READ

RECOGNITION & ENRICHMENT
OF ARCHIVAL DOCUMENTS

D5.11
Page Image Explorer (PIE)

Markus Diem, Stefan Fiel, Florian Kleber
CVL

Distribution: http://read.transkribus.eu/

READ
H2020 Project 674943

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943

Project ref no.	H2020 674943
Project acronym	READ
Project full title	Recognition and Enrichment of Archival Documents
Instrument	H2020-EINFRA-2015-1
Thematic priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date/duration	01 January 2016 / 42 Months

Distribution	Public
Contract. date of delivery	31.12.2016
Actual date of delivery	28.11.2016
Date of last update	21.12.2016
Deliverable number	D5.11
Deliverable title	Page Image Explorer (PIE)
Type	report
Status & version	in progress
Contributing WP(s)	WP5
Responsible beneficiary	CVL
Other contributors	CVL
Internal reviewers	NAF, ASV
Author(s)	Markus Diem, Stefan Fiel, Florian Kleber
EC project officer	Martin MAJEK
Keywords	Document Clustering, Visualization

Contents

1 Executive Summary

The Page Image Explorer (PIE) allows intuitive exploration of documents. The key idea is to access potentially unsorted document collections and connect/group their items by user defined criteria. Hence, PIE strongly focuses on user interaction and visualization of large document collections. PIE will be built upon the READ Framework[1] which is publicly available under LGPLv3.

2 Prototype

A prototype for document clustering was created prior to READ [1]. This prototype is created using C++ and Qt. Since it does not use the GPU for rendering, only up to 5000 document images can be visualized at the same time.

The prototype uses different visual features for clustering similar documents. While PIE will use non-text elements such as images, diagrams, or tables for document image clustering, the prototype clusters document using text elements only. Figure 1 shows the prototype's visual features for document clustering. Text areas are labeled according to printed/handwritten text. If a page contains both, form analysis is performed which sorts documents with respect to predefined classes such as invoices or page indexes. Pages that contain mostly handwritten text are further analyzed. Hence, handwriting features are extracted which allow for clustering documents written by the same scribe. In addition to these text features, the background color and texture (ruled, checked) are extracted.

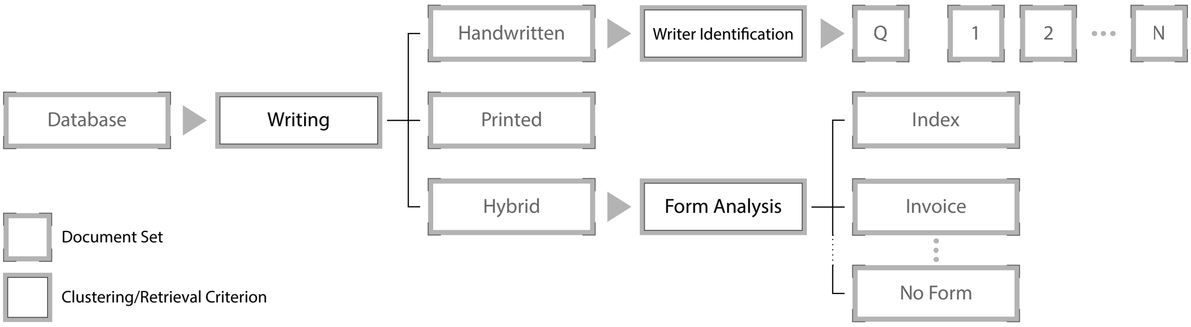

Figure 1: Illustration of different feature sets for hierachical clusteirng.

With all features cached in a database, the user can interactively group documents by choosing a combination of these features. To give an example, one can group all ruled A4 documents with 80% handwritten and 20% printed text. A hierarchical clustering approach allows the user to further split subsets retrieved with a coarse global clustering criteria.

[1] https://github.com/TUWien/ReadFramework

3 Visualization

While the previous prototype has only a visualization that shows thumbnails of document images grouped according to the currently selected cluster criteria, different visualizations are planned for PIE. The thumbnails are basically a 1D list which can be sorted with respect to user defined criteria. In contrast to this approach, visualization techniques that reduce high dimensional data to 2D such as the *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [2] will be used. This allows for demonstrating complex relationships between individual documents.

The visualization will utilize OpenGL viewports with Qt overpainting (for e.g. nice font rendering). Recent experiments conducted at CVL show that 32 viewports with 2 million dots each can be rendered in real-time using a standard Intel GPU[2]. Hence, PIE will scale better compared to its precursor.

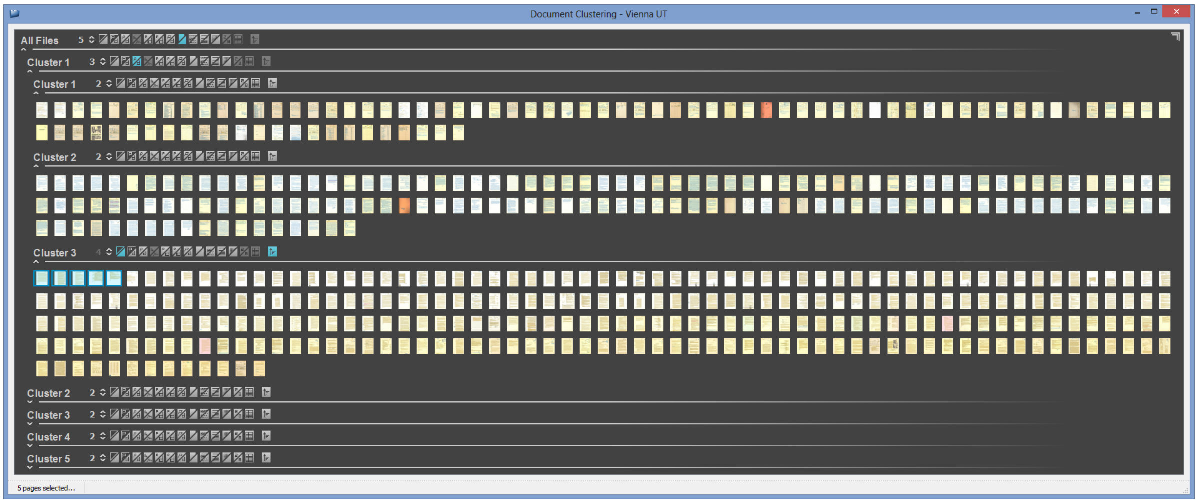

Figure 2: The first prototype's user interface.

The framework, which builds PIE's basis for reading results and clustering documents is developed and available on github.[3] Future work includes developing the front-end for 2D spacial clustering.

References

[1] Markus Diem, Florian Kleber, Stefan Fiel, and Robert Sablatnig, "Semi-Automated Document Image clustering and Retrieval," in *Proceedings of Document Recognition and Retrieval XXI*, Bertrand Coüasnon and Eric K. Ringger, Eds. SPIE, 2014, pp. 90 210M–1 – 90 210M–10.

[2] Intel HD Graphics 4000

[3] `https://github.com/TUWien/ReadFramework`

[2] Laurens van der Maaten, “Accelerating t-SNE using tree-based algorithms,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2697068