**WEI — Divadiawi**

<8 figures>

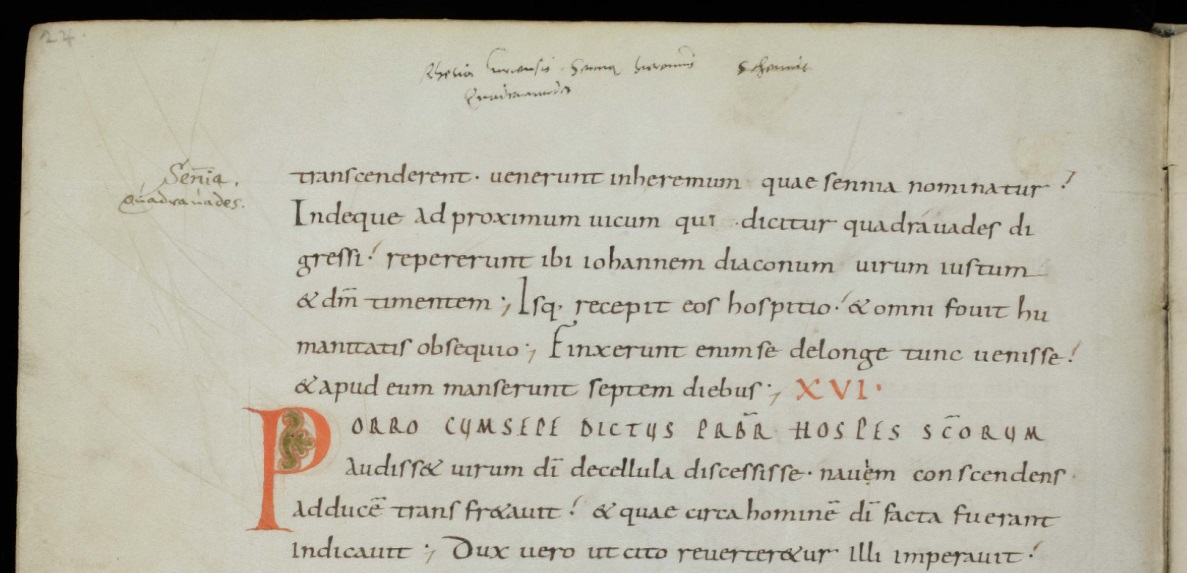
**DIVADIAWI: A Web-Based Interface for Semi-Automatic Labeling of Historical Document Images**

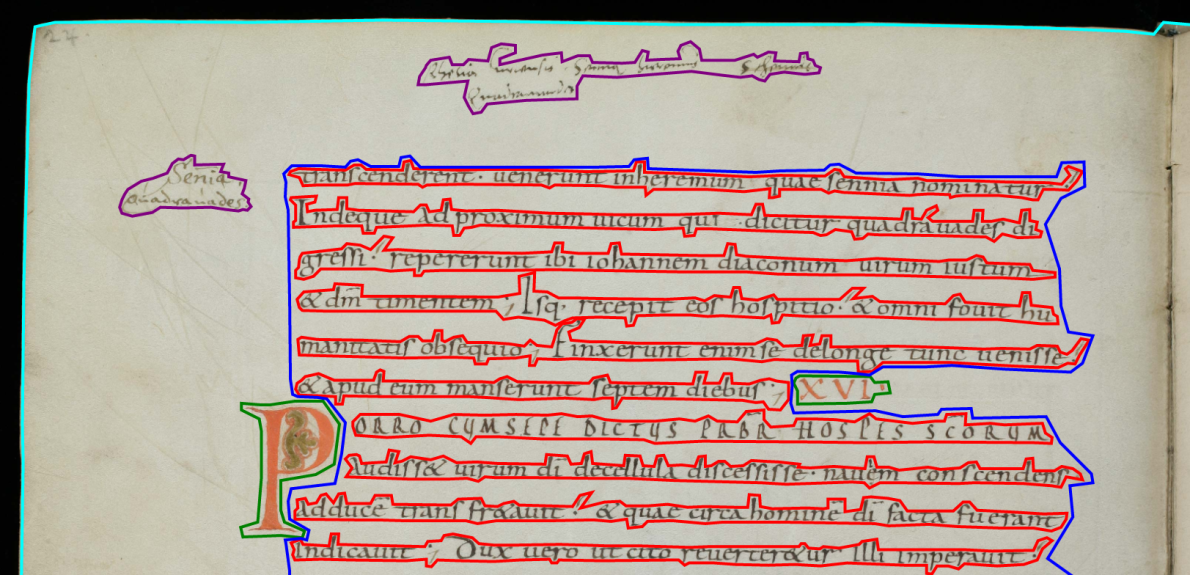
Wei, H., Chen, K., Seuret, M., Würsch, M., Liwicki, M. and Ingold, R.

According to the latest panel discussions at the Historical Image Processing (HIP 2013)1 and Document Analysis Systems (DAS 2014)2 conferences, historical documents are among the most challenging types of documents for automatic processing. This is due to the vast variety of challenges they pose to document image analysis (DIA) systems. In the pipeline of automatic DIA, layout analysis is an important prerequisite for further stages, such as optical character recognition and information retrieval. Layout analysis aims at splitting a document image into regions of interest and especially distinguishing text lines from other regions.

For automatic layout analysis methods, an essential prerequisite is the ground truth (GT), i.e., existing labels (text line, decoration, etc.) annotated by human experts for the corresponding regions. An example of historical document image3 and its ground truth are shown in Figure 1. The importance of GT is twofold: first, automatic methods could learn knowledge represented by GT from the experts, and then predict the layout of unseen images. Second, it is used to assess the performance of an automatic method by the accuracy of the prediction of the method against the GT. However, the generation of GT is time-consuming.

We propose a novel web-based interface called DIVADIAWI, in order to assist the user to semi-automatically generate ground truth for large numbers of historical documents. The name DIVADIAWI stands for a Web-based Interface for Document Image Analysis of DIVA group (Document, Image and Voice Analysis).4 DIVADIAWI incorporates two parts: automatic processing and manual editing. In the automatic processing part, the system automatically draws polygons representing text lines for which manual generation of GT is quite time-consuming. In parallel or sequentially, the user can manually edit the polygons by modifying them or directly drawing new polygons. DIVADIAWI greatly accelerates the generation of GT. It is robust to generate GT for historical document images of a diverse nature.



(a)

(b)

Figure 1. (a) An image example and (b) its ground truth. In (b), the cyan, blue, red, green, and purple polygons represent the regional contours of the page, text block, text lines, decorations, and comments, respectively.

**State of the Art**

In recent years, several tools for GT generation have been developed. Aletheia (Clausner et al., 2011) is an advanced system to generate GT for printed materials such as newspapers. It aids the user with a top-down method using split and shrink tools as well as a bottom-up method to group elements together. WebGT (Biller et al., 2013) is a web-based system for GT generation for historical document images. It provides a semi-automatic strategy enabling instant interaction between the user and the system. The Java-based DIVADIA tool (Chen et al., 2015) was developed by our group. It provides a user-friendly user interface to generate GT. The tool has been proven effective and flexible by performing a real task of GT generation. Other state-of-the-art tools include AGORA (Ramel et al., 2006), PixLabeler (Saund et al., 2009), TRUEVIZ (Lee et al., 2003), and GEDI (Doermann et al., 2010).

Compared to the state of the art, the novelty of DIVADIAWI is twofold. First, the high level of automation greatly reduces the human effort. The user needs to do only a few simple modifications after the automatic processing. Second, DIVADIAWI doesn’t require the user to install any software. It works on all modern browsers, e.g., Firefox, Chrome, and Internet Explorer.

**System Overview**

*Ground Truth Representation*

As it is shown in Figure 1(b), we define five types of regions of interest: *text line, text block, decoration, comment,* and *page* (Chen et al., 2015). The *text line* is the type of region where the main text is written. The *text block* incorporates text lines and document background between text lines. The *decoration* represents decorative elements such as figures, drop capitals, and decorative initials. The *comment* represents the annotations and inserted text in the margins. Finally the *page* outlines only the document part within the scanned image. The contours of the five regions are represented by polygons. All information—including the polygons, the document name, and the name of the author who generated the GT—are saved in XML files of the PAGE format (Pletschacher and Antonacopoulos, 2010), a widely used image representation framework for layout analysis. Furthermore, the GT also has the potential to be exported to the widely used TEI-P5 format for manuscript description5

*System Workflow*

DIVADIAWI is now publicly accessible at http://diuf.unifr.ch/diva/divadiawi/. A screenshot of DIVADIAWI is shown in Figure 2. DIVADIAWI integrates automatic processing and manual editing. The workflow is illustrated in Figure 3 and detailed below.

Among the five region types, the regions of text lines cost most of the working time for GT generation.6 The automatic processing of DIVADIAWI is developed to accelerate the GT generation specifically for text lines. The user needs to manually draw a rectangular polygon representing a text block that contains some text lines. As soon as the rectangular polygon is drawn, the automatic processing starts. We use Gabor filters, widely used for text extraction (Jain and Farrokhnia, 1991; Raju et al., 2004), to detect regions of text lines from the rectangle. After being detected, text lines are represented by polygons and are then drawn. The automatic processing performs well on our datasets. Most of the regions of text lines are properly outlined. However, there are still some minor mistakes. In small areas where two adjacent text lines intersect or are very close to each other, the two text lines are wrongly grouped into a single line in some cases. In addition, for some strokes lightly written near the boundary of a text line, automatic processing fails to detect them. Thus manual modification follows.

The manual editing has two modes: manual drawing and manual modification. Since automatic processing specifically deals with text lines, the user needs to manually draw regions of the page, text blocks, decorations, and comments. The user can select the shape (polygon or rectangle) and region type to draw. The drawn polygons and rectangles have different colors depending on their region types. After drawing, the user can modify polygons or rectangles by dragging a vertex to a new position. A new vertex can also be added to a polygon. In addition, a polygon can be deleted. Finally, an XML file storing the GT of the page is generated.

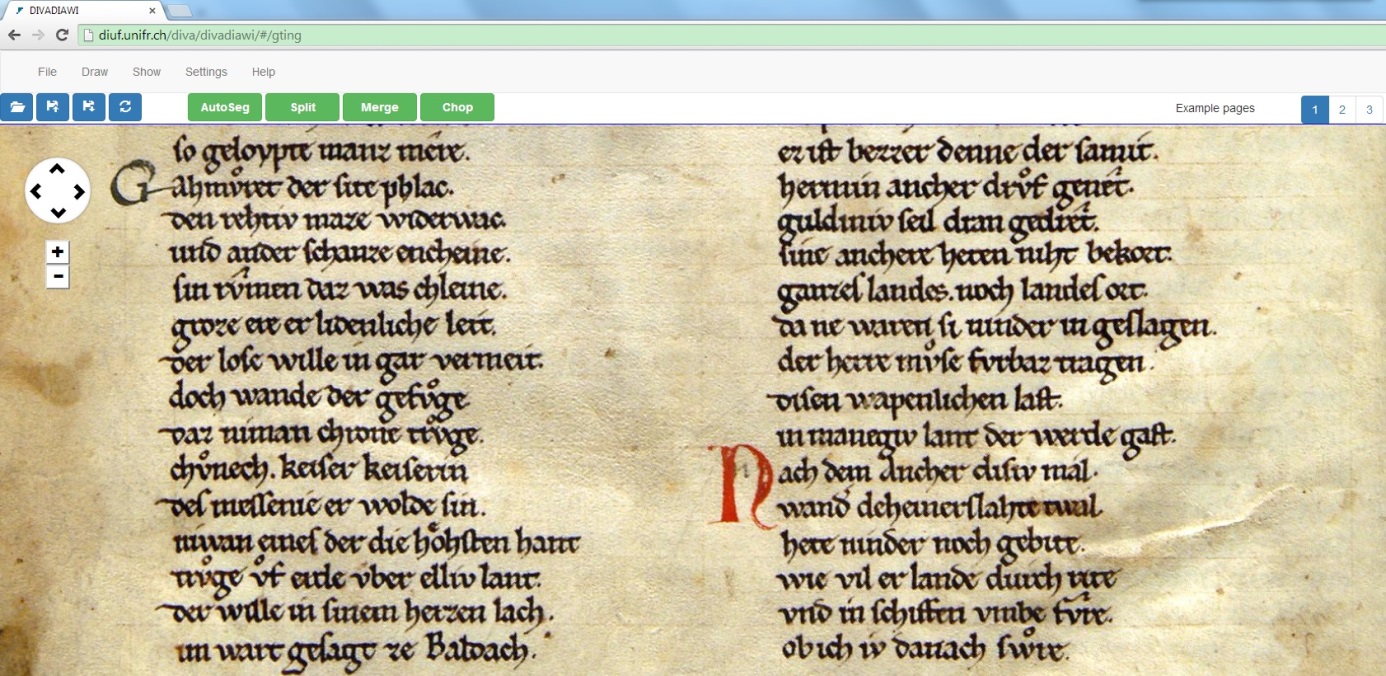


Figure 2. A screenshot of DIVADIAWI.

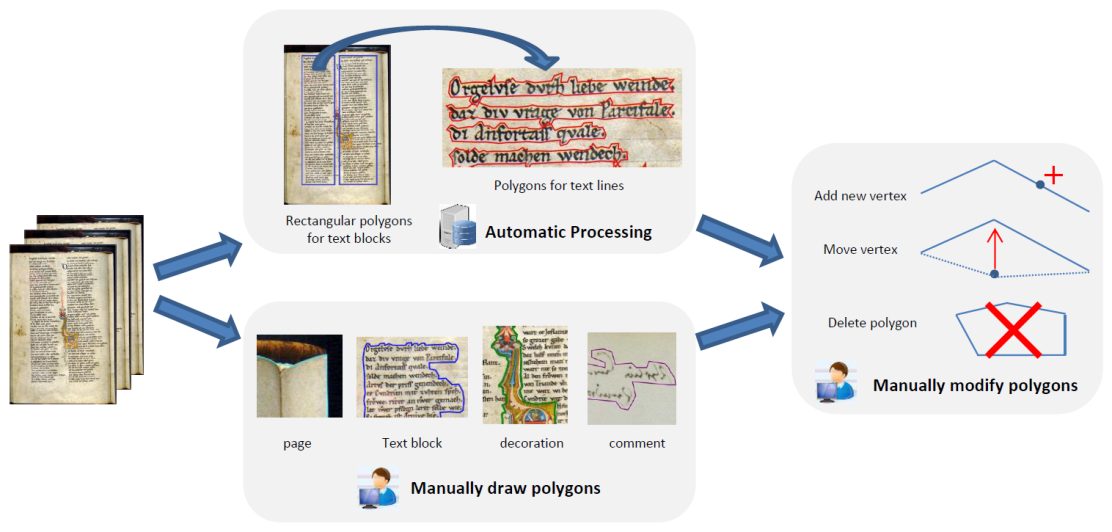
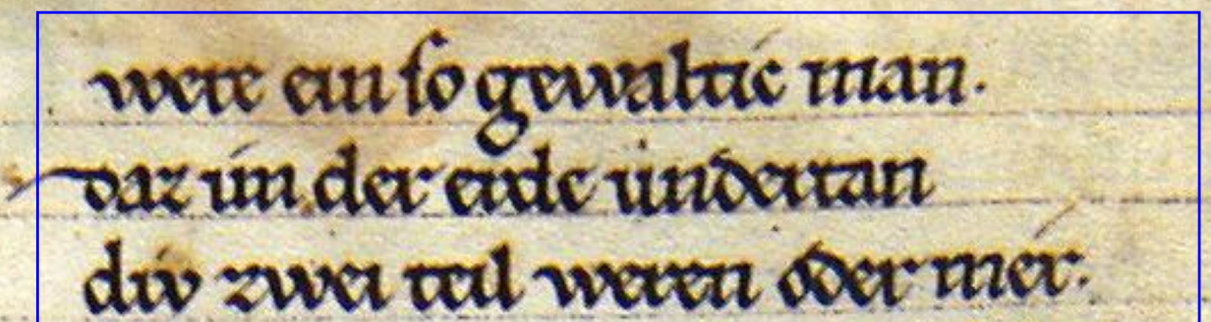
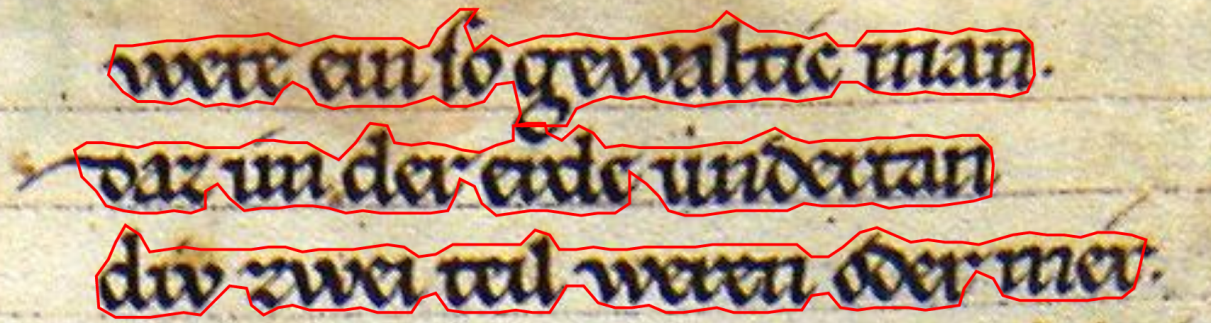


Figure 3. System workflow of DIVADIAWI.

(a)



(b)

(c)

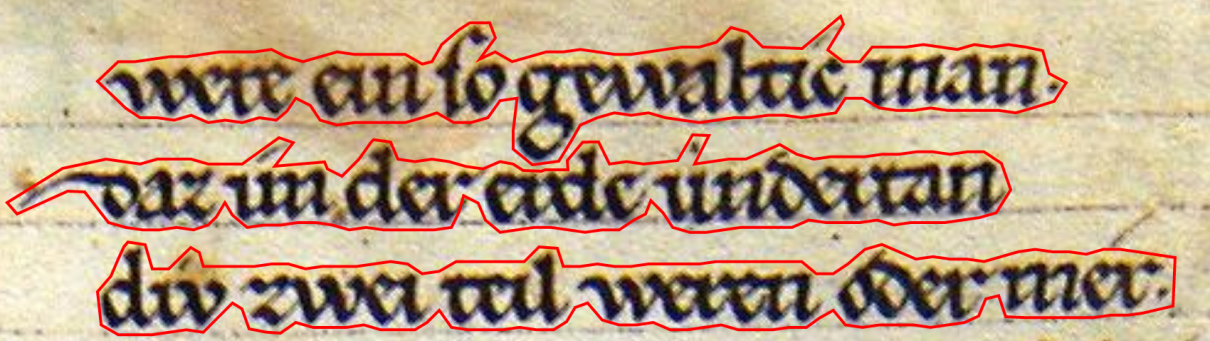
(d)

Figure 4. The process to draw polygons of text lines by automatic processing and manual modification. (a) A part of original image. (b) Rectangular polygon of text block drawn. manually by the user. (c) Polygons of text lines obtained by automatic processing. (d) Final result obtained by manual modification.

**Evaluation**

IAM Historical Document Database (IAM-HistDB)7 is used for the evaluation. The database contains a variety of historical document images, which are color or gray, single-column or double-column, and date from different centuries.

DIVADIAWI works well on the database. Thanks to the automatic processing, it greatly accelerates the GT generation. We show an example to draw polygons of text lines on a color image8 in Figure 4. Most pixels within regions of text lines are included in the polygons obtained by automatic processing (see Figure 4[c]). However, the result of this step is still not perfect. For example, the bottom part of the big character ‘G’ in the first text line is very close to the second text line, leading automatic processing to fail to separate them. The problems could be solved by manually dragging the problematic vertexes of the polygons to proper positions (see Figure 4[d]). In general, with the automatic processing, the user needs to do only a few simple manual modifications for each text line. This manual work is much less effort than completely manual editing from scratch.

System Usability Scale (SUS; Brooke, 1996), a measurement to assess the global performance of system usability, is used to assess DIVADIAWI. Researchers from the humanities and computer sciences are both invited to participate in the assessment. Quantitative evaluation details will be provided in the full paper.

**Conclusions and Future Work**

We propose DIVADIAWI, a web-based interface for the GT generation for historical document images. DIVADIAWI is efficient to accelerate the GT generation, thanks to the integration of automatic processing and manual editing. It is robust to work on different kinds of historical document images from the IAM-HistDB.

In the future we will keep improving DIVADIAWI. For example, automatic processing could be better designed to even decrease errors. Techniques to automatically outline other region types—e.g., page and text block—could also be applied.

**Notes**

1. Workshop on Historical Document Imaging and Processing, 2013, http://www.cvc.uab.es/~vfrinken/hip2013/.

2. 11th IAPR International Workshop on Document Analysis Systems, 2014, http://das2014.sciencesconf.org/.

3. Saint Gall DB: Cod. Sang. 562, Abbey Library of St. Gall (SG30).

4. http://diuf.unifr.ch/main/diva/.

5. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/MS.html.

6. In a process of manual GT generation of 100 document pages, about 80% of the time was spent on text lines.

7. http://www.iam.unibe.ch/fki/databases/iam-historical-document-database.

8. Parzival DB: Cod. 857, Abbey Library of St Gall (PAR23).

**References**

**Biller, O., Asi, A., Kedem, K., El-Sana, J. and Dinstein, I.** (2013). WebGT: An Interactive Web-based System for Historical Document Ground Truth Generation. *12th International Conference on Document Analysis and Recognition*, pp. 305–8.

**Brooke, J.** (1996). SUS—A Quick and Dirty Usability Scale. Usability Evaluation. In Jordan, P. W., Weerdmeester, B., Thomas, A. and McLelland, I. L. (eds), *Industry*. Taylor and Francis.

**Chen, K., Seuret, M., Wei, H., Liwicki, M., Hennebert, J. and Ingold, R.** (2015). Ground Truth Model, Tool, and Dataset for Layout Analysis of Historical Documents. *Proceedings of SPIE 22nd Document Recognition and Retrieval XXII,* vol. 9402.

**Clausner, C., Pletschacher, S. and Antonacopoulos, A.** (2011). Aletheia—An Advanced Document Layout and Text Ground-Truthing System for Production Environments. *International Conference on Document Analysis and Recognition*, pp. 48–52.

**Doermann, D., Zotkina, E. and Li, H.** (2010). GEDI—A Groundtruthing Environment for Document Images. *Ninth IAPR International Workshop on Document Analysis Systems (DAS 2010)*.

**Jain, A. K. and Farrokhnia, F.** (1991). Unsupervised Texture Segmentation Using Gabor Filters. *Pattern Recognition,* **24**(12): 1167–86.

**Lee, C. and Kanungo, T.** (2003). The Architecture of TRUEVIZ: a Groundtruth/Metadata Editing and Visualizing Toolkit. *Pattern Recognition,* **36**(3): 811–25.

**Pletschacher, S. and Antonacopoulos, A.** (2010). The PAGE (Page Analysis and Ground-truth Elements) Format Framework. *International Conference on Pattern Recognition,* pp. 257–60.

**Raju, S. S., Pati, P. B. and Ramakrishnan, A. G.** (2004). Gabor Filter Based Block Energy Analysis for Text Extraction from Digital Document Images. *First International Workshop on Document Image Analysis for Libraries,* pp. 233–43.

**Ramel, J. Y., Busson, S. and Demonet, M. L.** (2006). AGORA: The Interactive Document Image Analysis Tool of the BVH Project. *Second International Conference on Document Image Analysis for Libraries,* pp. 145–55.

**Saund, E., Lin, J. and Sarkar, P.** (2009). PixLabeler: User Interface for Pixel-Level Labeling of Elements in Document Images. *International Conference on Document Analysis and Recognition,* pp. 646–50.