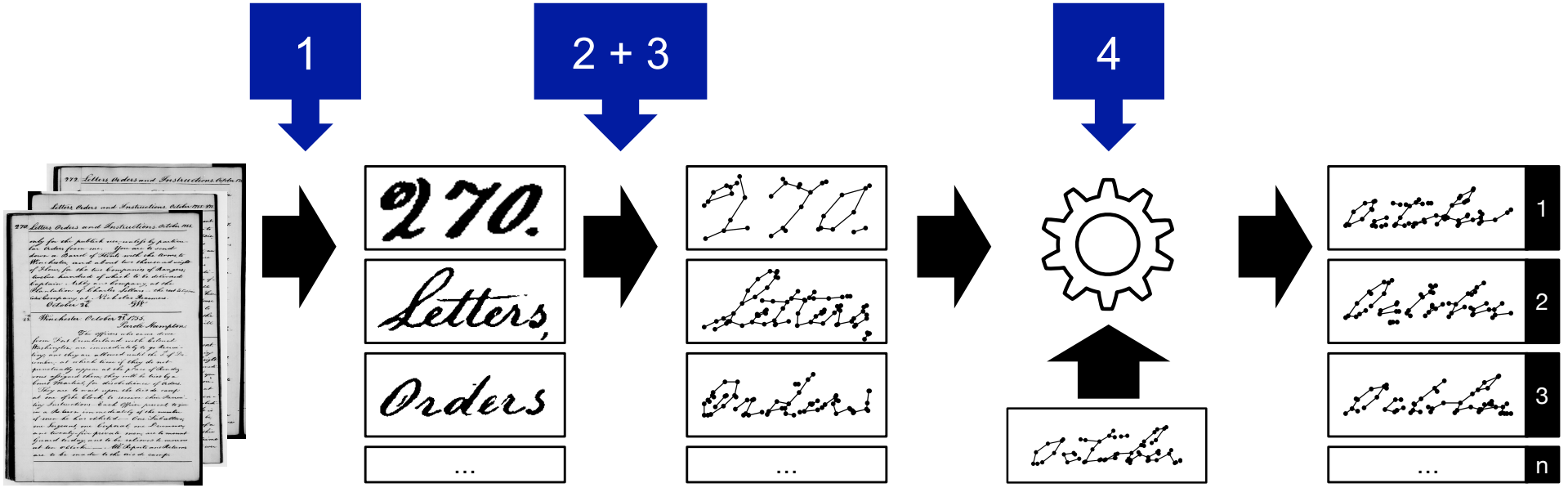


Process Legend

1) Image Preprocessing
2) Graph Extraction
3) Graph Normalisation
4) Pairwise Matching

Overview – From Documents to Graphs

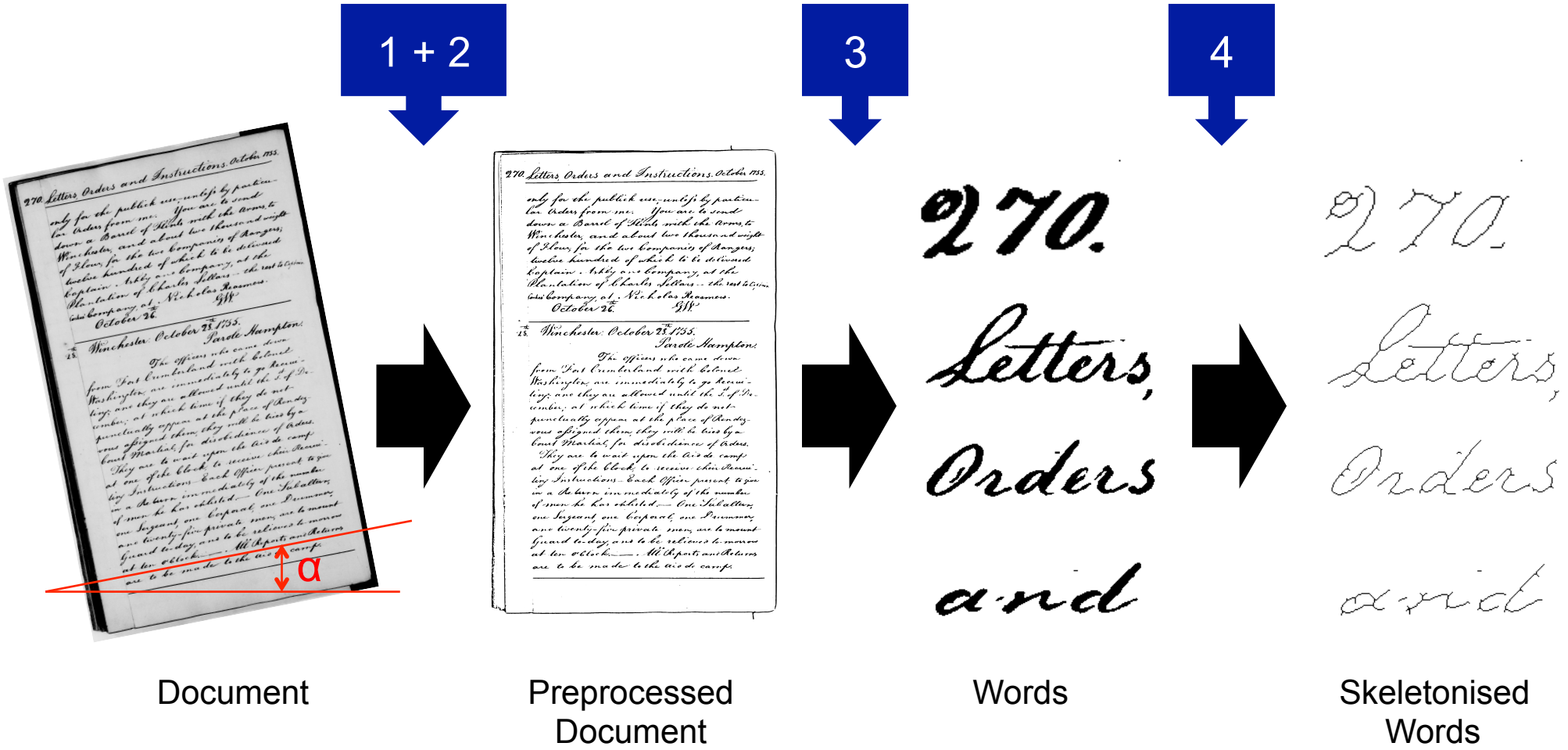
Graph-based KWS is based on the **representation of words** by means of different **graphs**. This representations are eventually used to **match a query** graph **against** all **document** graphs.



1) Image Preprocessing

Process Legend

1) Binarisation / Difference of Gaussian
2) Skew Correction (α)
3) Word Segmentation
4) Skeletonisation (optional)



2) Graph Extraction – Overview

Node Extraction
+
Edge Extraction

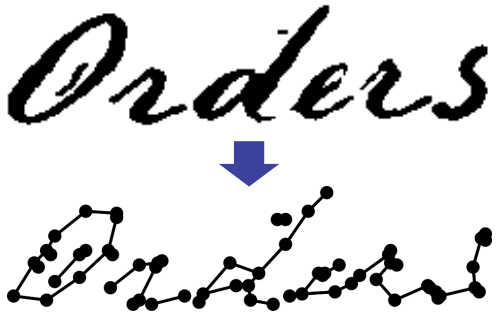
Keypoint 1 + 2
- Skeleton

Grid 1
- Neighbourhood (NNA)
- Minimal Spanning Tree (MST)
- Delaunay (DEL)

Projection 1
- Skeleton

Split 1
- Skeleton

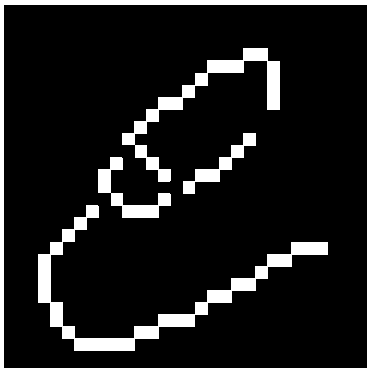
Papers
1) A Novel Graph Database for Handwritten Word Images
2) Graph-based Keyword Spotting in Historical Handwritten Documents



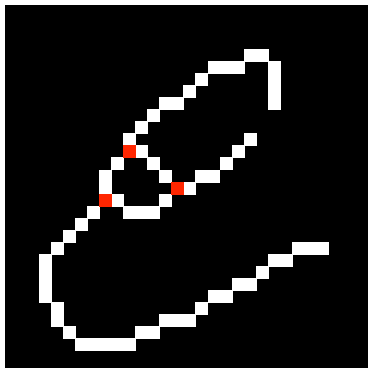
2) Graph Extraction – Keypoint (Node Extraction)

1) For each connected component…

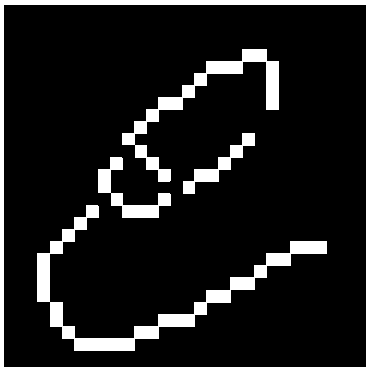
3) …invert junction points



2) …find junction points

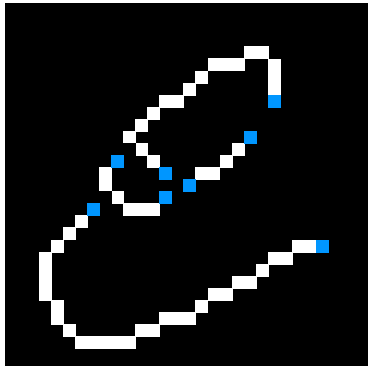


4) For each connected subcomponent…

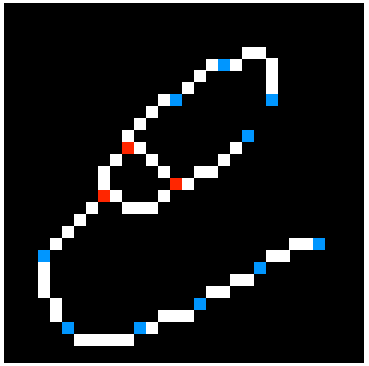


2) Graph Extraction – Keypoint (Node / Edge Extraction)

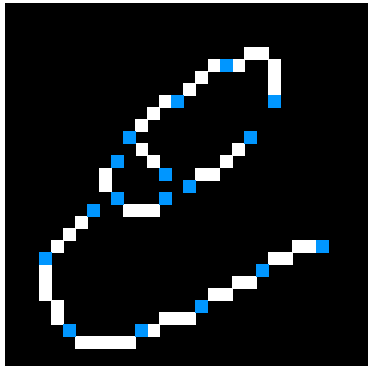
5) …find start- and end points



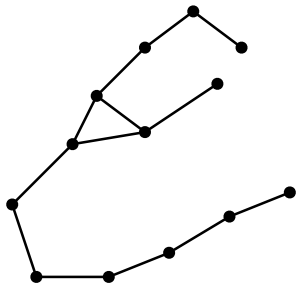
7) Replace neighbours of junction point

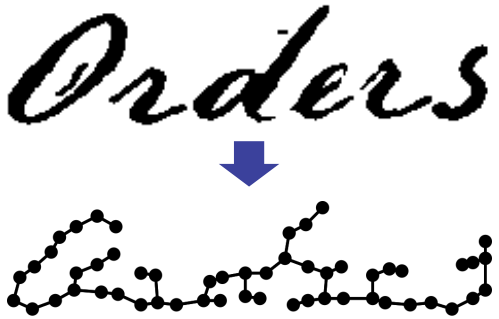


6) …add nodes along path



8) Add edges based on skeleton

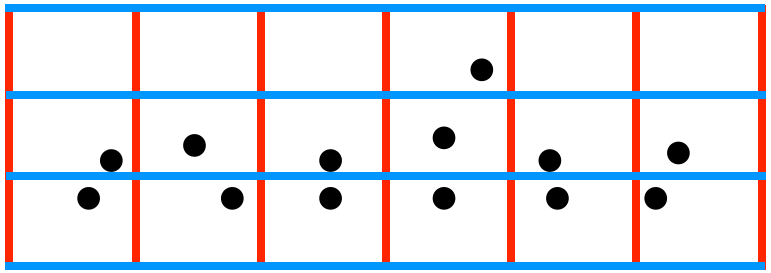




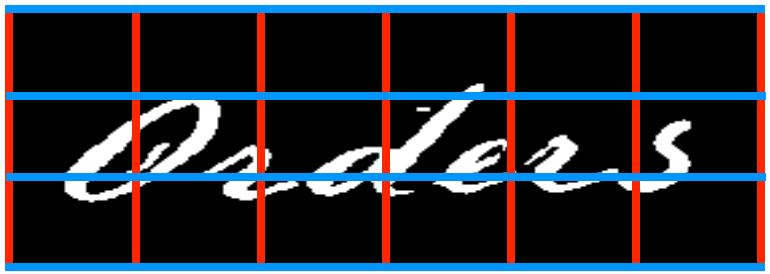
2) Graph Extraction – Grid (Node / Edge Extraction)

1) Add columns based on threshold

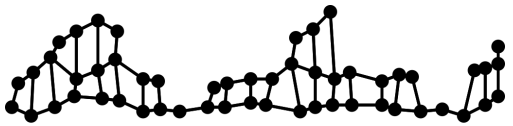
3) Add nodes based on centre of mass



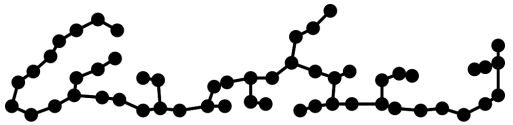
2) Add rows based on threshold



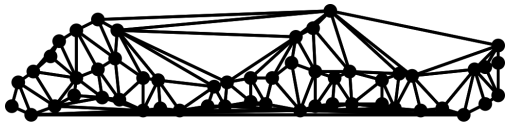
4) Add edges based on edge extraction



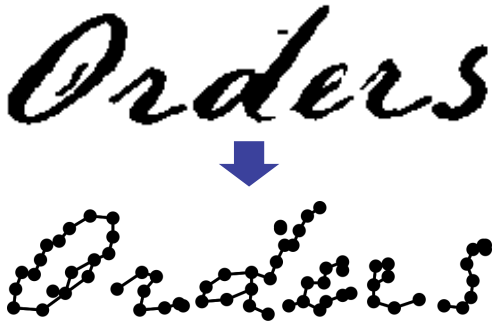
Neighbourhood Analysis (NNA)



Minimal Spanning Tree (MST)



Delaunay Tri-angulation (DEL)

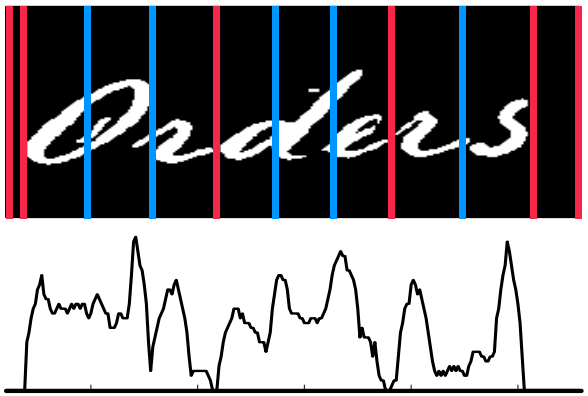


2) Graph Extraction – Projection (Node Extraction)

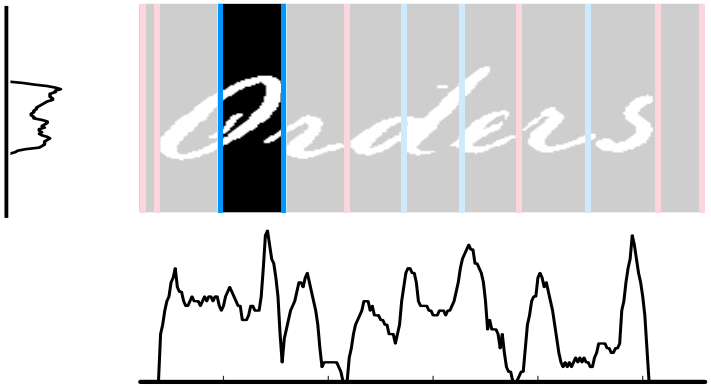
1) Vertical projection

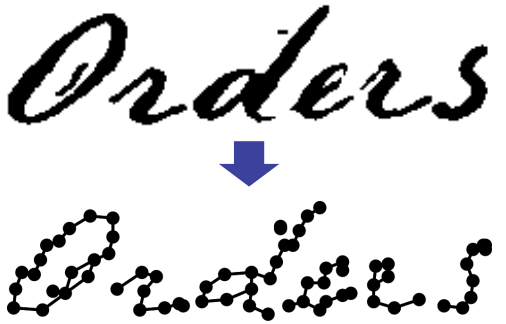
2) Add “null” positions as columns

3) Add further columns based on threshold



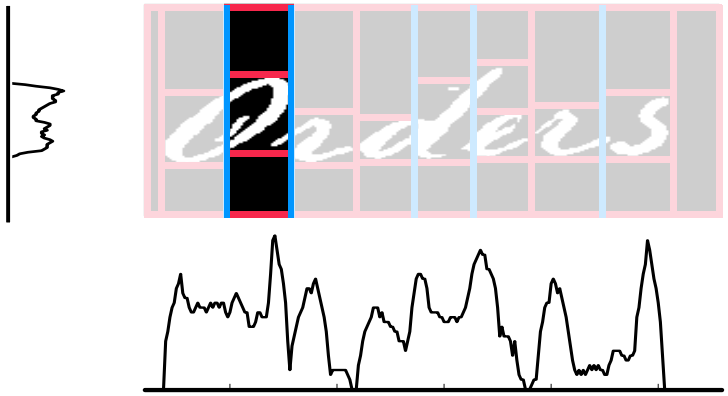
4) Horizontal projection per column



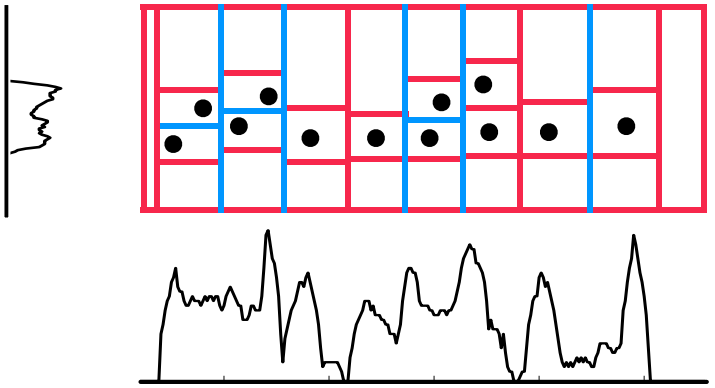


2) Graph Extraction – Projection (Node / Edge Extraction)

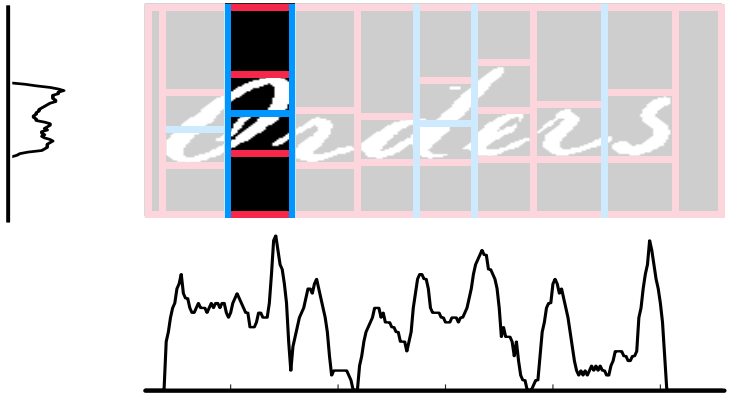
5) Add “null” positions as rows



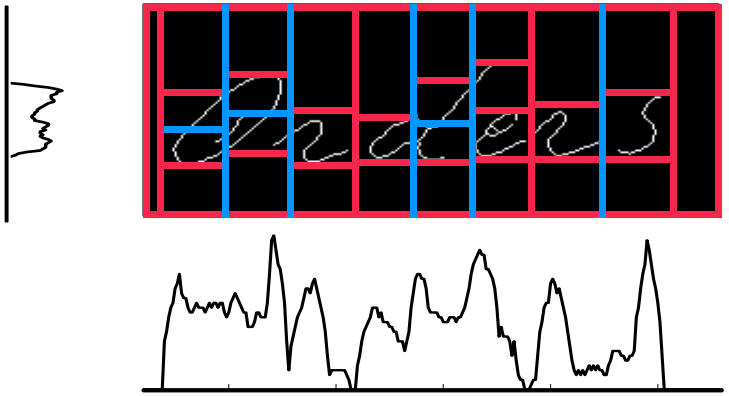
7) Add nodes based on centre of mass

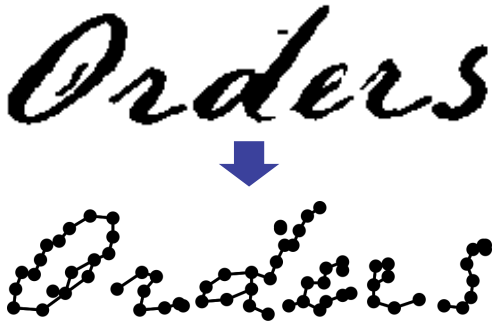


6) Add further rows based on threshold



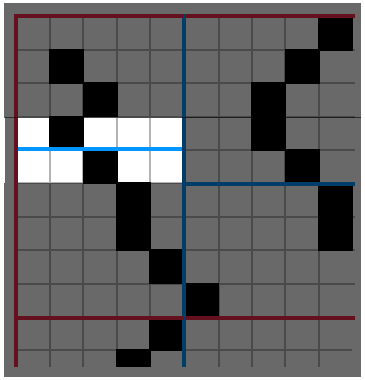
8) Add edges based on skeleton…



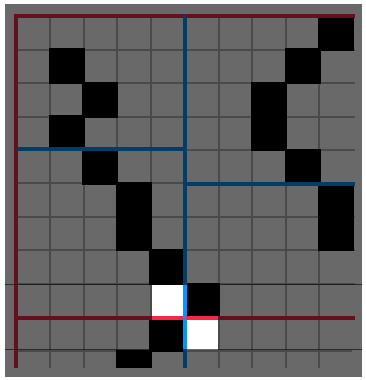


2) Graph Extraction – Projection (Edge Extraction)

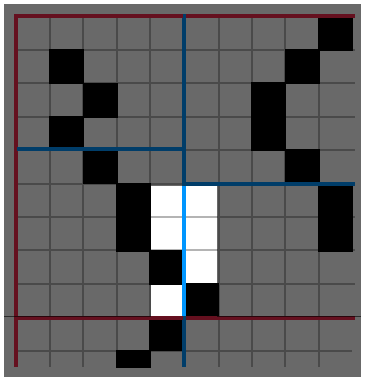
9) …vertical



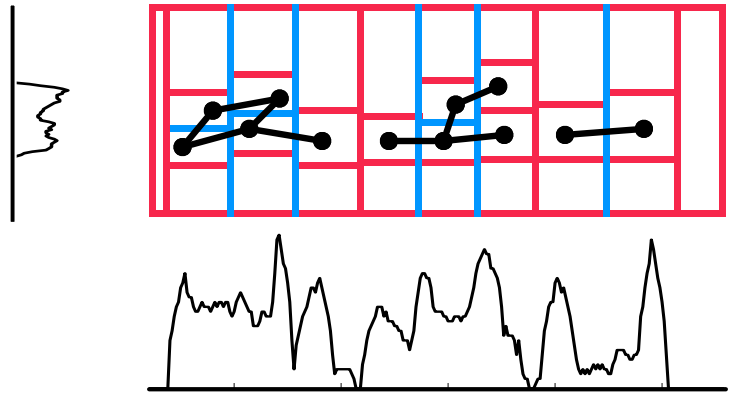
11) …diagonal

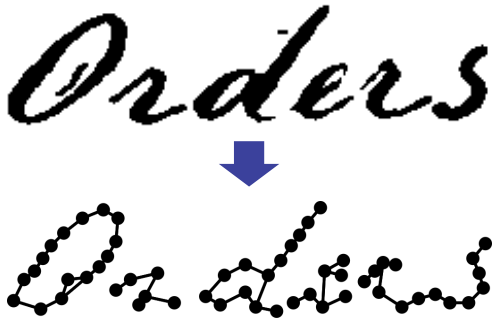


10) …horizontal



12) Complete graph





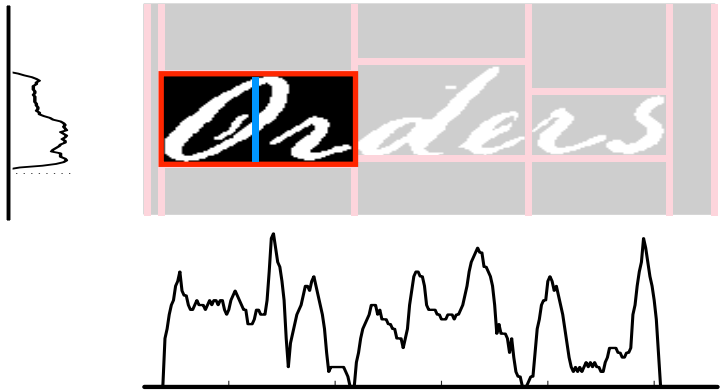
2) Graph Extraction – Split (Node Extraction)

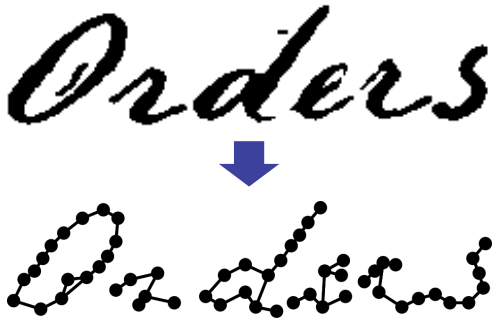
1) Split horizontal on null positions

2) Split vertical on null positions

3) For each subimage…

4) …too wide? Split horizontal

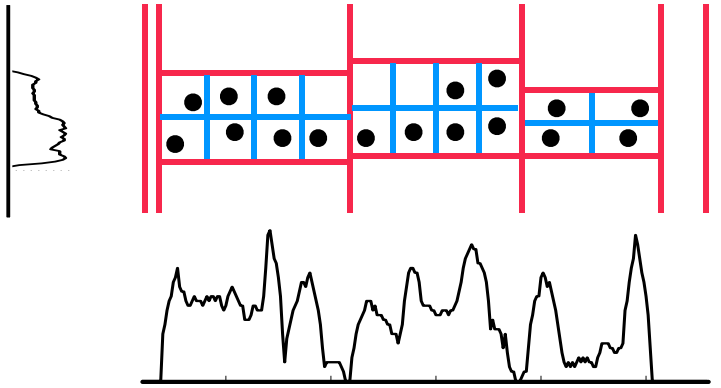




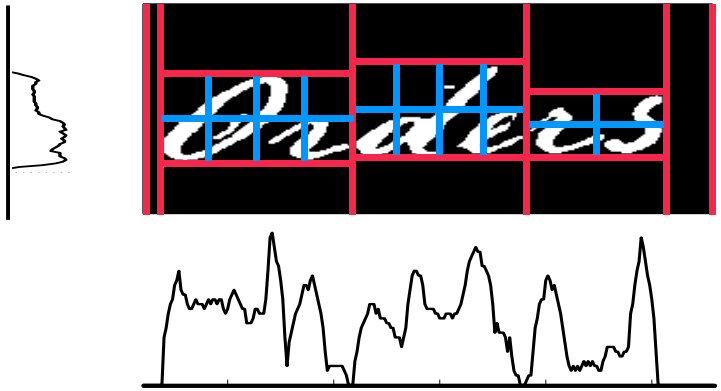
2) Graph Extraction – Split (Node / Edge Extraction)

5) …too high? Split vertical

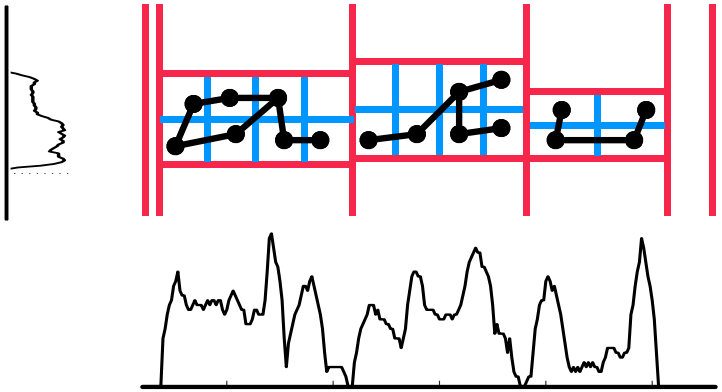
7) Add nodes based on centre of mass



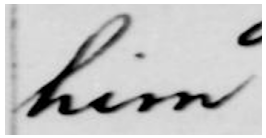
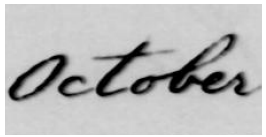
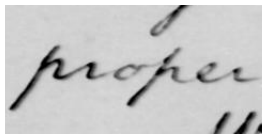
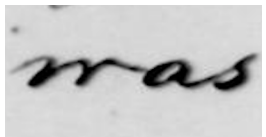
6) Split until subimages < threshold



8) Add edges based on skeleton



2) Graph Extraction – Resulting Graphs

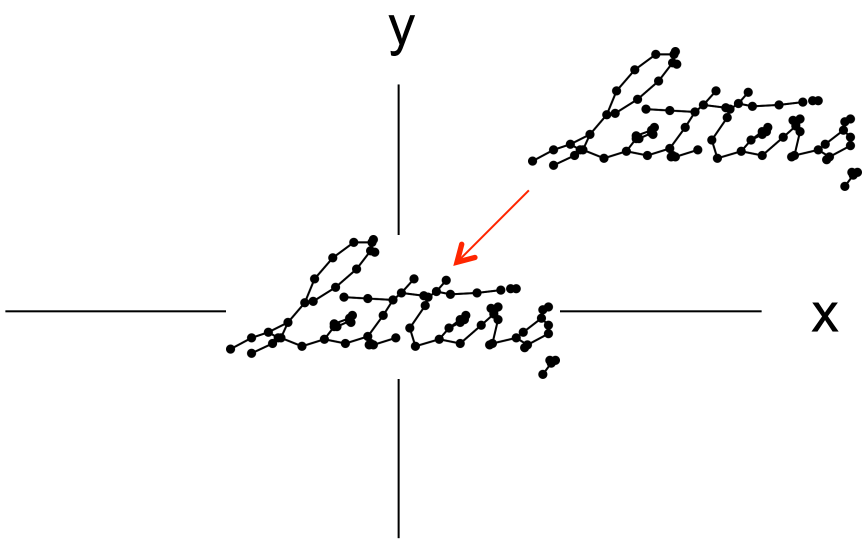
Word Image	Keypoint	Grid-NNA	Grid-MST	Grid-DEL	Projection	Split
						
						
						
						
						

3) Graph Normalisation – Size & Position of Nodes

Centering

$x_n = x - \mu_x$
$y_n = y - \mu_y$

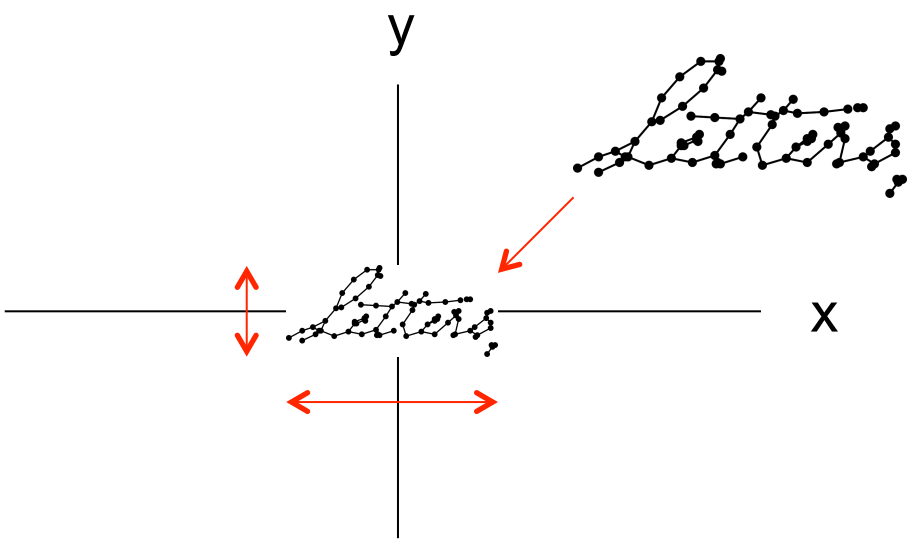
μ is mean of node positions



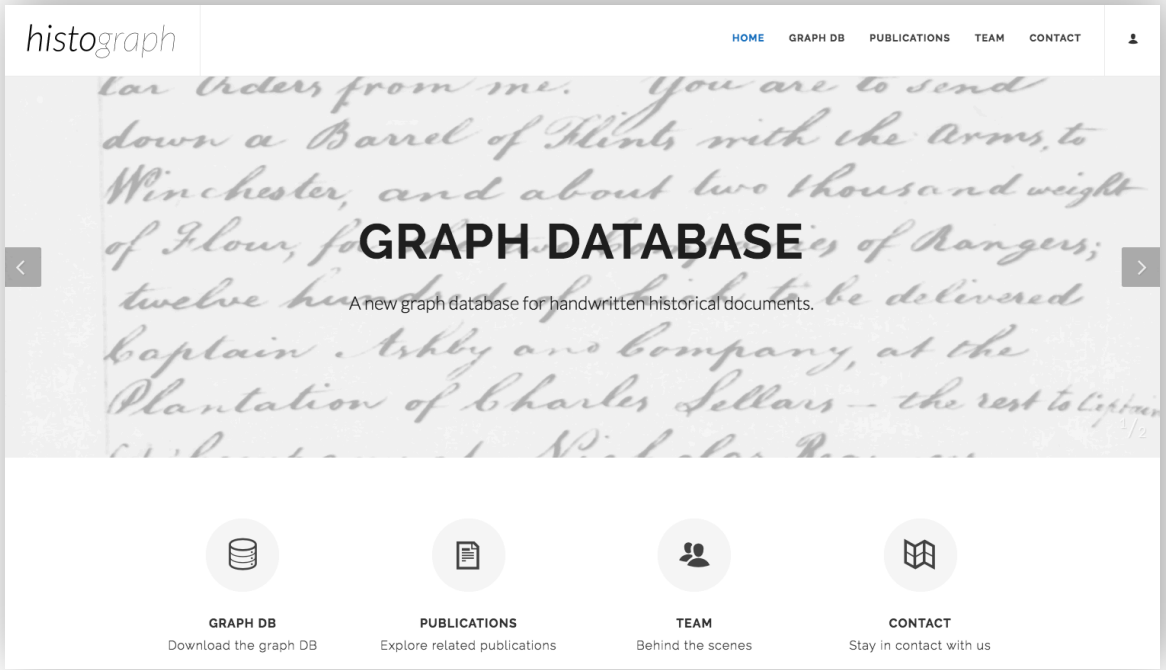
Centering & Scaling

$x_n = (x - \mu_x) / \sigma_x$
$y_n = (y - \mu_y) / \sigma_y$

σ is standard deviation of node positions



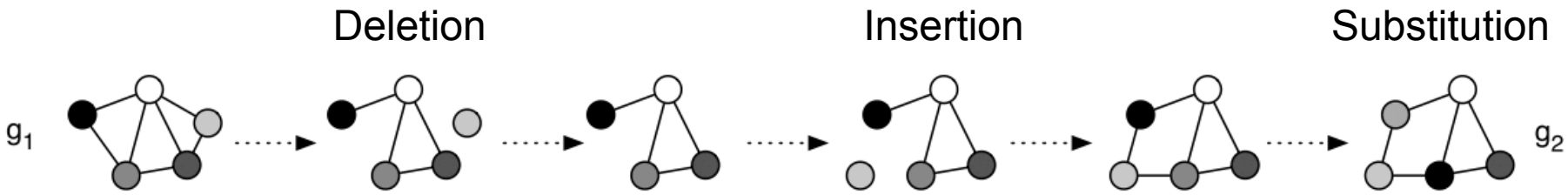
3) **Graph Normalisation** – Graph DB



All graphs are available at **histograph.ch**

4) Graph Matching – Graph Edit Distance

Graph Edit Distance (GED) measures the **minimum amount of distortion** that is needed to **transform one graph into another**.

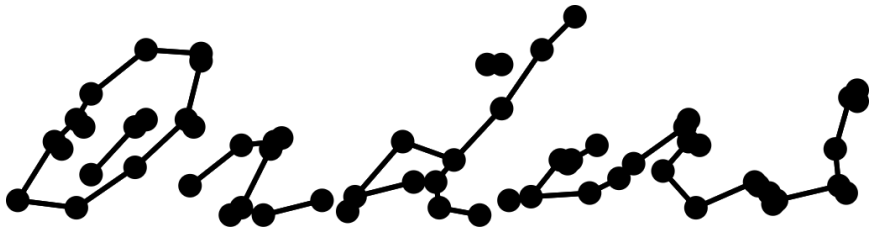


Exact GED is exponential in the number of nodes of the involved graphs and therefore intractable.

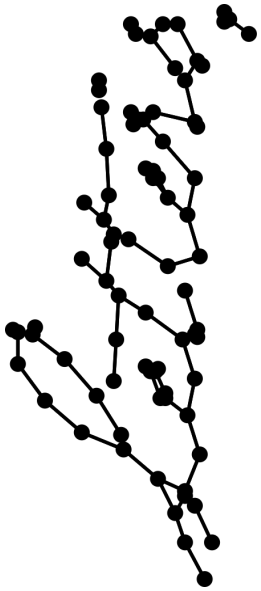
➡ We make use of approximated GED

4) **Graph Matching** – Bipartite Graph Matching

Target Graph



Source Graph



$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} & c_{1\varepsilon} & \infty & \cdots & \infty \\ c_{21} & c_{22} & \cdots & c_{2m} & \infty & c_{2\varepsilon} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \infty \\ c_{n1} & c_{n2} & \cdots & c_{nm} & \infty & \cdots & \infty & c_{n\varepsilon} \\ c_{\varepsilon 1} & \infty & \cdots & \infty & 0 & 0 & \cdots & 0 \\ \infty & c_{\varepsilon 2} & \ddots & \vdots & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \infty & \vdots & \ddots & \ddots & 0 \\ \infty & \cdots & \infty & c_{\varepsilon m} & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Substitution Costs

Deletion Costs

Insertion Costs

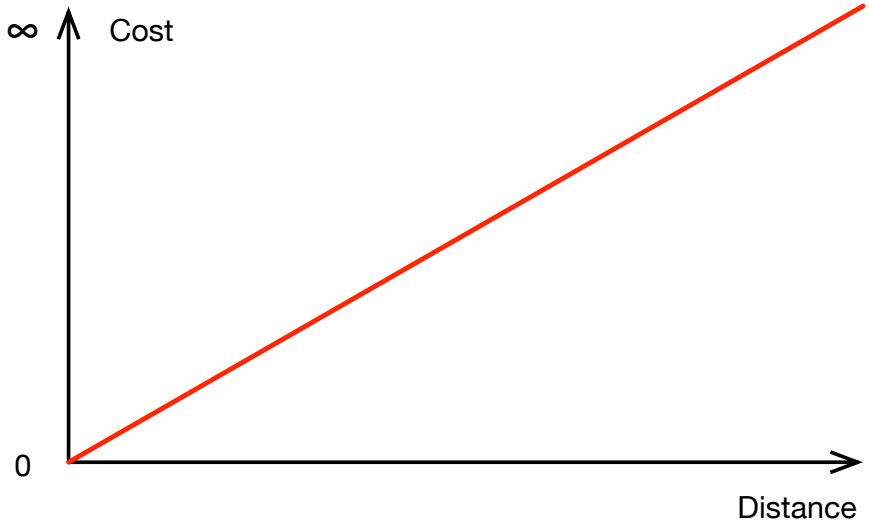
Cost Matrix with Node Edit Costs

Approximative GED via **L**inear **S**um **A**ssignment **P**roblem

4) **Graph Matching** – Substitution Cost Functions

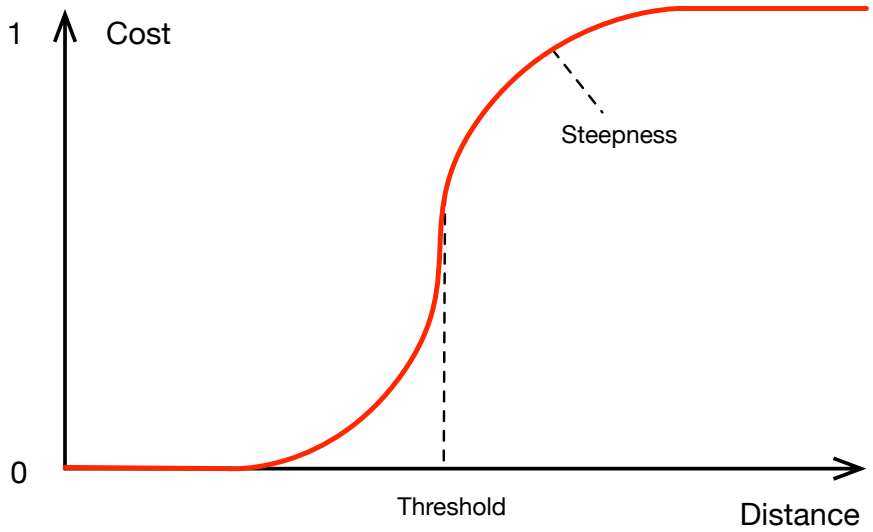
Euclidean $c_{Euclidean}$

$$c_E(u \to v) = \sqrt{\alpha(x_i - x_j)^2 + (1-\alpha)(y_i - y_j)^2},$$



Sigmoidal $c_{Sigmoidal}$

$$c_S(u \to v) = \frac{2\tau_v}{1 + e^{(kc_E(u \to v) - \gamma)}}$$



For deletion and insertion of nodes and edges we assume fixed costs

Abbreviations

|V| Median number of nodes

|E| Median number of edges

Experiments – Setup / Results Graph DB (kNN)

Set	Instances per Word Class	**Size**
Training	3	90
Validation	2	60
Test	3 - 5	143

Graph Extraction Algorithm	|V|	|E|	**Accuracy**
Keypoint	73	67	0.7762
Grid-NNA	39	55	0.6502
Grid-MST	46	44	0.7413
Grid-DEL	52	138	0.6294
Projection	44	41	0.8182
Split	51	48	0.8042

Experiments – Setup KWS

Set	Instances per Word Class	**Size**
Training	> 10	1000
Validation	1	10
Training	> 1	1223.5 (avg)
Test	> 1	2447.0 (avg)

➡ Quality is measured by the **mean Average Precision (mAP)**

Recall = True Positives / (True Positives + False Negatives)

Precision = True Positives / (True Positives + False Positives)

mAP = Average area under the curve of Precision-Recall curves

Abbreviations

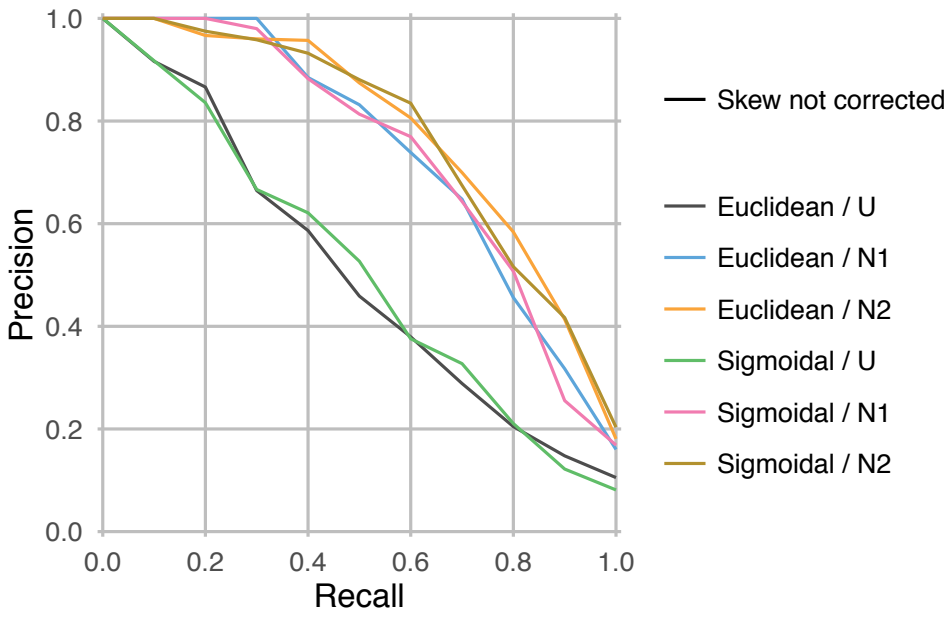
U Not normalised

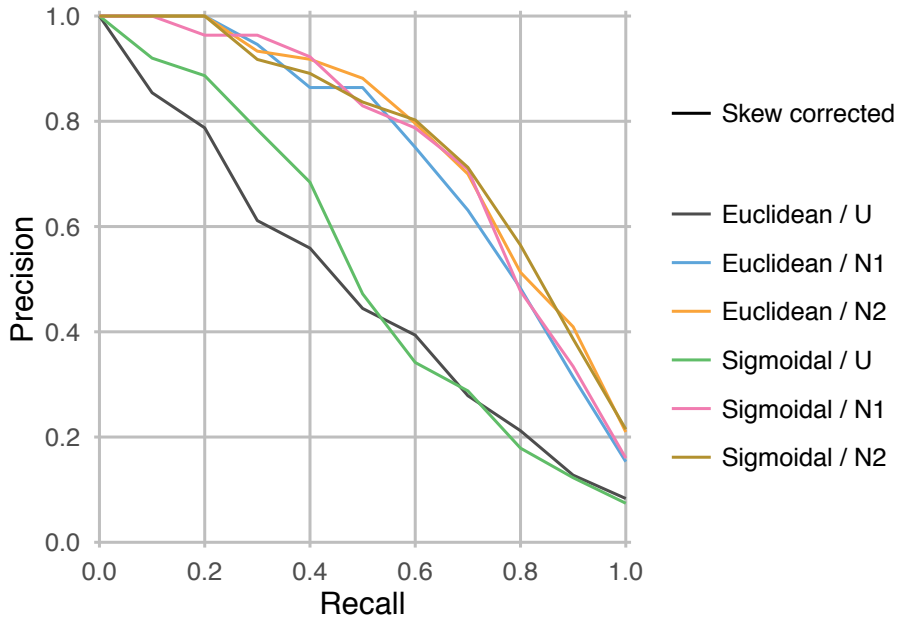
N_1 Centering

N_2 Centering + Scaling

Experiments – Results KWS (mAP) / Validation

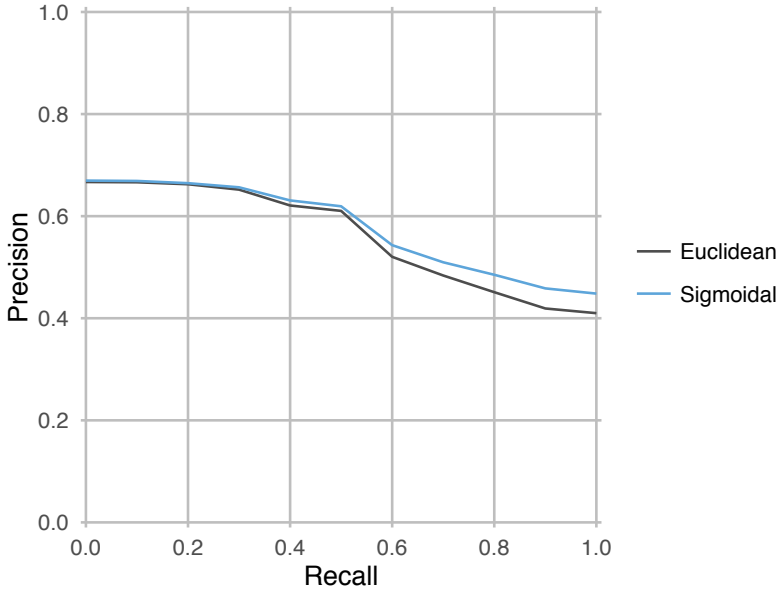
Preprocessing	Skew not corrected			Skew corrected		
Cost Function	U	N_1	N_2	U	N_1	N_2
$c_{Euclidean}$	50.17	72.87	**76.53**	47.08	72.24	75.59
$c_{Sigmoidal}$	49.71	72.72	**76.24**	50.60	73.53	75.24





Experiments – Results KWS (mAP) / Testing

System	mAP	**Improvement**
Dynamic Time Warping (DTW)	54.08	
Proposed $c_{Euclidean}$	55.33	**+ 2.31 %**
Proposed $c_{Sigmoidal}$	57.12	**+ 5.62 %**



Future Work

Database Extension

- Further documents (e.g. Parzival dataset)

Graph Representation

- Further node and edge labels

Graph Matching

- Speeding up graph matching procedure

Experiments

- Evaluate all graph extraction methods with KWS experiment

Q+A

?!