

Creating Ground Truth for Historical Manuscripts with Document Graphs and Scribbling Interaction

Angelika Garz*, Mathias Seuret*, Fotini Simistira*, Andreas Fischer*[‡] and Rolf Ingold*

*University of Fribourg, 1700 Fribourg, Switzerland, Email: {firstname.lastname}@unifr.ch

[‡] iCoSys Institute, University of Applied Sciences and Arts Western Switzerland, 1700 Fribourg, Switzerland

Abstract—Ground truth is both – indispensable for training and evaluating document analysis methods, and yet very tedious to create manually. This especially holds true for complex historical manuscripts that exhibit challenging layouts with interfering and overlapping handwriting. In this paper, we propose a novel semi-automatic system to support layout annotations in such a scenario based on document graphs and a pen-based scribbling interaction. On the one hand, document graphs provide a sparse page representation that is already close to the desired ground truth and on the other hand, scribbling facilitates an efficient and convenient pen-based interaction with the graph. The performance of the system is demonstrated in the context of a newly introduced database of historical manuscripts with complex layouts.

I. INTRODUCTION

Ground Truth (GT) for historical manuscripts is critical for the development of learning-based methods for all steps in an automatic document image processing pipeline: layout analysis, text recognition, word spotting, writer identification, as well as document understanding. These methods facilitate the integration of historical manuscripts in digital libraries that allow for full-text search, clustering and retrieval tasks, establishing interrelations between manuscripts held by different physical libraries.

As a consequence, numerous large-scale projects have been launched devoted to the problems associated with digital libraries, such as DEBORA [1], the European project IMPACT [2], or the Bovary project [3]. In particular, the task of GT creation for historical documents has been addressed in literature abundantly: examples include Aletheia [4] for historical printed documents, PhaseGT [5] for binarization of historical document images, WebGT [6], a web GT interface, the DMOS system [7] for old civil status registers and military forms, and the IAM-HistDB system [8] for old German manuscripts.

However, most existing systems have focused on highly structured manuscripts and those with plain, uncluttered text column layouts. Yet handwritten historical documents frequently bear more challenging layouts, making it tedious for human users to annotate. In this paper, we address such a challenging case by considering commentary notes between lines and in the page margins that interfere with the main text, making the layout a disarray of intertwined text (c.f. Figure 1). Further challenges are touching characters and overlapping decorations and initials. For annotating the layout of these documents we cannot rely on well-structured text blocks that can be decomposed into lines by segmentation methods such as dynamic programming [8]. Our

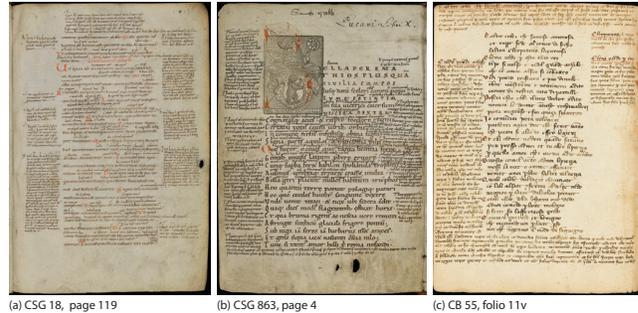


Figure 1. Sample pages of the three datasets in the DIVA-HistDB

goal is to extract the main text, the glossary annotations added to the document, and the decorations. The level of interest for text hereby are lines since they are the typical input for a range of methods such as handwriting recognition, keyword spotting, and writer identification. Decoration elements such as initials are interesting for image retrieval tasks in order to obtain paleographic information like date, or origin, or to draw conclusions about interrelations of different documents.

A basic approach toward this goal would be to make entirely manual annotations in form of bounding boxes, point-to-point polygons or freehand-drawings around the layout elements. Clearly, this simple approach is not efficient and rather tiresome. Instead, we propose a novel semi-automatic GT system that relies on sparse graphs, which capture a document’s structure in a way not unlike human perception, and a pen-based user interaction pattern.

Graphs have found widespread application in pattern recognition because of their ability to represent parts of an object as well as binary relationships [9], [10]. Different fields of research that leverage graph-based representation include bioinformatics [11], computer network analysis [12], and image classification [13] to name just a few. In the field of document image analysis, traditional applications are the recognition of handwritten Chinese characters [14] and the recognition of symbols in hand drawn diagrams [15], where graphs are used to model structural relationships among handwritten strokes. Recent contributions include graph-based approaches for cursive handwriting recognition [16], keyword spotting in historical manuscripts [17], [18], image retrieval in comics [19], and text line segmentation [20].

For graph-based GT creation, we create a sparse graph representation of a document’s foreground. The foreground

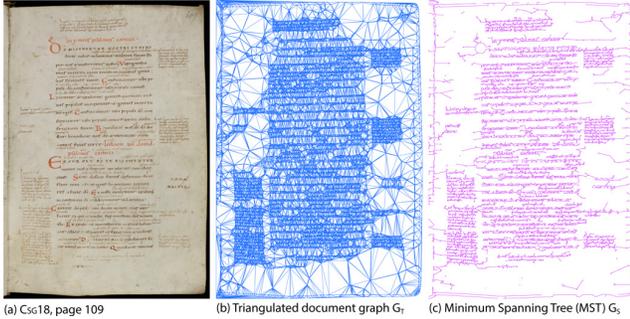


Figure 2. A sample document graph and its MST that clusters the layout elements.

estimation of a document follows the assumption that strong gradient changes indicate its presence [21]. We then extract points along those gradient changes as graph nodes. The graph’s edges are obtained from triangulating its nodes, and finally, we employ a Minimum Spanning Tree (MST) to cluster the graph into structures that constitute layout elements providing a sparse structural representation of the document (c.f. Figure 2 for an example).

This graph facilitates the efficient use of a pen-based interaction, namely scribbling, as a swift and natural user interaction to correct the graph. In particular, the user inserts and deletes graph edges in order to build text lines and decoration elements. We follow the idea of allowing the user to do imprecise interactions in a manner that is natural to him or her, while the precise segmentation is retrieved from the graph. The layout elements created by the user are then represented as non-convex polygons, which can efficiently be extracted from the graph, and stored in the PAGE XML format.

In order to evaluate the proposed GT annotation system, we apply it to three medieval manuscripts, two Latin manuscripts from the Abbey Library of Saint Gall and one Italian manuscript from the Foundation Bodmer, which exhibit challenging layouts. Both, the graph’s capability of capturing the desired document structure, and the efficiency of user interaction are evaluated. They can serve as baseline results for future comparisons. The resulting DIVA-HistDB database for historical manuscripts with challenging layouts along with its associated GT will be made publicly available for this purpose.

II. DIVA-HISTDB

We demonstrate the GT creation process on the new DIVA Historical Manuscript Database (DIVA-HistDB)¹, which is a collection of three challenging medieval manuscripts. The complexity of the manuscripts lies in their layout: while a relatively regular main text prevails, it is made complex by so-called glosses. Glosses are brief notes which illustrate the main text; they can be in the margin of a page (marginal gloss) or between lines of the main text (interlinear gloss). Clearly, a traditional text-block-based approach towards these documents is not feasible. Figure 1 shows three sample pages from each

¹The database as well as the tool will be made publicly available on the project page <http://diuf.unifr.ch/hisdoc2>

Table I
DIVA-HISTDB: SPECIFICATIONS DATASETS WITH THEIR SHORT NAME, CENTURY OF ORIGIN, THE AVERAGE NUMBER OF TEXT LINES (L) AND GLOSS LINES (G) PER PAGE, LANGUAGE (LANG.), AND SCRIPT.

Dataset	Cent.	l	g	Lang.	Script
CSG18	10 th	27	96	Latin	Carolingian minuscule
CSG863	11 th	30	73	Latin	Carolingian minuscule (oblique oval variant)
CB55	14 th	30	51	Italian, Latin glosses	Cancellaresca (Chancery script)

of the manuscripts, which have been digitized with a resolution of 600 dpi and have a size of approximately 20×25 cm.

The DIVA-HistDB consists of a total of 150 pages of three manuscripts (50 pages each) along with their associated GT in the PAGE [22] format. The GT focuses on the segmentation of manuscripts into text lines, gloss lines, and decorations. For the time being, we do not consider grouping (e.g. text blocks) or semantic labelling (e.g. reading order).

We compiled the dataset CSG18 from the largest unit of Cod.Sang.18², a collection of liturgical works. It is a so-called Psalterium Glossatum with glosses and embellished initials. CSG863³ describes the civil war between Gnaeus Pompeius Magnus (Pompey) and Gaius Julius Caesar. Its marginal and interlinear glosses illustrate the meaning of the main text. Occasionally it bears initials in red ink. CB55⁴ is a copy of Dante’s *Inferno* and *Purgatorio*, written shortly after his death. The first letter in the first line of each three-line stanza is highlighted in red ink.

III. DOCUMENT GRAPHS

The very core of the proposed GT system is a sparse document graph which captures a page’s structure and allows for semi-automatic processing. This section describes our notion of document graphs, from which we extract layout elements by manual interaction, and describe these elements with polygons derived from the graph.

A *graph* G is a four-tuple $G = (V, E, \mu, \nu)$ where V is the finite set of nodes, $E \subseteq V \times V$ is the set of edges, $\mu : V \rightarrow L_V$ is the node labeling function, and $\nu : E \rightarrow L_E$ is the edge labeling function. In this paper, we consider undirected graphs with the label domains $L_V = \mathbb{R}^2$ (image coordinates) and $L_E = \mathbb{R}$ (edge weights). Figure 3 gives a visual summary of the graph extraction process discussed below.

The type and density of graph nodes V can be chosen differently with respect to the document analysis task at hand: coarse graphs could be used to capture high-level structures, for example to derive the reading order, while dense graphs could be used to capture details, for example to recognize characters. For our GT system, we need a graph that accurately represents the foreground. Since our datasets allow for binarization, we use Difference-of-Gaussian (DOG) binarization [8] to obtain a reasonable estimation of the foreground, building upon

²St. Gallen, Stiftsbibliothek, Cod.Sang.18, DOI:10.5076/e-codices-csg-0018

³St. Gallen, Stiftsbibliothek, Cod.Sang.863, DOI:10.5076/e-codices-csg-0863

⁴Köln, Fondation Bodmer, Cod.Bodm.55, DOI:10.5076/e-codices-cb-0055

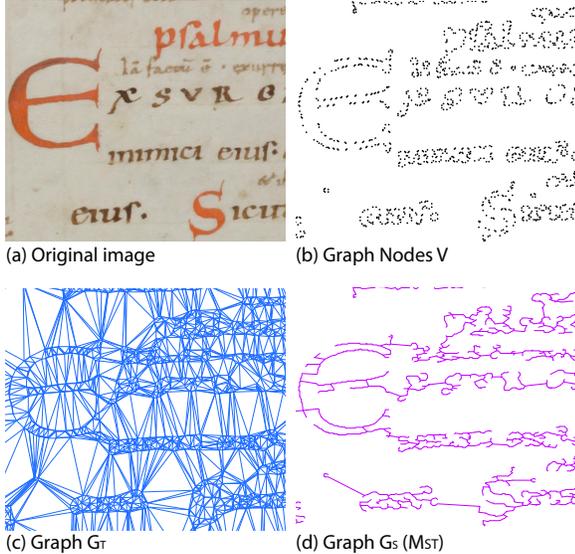


Figure 3. Graph creation from the original image (a). Nodes are depicted as white disks (b), the document graph G_T in blue (c), and the MST G_S in magenta (d).

experience gained from creating the IAM-HistDB [8]. We then define the nodes of the graph $v \in V$ as contour points extracted from the binarized image, which are labeled with their image coordinates:

$$\mu(v) = (x, y) \in \mathbb{R}^2 \quad (1)$$

The Douglas-Peucker algorithm [23] developed for simplifying curves is employed to reduce the number of nodes while preserving the character shapes in order to obtain a sparse representation. The resulting nodes are illustrated in Figure 3 (b). Note that if binarization is not possible, other node sources are conceivable, such as Harris corners, or DOG interest points [21].

We then define the graph edges E by Delaunay triangulation of the nodes, which offers a natural neighborhood definition for point clouds [24], [25]. The resulting triangulated document graph G_T models relations within layout elements as well as relations between layout elements (c.f. Figure 3 (c)). In handwritten documents, it is not guaranteed that within-element distances are smaller than between-element distances. Thus, we assign to each edge $e = (u, v) \in E$ a weighted distance as edge weight:

$$\nu(e) = \|(\mu(u) - \mu(v)) \times W\| \quad (2)$$

where $\mu(u) - \mu(v)$ is the vector difference of the node labels and W is a weight matrix that represents the prevailing orientation of the layout elements (in particular text lines); e.g. $W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ corresponds with the Euclidean distance, while a weight of $W = \begin{pmatrix} 0.5 & 0 \\ 0 & 1.5 \end{pmatrix}$ favors horizontal edges owing to the larger weight assigned to y-coordinates. In this paper, the latter is the default setting for the weight matrix, which is a user-defined parameter of the proposed document graphs.

In order to obtain a sparse graph representation that allows us to identify layout elements as distinct subgraphs, we

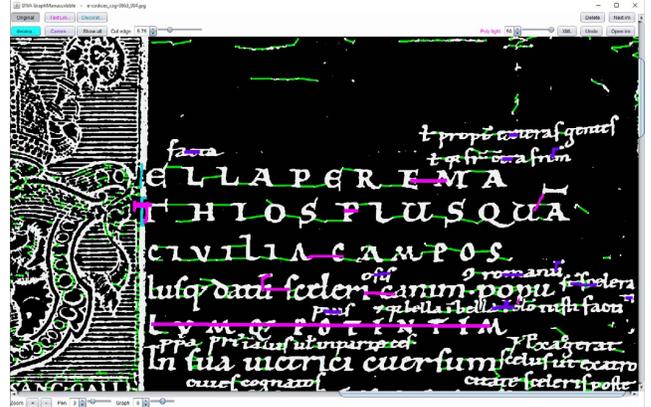


Figure 4. User interface of the DIVA GraphManuscribble system. Binary image with the graph plotted in green and several user annotations in different colors for each class (main, gloss, decoration). Graph edges within the same foreground component are not displayed to the user for convenience.

subsequently cluster the graph with an MST using Kruskal’s algorithm [26]. The resulting MST $G_S = (V, E, \mu, \nu) \subseteq G_T$ is an acyclic subgraph (no loops) of the triangulated graph G_T , which connects all nodes V with exactly $|E| = |V| - 1$ edges such that the sum of the edge weights is minimal. This clustering already reveals the main structures of a document, i.e., it captures mostly within-element edges and only few between-element edges (c.f. Figure 2).

Finally, in order to segment the MST into distinct layout elements, missing within-element edges have to be inserted and between-element edges have to be deleted. To avoid fully manual segmentation during GT creation, a global threshold T is applied to delete all edges with a weight $\nu(e) > T$ for suggesting an initial hypothesis of the document’s structure. This threshold T is the second user-defined parameter of the proposed document graphs. It can be fine-tuned by the user for each document page individually.

IV. SCRIBBLING INTERACTION FOR GT CREATION

The initial graph segmentation is then refined by users with a scribbling interaction pattern. They simply draw strokes with a pen-based input device (Wacom Intuos Tablet CTH-480/S) in order to interact with the graph, that is: (1) inserting edges between components of the same layout element (within-element edges), (2) deleting edges connecting two different layout elements (between-element edges).

No precise interaction is required to insert edges. It is sufficient that an imprecise scribbling stroke touches parts of the graph components which should be connected. At the same time, labels are assigned to the layout elements (main, gloss, decoration), which is facilitated by using different “pencils” (colors) for each label. To delete edges, both imprecise scribbling strokes as well as precise clicks on individual edges can be used. Unlabeled background elements, which have not been assigned to a layout class, are finally discarded. Figure 4 illustrates the user interface of the system, with a binarized image and the MST plotted in the background. Several layout elements have been annotated in their different colors.

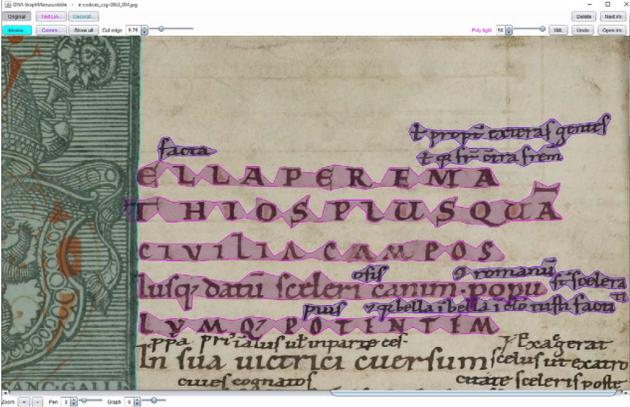


Figure 5. Second perspective of the user interface. Original image with the GT polygons plotted in different colors for each class (main, gloss, decoration).

After segmenting the MST into distinctive layout elements by means of scribbling, a polygonal representation is extracted as the final GT. Intuitively, triangulated graphs permit deriving polygons to represent layout elements in an elegant and direct manner: by iteratively removing exterior edges until the desired shape is obtained (c.f. Figure 6). Accordingly, we triangulate each MST subgraph representing a layout element and iteratively remove the longest exterior edges with a weight smaller than P as long as the resulting polygon fulfills following constraints: it contains neither holes nor islands, and each exterior node has exactly two exterior edges (see Figure 7) [27]. The user-defined parameter P is used to fine-tune the desired tightness of the resulting polygons (see Figure 6 (c)). In order to simplify the polygons while maintaining their original topology, we use an algorithm grounded on the curve-simplification Douglas-Peucker algorithm [23] with additional constraints [28] for the final polygons (c.f. Figure 6 (d)).

The resulting GT is illustrated in Figure 5 showing the original image in a second perspective of the user interface. The user can continue to scribble directly on the original image in order to correct layout annotations. In particular, it is possible to scribble over faded-out parts of the handwriting that might be missing in the binary image, in order to include them in the GT polygons.

V. EXPERIMENTAL EVALUATION AND DISCUSSION

The experimental evaluation of our semi-automatic GT system covers three aspects. First, the quality of the document graphs is evaluated. Secondly, the manual user effort of the scribbling interaction is estimated and, thirdly, we compare the proposed system with previous work [8]. Evaluations are done separately for each dataset of the DIVA-HistDB, where we selected 10 pages each to conduct the evaluation. Measures are expressed as mean ($\mu \pm \sigma$) and median \tilde{x} over the pages of the evaluation dataset.

A. Graph Evaluation

In order to assess the quality of the sparse document graph, i.e., how well the graph corresponds a page's structure, we compute the graph edit distance [29] between the document

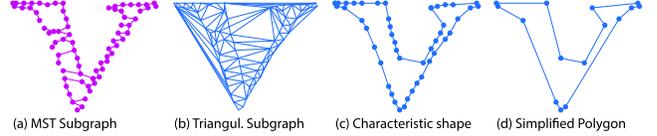


Figure 6. Extracting simplified polygons of layout elements. Nodes are indicated by disks.

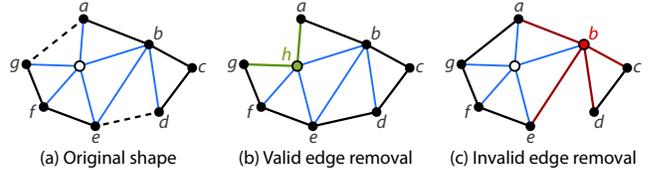


Figure 7. Polygons from a triangulated graph (a): removing the edge $a-g$ is valid (b), since all exposed exterior nodes have exactly two exterior edges. Conversely, removing edge $d-e$ leads to an invalid polygon (c) since node b has now 4 exterior edges.

graph obtained from the page and the graph that optimally corresponds to the GT. It is expressed in terms of the minimum amount of edges that need to be inserted and deleted in order to transform the document graph into the GT graph. The evaluation measures are computed on the final MST graph G_S after removing edges with a weight larger than the user-defined threshold T (see Section III).

The number of insertions indicates how fragmented layout elements are, i.e., how many missing within-element edges have to be inserted. We consider a ratio

$$I = \frac{\#\text{insertions}}{\#\text{correct edges}} \quad (3)$$

with respect to the number of correct edges in the document graph that already correspond with the GT graph. In addition, we consider the total number of insertions I_L with respect to a polygon of an individual layout element.

On the other hand, the number of deletions expresses how connected different layout elements are, i.e., how many between-element edges have to be deleted. Again, we consider a ratio

$$D = \frac{\#\text{deletions}}{\#\text{correct edges}} \quad (4)$$

with respect to the correct edges in the document graph. The total number of deletions for a polygon of an individual layout element is denoted D_L .

Table II shows the edit distance for each dataset of the DIVA-HistDB at the user-defined threshold T . We observe that less than a total of 1% of the edges are missing or surplus, respectively, and that there are significantly fewer insertions than deletions.

The ratio between insertions and deletions is dependent on the threshold T chosen for each page individually by the user. An exemplary analysis of the influence of this threshold is shown in Figure 8 based on CSG18. The graph shows the measures I and D – i.e., the ratio of the edges that have to be inserted and deleted, respectively, at a given threshold T . *A posteriori*, that is after creating the GT, we can state that a value around 4.75 would have been a good choice for T ,

Table II
CORRELATION OF THE DOCUMENT GRAPHS AND THE GT AT THE USER-SELECTED THRESHOLD ($\mu(T) = 5$). THE UNIT U IS EITHER THE PERCENTAGE (%) OF EDGES WITH RESPECT TO THE CORRECT EDGES, OR THE NUMBER (#) FOR EACH LAYOUT ELEMENT.

measure	CSG18	CSG863	CB55	total	u
	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	$\mu \pm \sigma$	
with respect to correct edges					
I	0.18 ± 0.1	0.17 ± 0.16	0.23 ± 0.05	0.19 ± 0.11	%
D	0.56 ± 0.22	0.46 ± 0.33	0.58 ± 0.07	0.53 ± 0.23	%
with respect to layout elements					
I_L	1.01 ± 0.66	1.42 ± 0.97	3.17 ± 0.73	1.87 ± 1.22	#
D_L	3.49 ± 1.30	4.05 ± 1.59	7.98 ± 1.28	5.17 ± 2.44	#

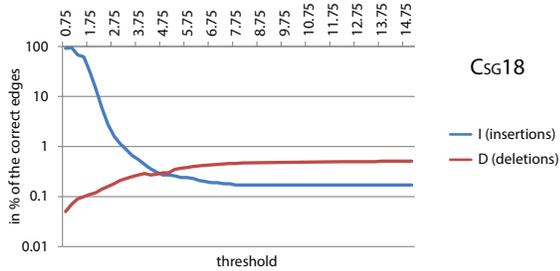


Figure 8. Influence of the edge weight threshold T on I and D .

balancing the number of insertions and deletions.

The layout-element-based measure gives a lower bound to the expected user effort: for each layout element, there are about 1 ± 0.7 edges missing on average that have to be inserted, and 3.5 ± 1.3 edges connecting different layout elements that have to be deleted. The high number of surplus edges can be explained by the proximity of the text lines, especially for interlinear glosses.

As regards the computation time, on the three datasets, the triangulation takes 0.68 sec (CSG18), 0.78 sec (CSG863) and 1.61 sec (CB55) per page on average on an Intel Core i5-3360M, 16GB without multi-threading. The MST computation takes 7.42, 15.84, and 24.78 sec, respectively.

B. User Interaction Evaluation

In order to estimate the user effort required for ground truthing, we have evaluated the interaction time and the number of interactions for a user who did not have thorough knowledge about the method but has already annotated several document pages beforehand. The user was asked to create the GT for 10 pages of each dataset of the DIVA-HistDB.

The user effort needed in order to create the GT is evaluated per page and per layout element for each dataset. It is expressed by the interaction time and the average number of insertion and deletion interactions required. We focus on the evaluation of user interactions per layout element; the full evaluation is given in Table III. On average, a user spends about 14.92 ± 3.93 seconds in order to process a single layout element with an average of 5.06 ± 1.62 interactions. The ratio of insertions and deletions is approximately 2:3.

The number of required user interactions per layout element does match the lower bound of the expected user effort

Table IV
ANALYSIS OF THE USER INTERACTION PATTERNS EXPRESSED AS MEAN ($\mu \pm \sigma$) AND % OVER ALL DATASETS

interaction	insert		delete		total	
	$\mu \pm \sigma$	%	$\mu \pm \sigma$	%	$\mu \pm \sigma$	%
total	2.01 ± 0.44	40	3.06 ± 1.39	60	5.06 ± 1.62	100
scribbles	1.99 ± 0.44	99	2.44 ± 1.43	80	4.42 ± 1.60	87
clicks	0.02 ± 0.02	1	0.62 ± 0.50	20	0.64 ± 0.50	13

computed in the previous section as edit distance: we computed a total average of 1.87 ± 1.22 insertions and 5.17 ± 2.44 deletions on the entire DIVA-HistDB (see Table II).

Note, however, that one scribble interaction can correspond to several actions: e.g. several fragments can be connected, or equally, multiple edges can be cut with just one scribble.

Table IV analyses the interaction patterns; we distinguish two types of interactions: scribbles (pen strokes) and clicks. The tables clearly show that scribbling is preferred (87% of all interactions) over clicking. However, a distinction has to be made for the separate tasks of insertion and deletion. Insertion was almost exclusively performed with scribble interactions (about 99% of those interactions are scribbles). On the contrary, clicking is used in 20% of the cases for deleting edges.

C. Comparison with the IAM-HistDB GT tool

We compare our DIVA GraphManuscribble GT system with the semi-automatic process previously proposed for the IAM-HistDB [8]. In order to allow a direct comparison, the evaluation is based on 3 pages of the Parzival⁵ dataset from the IAM-HistDB [8]. Here, an expert user with thorough knowledge about the method has conducted the study in order to allow for a fair comparison.

Our expert user needed 6:30 min on average in order to segment and label one page of Parzival. The users of the previous HisDoc GT tool needed a total of 9 min of manual interactions, which included selecting the text regions and correcting the nodes of the automatically suggested polygons.

VI. CONCLUSIONS

In this paper, we have proposed a novel approach to ground truth creation based on document graphs. The two major benefits of using graphs are that nodes as simple as contour points accurately model the layout of a page, and that the Minimum Spanning Tree (MST) leads to a data reduction allowing users to deal efficiently with large graphs.

We have demonstrated that the sparse document graph structure is already very close to the desired ground truth annotation despite the chosen documents' complexity and only few user interactions are needed to insert and delete edges. This makes the proposed system an efficient tool to create ground truth even for challenging documents. Furthermore, pen-based scribbling as a user interaction has also proven to be swift and natural for human annotation.

⁵St. Gallen, Stiftsbibliothek, Cod.Sang.857, DOI:10.5076/e-codices-csg-0857

Table III
 USER INTERACTIONS EXPRESSED IN INTERACTION TIME AND NUMBER OF INTERACTIONS PER PAGE AND LAYOUT ELEMENT

measure	CSG18				CSG863				CB55				DIVA-HistDB				unit
	μ	\pm	σ	\tilde{x}	μ	\pm	σ	\tilde{x}	μ	\pm	σ	\tilde{x}	μ	\pm	σ	\tilde{x}	
per page																	
interaction time (m:s)	33:40	\pm	16:49	40:55	26:52	\pm	22:17	25:22	19:24	\pm	09:36	18:09	26:39	\pm	17:29	22:56	m%/page
# interactions	585.00	\pm	324.60	680.00	533.00	\pm	473.56	676.00	552.80	\pm	237.19	571.00	556.93	\pm	346.74	618.00	%/page
# layout elements	125.90	\pm	64.90	134.00	103.70	\pm	76.98	135.00	82.40	\pm	20.54	89.00	104.00	\pm	60.03	91.00	%/page
per layout element																	
interaction time (s)	15.76	\pm	4.93	15.55	23.14	\pm	25.67	19.84	19.45	\pm	28.94	8.13	14.92	\pm	3.93	14.49	%/poly
# interactions	4.30	\pm	1.26	4.58	4.40	\pm	1.24	4.77	6.49	\pm	1.40	6.79	5.06	\pm	1.62	5.01	%/poly

Future work includes the development of methods for suggesting potential edge insertions and deletions to the user in order to improve the efficiency of the ground truth creation. A particular aim is to learn from the user interaction to improve such suggestions over time. Furthermore, a promising line of research would be to investigate the proposed document graphs in the context of other document analysis tasks ranging from line detection over writer identification to image retrieval. A more extensive user study that includes a bigger set of participants as well as a wider selection of manuscripts is under development.

ACKNOWLEDGMENTS

This work has been supported by the Swiss National Science Foundation project 205120_150173.

REFERENCES

- [1] F. Le Bourgeois and H. Emptoz, "Debora: Digital Access to Books of the Renaissance," *Int. J. of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 193–221, 2007.
- [2] H. Balk and A. Conteh, "IMPACT: Centre of Competence in Text Digitisation," in *Wkshp. on Historical Document Imaging and Processing*, 2011, pp. 155–160.
- [3] S. Nicolas, T. Paquet, and L. Heutte, "Enriching Historical Manuscripts: The Bovary Project," in *Document Analysis Systems VI*, ser. Lecture Notes in Computer Science. Springer, 2004, vol. 3163, pp. 135–146.
- [4] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments," in *Int. Conf. on Document Analysis and Recognition*, 2011, pp. 48–52.
- [5] H. Z. Nafchi, S. M. Ayatollahi, R. F. Moghaddam, and M. Cheriet, "An Efficient Ground Truthing Tool for Binarization of Historical Manuscripts," in *Int. Conf. on Document Analysis and Recognition*, 2013, pp. 807–811.
- [6] O. Biller, A. Asi, K. Kedem, and I. hak Dinstejn, "WebGT: An Interactive Web-Based System for Historical Document Ground Truth Generation," in *Int. Conf. on Document Analysis and Recognition*, 2013, pp. 305–308.
- [7] B. Couasnon, J. Camillerapp, and I. Leplumey, "Access by Content to Handwritten Archive Documents: Generic Document Recognition Method and Platform for Annotations," *Int. J. of Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 223–242, 2007.
- [8] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz, "Ground Truth Creation for Handwriting Recognition in Historical Documents," in *Int. Wkshp. on Document Analysis Systems*, 2010, pp. 3–10.
- [9] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty Years of Graph Matching in Pattern Recognition," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.
- [10] A. Kandel, H. Bunke, and M. Last, Eds., *Applied Graph Theory in Computer Vision and Pattern Recognition*, ser. Studies in Computational Intelligence. Springer, 2007, vol. 52.
- [11] K. Borgwardt, C. Ong, S. Schönauer, S. Vishwanathan, A. Smola, and H. Kriegel, "Protein Function Prediction via Graph Kernels," *Bioinformatics*, vol. 21, no. 1, pp. 47–56, 2005.
- [12] P. Dickinson, H. Bunke, A. Dadej, and M. Kraetzl, "Matching Graphs with Unique Node Labels," *Pattern Analysis and Applications*, vol. 7, no. 3, pp. 243–254, 2004.
- [13] Z. Harchaoui and F. Bach, "Image Classification with Segmentation Graph Kernels," in *Int. Conf. on Comp. Vision and Pattern Recognition*, 2007, pp. 1–8.
- [14] S. Lu, Y. Ren, and C. Suen, "Hierarchical Attributed Graph Representation and Recognition of Handwritten Chinese Characters," *Pattern Recognition*, vol. 24, no. 7, pp. 617–632, 1991.
- [15] J. Lladós, E. Martí, and J. Villanueva, "Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1137–1143, 2001.
- [16] A. Fischer, K. Riesen, and H. Bunke, "Graph Similarity Features for HMM-Based Handwriting Recognition in Historical Documents," in *Int. Conf. on Frontiers in Handwriting Recognition*, 2010, pp. 253–258.
- [17] P. Wang, V. Eglin, C. Garcia, C. Llargeron, J. Lladós, and A. Fornés, "A Coarse-to-Fine Word Spotting Approach for Historical Handwritten Documents Based on Graph Embedding and Graph Edit Distance," in *Int. Conf. on Pattern Recognition*, 2014, pp. 3074–3079.
- [18] P. Riba, J. Lladós, A. Fornés, and A. Dutta, "Large-Scale Graph Indexing Using Binary Embeddings of Node Contexts," in *Wkshp. on Graph-Based Repres. in Pattern Recognition*, 2015, pp. 208–217.
- [19] T. Le, M. Luqman, J. Burie, and J. Ogier, "A Comic Retrieval System Based on Multilayer Graph Representation and Graph Mining," in *Int. Wkshp. on Graph-based Repres. in Pattern Recognition*, 2015, pp. 355–364.
- [20] F. Yin and C. L. Liu, "Handwritten Chinese Text Line Segmentation by Clustering with Distance Metric Learning," *Pattern Recognition*, vol. 42, no. 12, pp. 3146–3157, 2009.
- [21] A. Garz, A. Fischer, R. Sablatnig, and H. Bunke, "Binarization-Free Text Line Segmentation for Historical Documents Based on Interest Point Clustering," in *Int. Wkshp. on Document Analysis Systems*, 2012, pp. 95–99.
- [22] S. Pletschacher and A. Antonacopoulos, "The PAGE (Page Analysis and Ground-truth Elements) Format Framework," in *Int. Conf. on Pattern Recognition*, 2010, pp. 257–260.
- [23] D. Douglas and T. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature," *Cartographica: Int. J. for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [24] R. Sibson, "A Brief Description of Natural Neighbor Interpolation," in *Interpreting Multivariate Data*, V. Barnett, Ed., 1981, ch. 2, pp. 21–36.
- [25] W. Frey, "Selective Refinement: A new Strategy for Automatic Node Placement in Graded Triangular Meshes," *Int. J. for Numerical Methods in Engineering*, vol. 24, no. 11, pp. 2183–2200, 1987.
- [26] J. Kruskal, "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," in *American Math. Society*, vol. 7, no. 1, 1956, pp. 48–48.
- [27] M. Duckham, L. Kulik, M. Worboys, and A. Galton, "Efficient Generation of Simple Polygons for Characterizing the Shape of a Set of Points in the Plane," *Pattern Recognition*, vol. 41, no. 10, pp. 3224–3236, 2008.
- [28] Vivid Solutions, "JTS Java Topology Suite." 2004. [Online]. Available: <http://www.vividsolutions.com/jts/jtshome.htm> (2004-03-11)
- [29] A. Sanfeliu and K. S. Fu, "A distance measure between attributed relational graphs for pattern recognition," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, no. 3, pp. 353–363, 1983.