

Towards the interactive transcription of handwritings

Anytime anywhere document analysis

Björn Gottfried · Marius Wegner · Mathias Lawo

Received: date / Accepted: date

Abstract This paper introduces the *anytime anywhere document analysis* methodology applied in the context of computer aided transcription. Its utility is revealed for documents which are difficult to analyse, as in the case of handwritten texts. A special focus lies on the glyph separation problem which turns out to be particularly complicated. As automatic methods show fundamental limitations, a number of interactive methods are proposed which are based on the interplay between user and machine. These methods get along without any assumptions concerning underlying languages or appearances of texts. An evaluation in the context of palaeography and applied to a well-established data set illustrates how well handwritings are dealt with, although they offer distinct differences in their regularity and shape.

Keywords Digital Palaeography · Computer aided transcription · Anytime anywhere document analysis · Glyph separation · Latin medieval handwritings

This work was funded by the German Research Foundation (DFG) under grant numbers {GO 2023/4-1, LA 3066/4-2} and LA 3007/1-1 under the project name Diptychon and partially under grant number GO 2023/3-2.

B. Gottfried
Centre for Computing and Communication Technologies
University of Bremen
E-mail: bg@tzi.de

M. Wegner
E-mail: mwegner@tzi.de

M. Lawo
Monumenta Germaniae Historica
Berlin-Brandenburgische Akademie der Wissenschaften
E-mail: lawo@bbaw.de

1 Introduction

The automatic analysis of handwritings remains a difficult research problem, although much work has been done in this field [22]. This results both from the enormous variations in the handwriting style of different writers, and from the changing style over the centuries due to the evolution of the various Latin scripts [17]. Even the appearance of the text of a single writer varies to a large extent, since there are no restrictions regarding his way of writing apart from the attempt to provide a legible manuscript; this, however, causes inconsistent spacing between lines and among words [14], not to mention the skewness of lines, the slant of characters, and their differences in size as well as shape.

In consequence, any algorithmic approach that tries to recognise handwritten texts will be faced with an entirely novel piece of handwriting style, showing an individual visual appearance. This happens at least, whenever the algorithm is fed with a manuscript of a writer whose handwriting has not been processed before. While one approach consists in employing learning algorithms that adapt the performance of the recognition system to specific writing styles, there might be a lack of data when only short documents are available or a learning phase might be inappropriate given the workflow in a specific application. In this case, a linguistic processing level can compensate for difficulties arising at the visual level [14]. But such a linguistic model might not be available in the process of transcription. In fact, a model of that kind might itself be one of the goals to be achieved, based on a set of transcribed documents, and this even more so, as the individual orthography of a scribe might be unusual and itself the subject of interest, while a general linguistic model might tend to normalise unusual orthography.

Indeed, from the palaeographer's point of view it cannot be the exclusive purpose to provide the transcription of a manuscript. It is rather of interest to analyse, among others, the appearance of a writing style, the within-writer variability with respect to single alphabetic characters, to compare similar characters in their appearance, or to look at the differences the writer made with regard to the written letters of a single character class, depending on their position in a word as well as in a document, the orderliness frequently degrading the longer the text. Examining characteristics like that makes the thorough analysis of writing hands possible.

In this paper we introduce a methodology for the interactive transcription of handwritings. Alongside the transcription output, the shape of each single character is precisely extracted. This allows a detailed description of a manuscript at the level of alphabetic characters. Because of the difficulties of non-restrictive handwritings, the method introduced here relies on the interaction between efficient algorithms on the one hand and the flexible abilities of the user on the other hand. This enables what we refer to below as the anytime anywhere document analysis paradigm.

In the following section, three typical application scenarios are introduced that motivate the need for a new interaction paradigm. A number of systems supporting the transcription of handwritings as well as specific interaction paradigms are shown and terms employed in this context are discussed in section 3. The notion of anytime anywhere document analysis methods is introduced afterwards. One of the most intricate problems in the context of handwriting analysis is dealt with in section 5. It presents three different glyph separation methods, which are evaluated in the following section with respect to different writing hands. A general discussion and summary will close this paper.

2 Typical application scenarios

2.1 Mass data processing by amateurs

For the expert the amount of data sometimes makes it necessary to delegate work to the staff. However, transcribing a Latin manuscript with its incomplete or not existing word separation and dozens, if not hundreds of ambiguous abbreviations there are many potential mistakes an amateur can commit and which need to be corrected later at low effort by a qualified palaeographer who is familiar with such handwritings. E. g. there might be a wrong letter within the final transcription. A fault, however, which can be easily revised.

The underlying causes for a given fault might have deeper roots: A defective letter could have been ex-

tracted much too imprecise. As a consequence, the binarisation needs to be adjusted. Changing, however, the figure-ground separation, results of subsequent processing steps like the separation of glyphs in the neighbourhood or their transcription must not be affected.

The solution would be the local application of some image pre-processing methods that do not affect any other part of the image or any glyph already extracted. In the current scenario, image enhancement should be possible for a local area at any time, even if all other subsequent image processing steps have already been applied, including the extraction and transcription of all other letters.

2.2 The preparation of critical editions

In most cases, the treatment of old medieval handwritings is neither simple nor clear. The opinions of experts may differ as to how to extract all signs correctly, for example whether a region is part of an abbreviation sign or a punctuation mark or simply an inkblot.

While a transcription has been provided by some assistant transcribers, two or more editors should take a copy of that very same transcription in order to adjust it according to their own ideas and research assumptions. This might imply to change the extraction of signs locally depending on how they are interpreted. As a matter of fact, sometimes a meaningful interpretation becomes clear just as soon as a partial transcription of the document is available, making it necessary to mix image enhancement, region extraction, and transcription routines.

2.3 The joint work of editors

Another scenario is about two editors who are working together on the same documents in order to share their experiences and skills. Mistakes made by one of them are identified by the other one and vice versa. Several kinds of oversights are conceivable, for example that letters have not been separated correctly.

Such segmentation steps need to be adjusted independently on all other letters which have already been correctly segmented within the same image. Additionally, a revision of single letters should be possible no matter how far the whole document is already transcribed by the other editor or by oneself. Both editors need to work up and down across a single document page which needs to be exchanged among them with its current revision status.

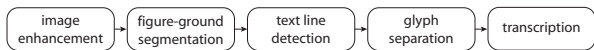


Fig. 1 A conventional document processing workflow.

2.4 The principles of temporality and locality

In all those scenarios it is neither sufficient just to revise the output at the end of the whole processing chain (like in [31] or as shown in Fig. 1), nor is it sufficient to provide user interactions for the different processing steps individually (as provided by [3,5,7,27]), i. e. before the next step is processed. The conventional document image processing chain is not longer applied to a document image as a whole, but, if necessary, to each locality of interest separately. It must be possible to revise just parts of the processing chain whenever and wherever necessary. In a nutshell, a method is required that meets two conditions:

- The principle of temporality: Revisions must be possible for any processing step regardless of how far the document is already transcribed.
- The principle of locality: Revisions need to be applicable locally without influencing any processing results of other locations within the same document image.

Above all, both conditions must apply at any time. That means, any revision at any location can on principle be redone ever and ever again.

3 Related work

A number of transcription systems do exist which are outlined in the following. These systems can be roughly classified with regard to the textual level they mainly take into account. Accordingly, line-based and word-based methods can be distinguished as well as those which directly work on alphabetic characters. Examples of all three kinds of approaches are described in turn, followed by a number of systems that are reviewed from the standpoint of user interactions to compare them with the desired interaction methodology introduced in the previous section. Eventually, the employed terminology is clarified.

3.1 Transcription systems

The computer assisted transcription system described by [24] proposes transcriptions of text lines. The user

corrects these transcriptions and the system adapts itself to those corrections, in order to make improved suggestions. This process is repeated for a single text line until the user has validated the entire line. While the extracted features of a text line image are processed by a Hidden Markov Model, at the linguistic level this system employs a word model and a character model. The approach by [16] is also text line based. It covers the whole workflow from accessing documents to the recording of annotated transcriptions.

In [26] a computer assisted transcription approach is described which works at the level of words. The system tries to recognise whole words by means of word graphs that show the probabilities of similar words to come into consideration, while words are prompted to the user where the system lacks confidence about their transcription. The feedback of the user improves the performance.

Both [24] and [26] employ a language model, as does [30], where a recognition system for modern scripts has been adapted to medieval scripts. In particular, the authors investigate the creation of suitable language models, which is often difficult due to the small quantity of verified ground truth transcriptions available.

As opposed to the mentioned systems, slant corrections and size normalisations are not desirable for our transcription system, since the original appearance of the handwriting is a valuable source for the palaeographic study. Additionally, the way how words separate into single characters is of no relevance within the systems mentioned thus far. They are restricted to export transcriptions, but lack any means for providing palaeographic methods at the level of single symbols and their visual appearance. Such palaeographic features are also needed elsewhere [12]. But up to now, the extraction of single character symbols during the transcription process is confined to rectangular areas that enclose single characters.

The clustering of similar character symbols based on their visual appearance is another approach [10,23]. In this case, a transcription amounts to the assignment of a character to each cluster. This method works most efficiently if all the character symbols of one character class can be grouped into the same cluster, and reduces the transcription effort inversely proportional to the number of character symbols contained in the clusters. Conversely, a poor grouping result, which is due to a large within-writer variability, results into many clusters. These have to be managed separately during the transcription process.

3.2 Interaction methods

In [4] a *directed interaction model* is proposed which prompts questions to the user whenever it detects any problems in the context of layout analysis. The initiative is, so to speak, taken by the system. Such an approach is possible when employing a model that defines potential problems the user would have to resolve for the system.

Along the same lines, there are other interaction models based upon page descriptions. They pertain to the paradigm referred to as *concept-driven grammatical document analysis methods* [3] and they are confined to specific document layouts or even to specific contents. Another one is the *spontaneous interaction model*. Here, it is not the system but the user who has to take the initiative and who is allowed to change parts of the image content in order to let the system know how to improve automatic processing [3]. Furthermore, it is possible to annotate the employed document model itself in order to show the system potential pitfalls for which it gets prepared to receive corrections by the user during interaction steps.

An example for systems allowing user interactions at the final stage is the post-processing tool provided by [31]. The document structure as well as erroneous character recognition results can be corrected for digitised documents to be published. In contrast to this method, interactions are possible at different processing steps within the framework of [5]. This approach uses a scheduling module that makes use of a predefined scenario for which documents are to be analysed. It allows in particular the integration of user interaction modules within that framework.

Another example for user interactions at different processing steps is the system provided by [7] for the purpose of ground truth data generation in the context of handwriting recognition. The user has to select the text areas by means of the GNU image manipulation program called *gimp* [20], which is also used for binarisation where the user is asked to find appropriate parameters. Text lines are found by their system afterwards and can be corrected manually, as can the text alignment. Word boundaries for each text line are detected by means of a Hidden Markov Model. Those boundaries can be adjusted by the user if necessary.

Similar to [7], the approach of [27] builds upon the *gimp* toolbox. They also allow interactive corrections regarding the detection of text blocks, text lines, and the transcription. However, this approach is designed to work with collections of homogeneous documents, sharing a similar structure and writing style. The user can decide in which order to transcribe text lines. But

the document processing order of block and text line detection as well as transcription has to be followed.

For the sake of completeness, investigations should also be mentioned that do not primarily focus on user interactions in the context of transcriptions, but which generally relate to the managing and archiving of documents. Thus, [15] describes a collaborative managing and remote access platform for digital reproductions of books of the Renaissance. As the fast access of digitised documents via the Internet is one of the main issues there, compression methods have been developed within that project. In order to make the contents of books available, manual labeling of character patterns is made possible, though this project is confined to printed texts and is not meant to provide complete and well-edited transcriptions, but search engines with a certain minimum amount of contents to support the search for those documents.

None of these approaches can deal with scenario 2.1 that requires a local change by means of an early processing step while the text is already transcribed. Regarding scenario 2.2, there might be ways to save and multiply intermediate results for some of the aforementioned systems. Here again, binarisation results or those relating to the extraction of particular regions concern earlier processing stages which need to be revised. But this should be possible without changing anything but the affected abbreviation signs, punctuation marks, or inkblots, depending on the interpretation for a specific critical edition. Finally, scenario 2.3 does not only require the former flexibility concerning punctual corrections within earlier processing steps. It additionally demands the exchange of a document at any current editing state. Conventional approaches are designed to enable interactions at different processing stages, but usually in a particular order and for the entire document page, sometimes even confined to specific document types. Sooner or later, earlier processing methods become inaccessible. However, for an editor the principles of temporality and locality in document processing are necessary.

3.3 Glyph versus abstract symbol

In order to distinguish the visual appearance of information contained in a document image and its abstract description we refer to the linguistic notions of writing systems. In their context, a grapheme is the smallest semantically distinguishing unit [1]. The concrete written graphical symbol of a grapheme is referred to as a *glyph* or *graph*. Those glyphs which represent the same grapheme are called allographs.

In our case glyphs usually correspond to single letters but might also refer to ligatures, diacritic marks, or punctuation marks. In other words, the document image contains glyphs, while the transcription represents those glyphs by means of abstract symbols. It is the purpose of the presented work to extract all glyphs from a given document image and to assign to each glyph the correct abstract symbol.

The state of the art shows the difficulties of separating glyphs from each other automatically, which is also referred to as *character segmentation* [2]. This is why many methods rely on the recognition of words instead of single characters, together with word models of the underlying language [24, 26, 30]. Even more difficult than the segmentation of characters is to extract the shapes of single glyphs from the image precisely. This however is necessary, because it is not the only purpose to assign to each glyph its abstract symbol. Palaeographic researchers are rather interested in analysing the visual appearance of glyphs and want to compare glyphs in different contexts [29]. Consequently, methods are required which allow the careful separation and extraction of single glyphs. Due to the problems arising with any non-restrictive handwriting such methods will hardly work without the user who will have to resolve difficult cases.

4 Semi-automatic document analysis

In contrast to the conventional document image processing approach which distinguishes a number of processing steps [18], for instance those shown in Fig. 1, the presented approach allows to adjust everything, at any time and wherever necessary on the document page. While the user is inspecting the document, the system is asked to make suggestions about how to extract lines of text, for example. But as it is difficult to make perfect suggestions at the algorithmic level, the user can freely adjust the detected text lines. The design of the presented approach grants to her full control of the entire analysis process, while she benefits from efficient image processing algorithms.

4.1 Palaeographic studies

The palaeographic study of a document image usually means to be engaged in the inspection of that image for a long while. In doing so, one possible aim is to obtain all alphabetic characters and abbreviations employed by the writer. Moreover, palaeographical features about the document are to be determined in order to characterise his peculiar handwriting. While this can

be supported algorithmically, a number of interactions with the document would support the researcher to become familiar with the handwriting. This includes the search for strings in the original document, to show specific glyphs in their document context, and to pick out glyphs in order to compare them directly.

The continuous treatment of the document also implies, little by little, that the user detects glyphs hitherto not precisely extracted from the image, not completely lying in a text line, or other peculiarities which need adjustments. This is where the anytime anywhere document analysis paradigm gets involved, which allows the user to introduce any improvements concerning image enhancement, feature detection, and glyph recognition. Those adjustments, however, must not have an effect on any features already extracted. Any side effects are to be avoided. This is demanded by the anytime anywhere paradigm. Besides, the user can directly choose interaction methods, e. g. in order to mark a text line or glyph by herself, instead of letting the system deal with particular complex cases which would be difficult to adjust afterwards. Nevertheless later processing steps can again be automatic: User interventions and automatic methods interact with one another.

4.2 Adjustments

The methodology consists in the interplay between user and machine. This concerns image enhancement, figure-ground segmentation, text line detection, glyph separation, and transcription. The suggestions made by the system are immediately visualised:

- image enhancement: the image can be enhanced globally; alternatively, by means of a virtual *rubber band* an area can be defined for which the contrast is changed or in which noise is removed;
- figure-ground segmentation: by a toggle-button the user can switch between the original image and black & white image, showing the quality of the extracted glyphs; a rubber band can be dragged around an area for which the figure-ground segmentation is to be adjusted independent of the remaining image;
- text line detection: each text line is framed by a rectangle; the user can grab that rectangle at each side in order to change its size towards all directions;
- glyph separation: adjacent glyphs are dyed with different colours; a glyph can be selected in order to align its shape precisely to the document image;

- transcription: for each line the transcription is shown at the same vertical level on a panel right of the original document page, as shown in Fig. 3; the transcription can be changed for the whole text line or for each single glyph independent of the rest of the text line.

The user can at any time, anywhere change these features locally and irrespective of the remaining parts of the document whenever facing a defect. Fig. 2 illustrates this kind of flexible workflow which has abandoned the conventional document processing chain [31]. In fact, there is no fixed workflow anymore: The user loads a document image and can apply the methods locally just as desired. Alternatively, it is possible to load the current editing state and to continue working with it. Fig. 3 shows the user interface with a document and its transcription.

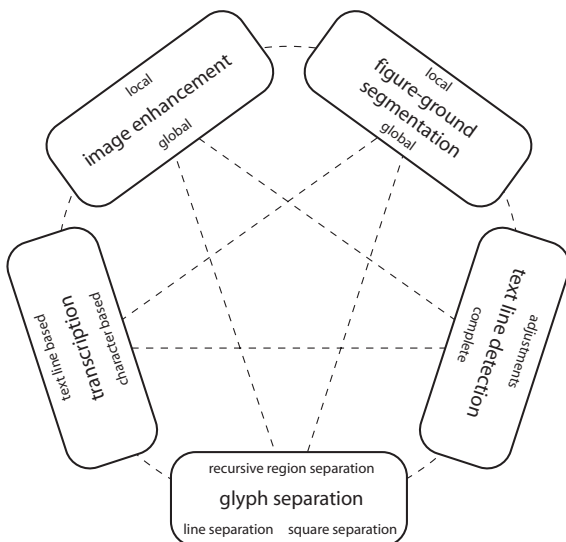


Fig. 2 Instead of the usual workflow shown in Fig. 1, in the Diptychon system presented here any method can follow any other method somewhere on the document page. There is only the *transcription* which either requires a single glyph already being extracted by means of local figure-ground segmentation or by having extracted a first text line. Apart from that, each method, automatically or interactively applied, can follow each other method anywhere on a document image.

In the following the algorithms are shortly outlined. It turns out that the advantage of our methodology consists in simple algorithms which just need to produce more or less accurate results, since the user will compensate for any imperfections. Additionally, the simplicity of the algorithms keeps them general, not confined to specific handwritings.

4.3 Figure-ground segmentation

Connected components which represent single glyphs or parts of disconnected glyphs are determined by applying Sauvola's binarisation filter [25]. The user can determine the size of influence of the filter as well as a threshold Θ_k for the deviation allowed from the average grey value within the area of influence. The latter adapts the threshold to the local contrast in the neighbourhood.

Then, for each pixel $I(x, y)$ in the image I an individual threshold Θ_{bin} is computed in order to decide whether $I(x, y)$ pertains to the figure or the background:

$$(1) \quad \Theta_{bin}(x, y) = \bar{x}(x, y) [1 + \Theta_k (\frac{s(x, y)}{s_{max}} - 1)]$$

The mean \bar{x} and standard deviation s are computed within a range of the image corresponding to the area of influence. s_{max} is the maximal value for the standard deviation, which is 128 in a grey scale image with 256 different possible values.

4.4 Text line detection

Text lines can be detected by means of projection profiles along the horizontal of the image [19]:

$$(2) \quad P(y) = \sum_{x=1}^{x_{max}} I(x, y)$$

Deskewing algorithms that aim at the complete alignment of a document [13], which has been inappropriately digitised, for example, should be applied beforehand in order to obtain a useful projection profile. They often fail, however, when facing a dropping handwriting. In order to manage such slanting text lines, an algorithm has been devised which adapts the idea underlying the approach of [28]. It determines parts of a text line which are given by connected components. The smallest enclosing rectangles are defined around those components. By means of a least median square method the enclosing rectangles are collected to define a coherent text line. Thereby, a number of constraints have to be satisfied, especially in regard to the difference in height of the rectangles. Deviations from these constraints lead to the definition of new text lines.

4.5 Single glyph extraction

The separation of glyphs proves to be particularly difficult, as different handwritings show differences in how

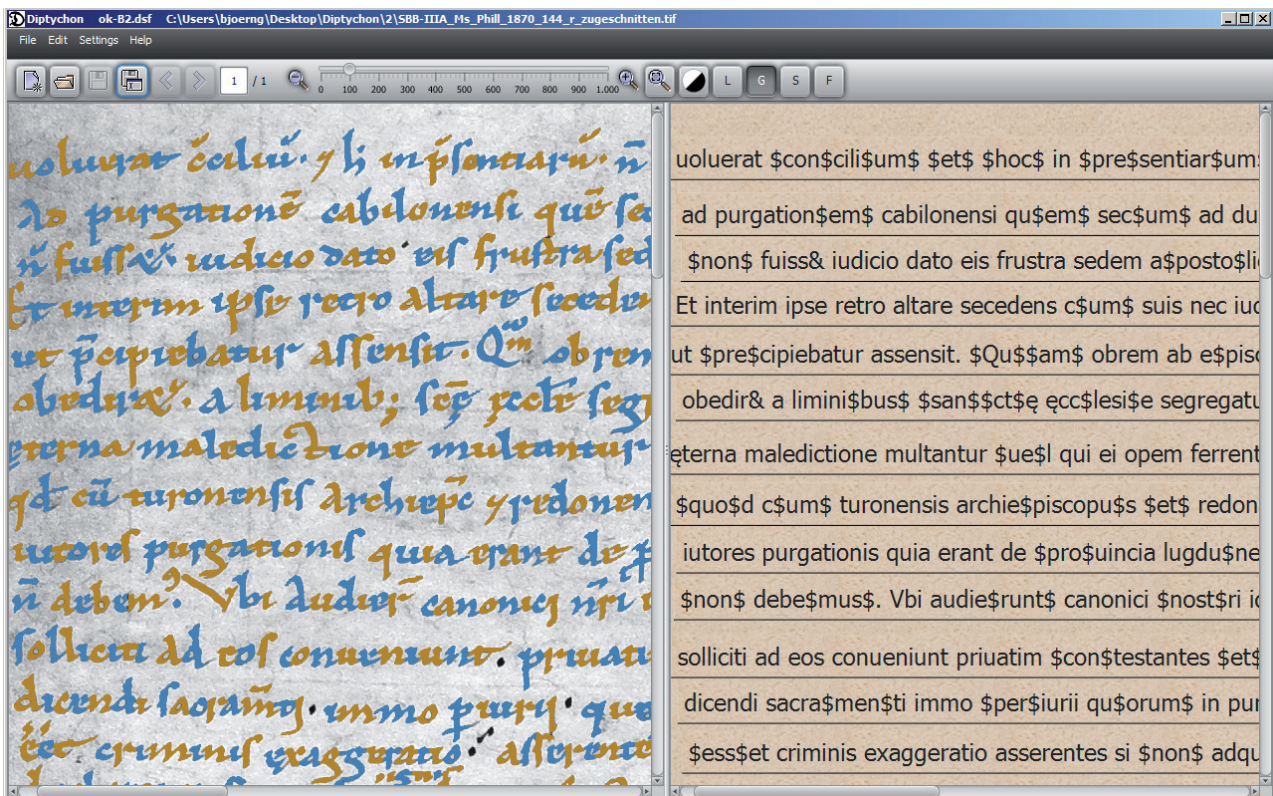


Fig. 3 A document page from [21] displayed within the user interface of the Diptychon system. Left: the separated glyphs are displayed over the original document. The changing colours of the glyphs show how they got separated. Right: the transcription. The \$-signs enclose the abbreviations.

characters connect. As a consequence, many methods rely on the interaction between character segmentation and classification [2].

Here, we follow a common heuristic that looks for thin sections within the handwriting, assuming that these refer to paths along which adjacent characters meet. Anywhere on the document page single glyphs can be locally extracted from the image. For this purpose, the figure-ground segmentation methodology described in section 4.3 is applied. If it is accidentally connected to adjacent regions, the user has the opportunity to crop the glyph from those regions.

5 Interactive glyph separation

The three categories of document analysis addressed in the previous section can be applied to any part of the document at any time. Among these methods, the extraction of single glyphs is the main challenge. It requires an efficient procedure for the separation of glyphs, i. e. easily manageable for the user but nonetheless very effective. The glyph extraction method, described in section 4.5, is much too laborious when applied to a document page with a few thousands of glyphs to be

separated. Therefore, a new interactive glyph separation algorithm is introduced in this section which is dedicated to solve this problem.

5.1 A text line based method

The new method is text line based. It processes simultaneously all the glyphs contained in a single line. Separating the whole bulk of glyphs in a row is more efficient than treating each single glyph apart from its context.

5.1.1 Known versus unknown transcriptions

The user has two alternatives. Either she lets the system know about the transcription of that text line or she simply requests the system to find a separation independent of any transcription. The former mode is applied by a user who is basically familiar with the given handwriting and who is mainly interested in the extraction of features for further palaeographic studies. The latter mode is helpful if the user herself has difficulties in separating glyphs, as it might be the case for complex and unknown handwritings. The first mode has the advantage that the number of characters n is passed to the

separation algorithm. This knowledge helps the system to find an appropriate number of glyph candidates. If the transcription is not available, n is estimated by the length of the text line divided by a heuristic value for the width of a single glyph.

5.1.2 Recursive region separation

Initially, all connected components \mathcal{R} are determined by means of a sequential region labeling algorithm [11]. Those connected components whose size is below a certain threshold Θ_{noise} are conceived as noise and are removed from the set \mathcal{R} . Then, the recursive separation algorithm f is applied which is a binary function of the set of all regions \mathcal{R} and the given number of desired glyphs n or their estimation if that number is not available:

$$f(\mathcal{R}, n) = \begin{cases} \mathcal{R} & \text{if } |\mathcal{R}| \geq n \\ f(\mathcal{R} \setminus r, n - n_r) \cup f(h(r), n_r) & \text{else} \end{cases}$$

with $r = \max(\mathcal{R})$, $h : r \rightarrow 2^r$, $n, n_r \in \mathbb{N}$.

The algorithm terminates if there are at least as many regions as glyphs desired: $|\mathcal{R}| \geq n$. If there are not yet enough regions, the largest one r is splitted into small subregions by the auxiliary function h and the result is passed to f together with an estimation of the number of glyphs n_r fitting into r : $f(h(r), n_r)$; thereby, 2^r denotes the set of possible partitions of r . In order to get to the next largest region, f is also applied to \mathcal{R} without the largest region: $f(\mathcal{R} \setminus r, n - n_r)$. Both recursion results are put together when they pop back upwards.

5.1.3 Splitting r into subregions

The actual separation algorithm for a single region r is defined by h . The rows and columns are sampled for r , and those paths that have a length below a certain threshold Θ_β are marked to be candidates \mathcal{C}_x for boundaries between glyphs within the rows of r :

$$(4) \quad \mathcal{C}_x = \bigcup (p_{x,y}, \dots, p_{x+k,y}), k \leq \Theta_\beta, \forall p \in r \wedge \forall p_{x+l,y} = 1, l = 0 \dots k \wedge p_{x-1,y} = 0 \wedge p_{x+k+1,y} = 0$$

Background pixels are denoted by 0 and foreground pixels by 1. Thin paths along the columns of r are determined accordingly:

$$(5) \quad \mathcal{C}_y = \bigcup (p_{x,y}, \dots, p_{x,y+k}), k \leq \Theta_\beta, \forall p \in r \wedge \forall p_{x,y+l} = 1, l = 0 \dots k \wedge p_{x,y-1} = 0 \wedge p_{x,y+k+1} = 0$$

In general, this process results into regions instead of paths, because very often there are a number of thin paths with lengths below Θ_β side by side. Therefore, approximately the middle of such a region is taken for splitting r into subregions. The emerging boundaries run more or less transverse depending on how adjacent paths differ with respect to their lengths and relative positions. But they might also be perfectly vertical or horizontal depending on the orientation of those paths. Since Θ_β determines what is conceived as thin, this parameter can be adapted to different handwritings.

Whenever resulting subregions are getting too small they are added back to one of the other adjacent regions. For this purpose, the size of any region is controlled by another parameter Θ_A which avoids an over-segmentation with too many small regions. This is necessary as many available digital scans have a high resolution so that even noisy regions contain 50 or even more pixels. This parameter can be adapted to different documents depending on their quality.

5.1.4 Optimisations

The implementation of this algorithm is iterative due to stack overflow problems which arise as soon as the recursion gets too deep. This happens for long text lines with many large regions and when text lines are interspersed with descenders and ascenders of the preceding and next text line, respectively. Moreover, a number of additional data structures are employed for optimisation purposes.

5.2 User interactions

While the previously outlined algorithm generates a first solution to the glyph separation problem, the interactive part of the method allows the user to let the system know which regions need further treatment.

There are two converse operations, namely to join regions or to separate them. Both operations can be easily launched by successively moving the pressed mouse cursor over them in order to join regions and by clicking into a single region in order to separate it.

In fact, there are three different methods for separating regions. Their combination enables the user to manage every conceivable situation quite efficiently. All methods are described in turn.

5.2.1 Recursive region separation locally applied

The application of the glyph separation algorithm to a long text line will produce a perfect separation on rare occasions. However, for the user it is simple to select

all regions of the outcome that need further consideration. Selecting a region by the mouse cursor, the glyph separation method described in section 5.1 is applied to that region. Whereas this method has been developed for managing a whole text line, yet a more or less complex ensemble of regions, or just a single one, can itself be conceived as a short text line. The glyph separation method can be initialised by other parameters concerning the expected number of regions and their sizes. Those parameters adjust the glyph separation algorithm locally, and thus have a more effective impact on specific regions.

In particular Θ_β should be chosen higher for local regions, since those regions obviously have not been split beforehand when the whole text line was treated. Hence, less strong constraints where to break up a region are required. The higher Θ_β the more likely it is that a region is broken up. Moreover, a lower value for Θ_A allows to accept the separation of a region even if the resulting subregions get quite small. The higher Θ_A the more probable it is that the algorithm decides to assign small pieces again to neighbouring regions.



Fig. 4 Recursive glyph separation of the word *cabilonensi*.

The algorithm systematically determines thin paths, it takes into account the sizes of regions, and it is recursively applied to resulting regions. For the user it would be quite cumbersome to analyse the regions in this way all by herself. But the quality of the final outcome of the algorithm can in most cases be assessed quite easily, due to the systematic distribution of colours which visualise the discovered separations. Fig. 4 shows an example, in which the user only had to click once into the first yellow region (a) and where it was necessary to join the first two regions as well as the following two pairs of regions (b) in order to get the first three letters properly segmented (c).

5.2.2 Line separation

There are a number of cases which are particularly challenging, for example when glyphs are very close to each other so that there is no clear transition between adjacent glyphs, when ligatures are to be separated into their constituent parts, or when there are imperfections, such as blobs or holes in the paper, smearing the boundaries of the glyphs. Yet another common problem are nearby text lines. In this case, descenders are overlapping with ascenders of the next text line. The heuristics of the recursive region separation algorithm are generally not very successful in such cases.

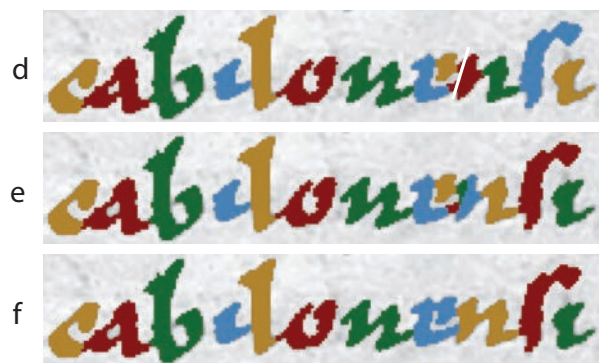


Fig. 5 Corrections by the line separation method.

The line separation method allows the user to draw a line along which a region is supposed to be separated into two pieces. In this way, even oblique transitions between glyphs can be dealt with. The line might even cross a couple of adjacent regions all at the same time. As a consequence, they are simultaneously divided into twice as many pieces as there have been regions before. In Fig. 5 the example of Fig. 4 gets completed: The top image (d) shows in particular the white line which has been drawn by the user, in order to separate the letters 'e' and 'n' (e); resulting fractions are assigned to their glyphs (f).

A separation line can be drawn wherever desired in order to divide regions. The sole exception is that such a line needs to start in the background and that it ends somewhere else in the background. This is to ensure that such a line does not end up in the midst of a region; if it would end up there, not all boundaries of the intended regions could be determined. Provided that the user does not care about this restriction the triggered method will simply do nothing.

The line separation method can always be applied, but needs some effort and carefulness by the user. By contrast, the recursive region separation method is much easier to employ, but fails for some complex regions.

Just in such cases, the employment of the line separation method resolves what the automatic algorithm is not able to handle.

5.2.3 Square separation

In the last resort, if none of the previous methods yields satisfying results, a region can be regularly tessellated into small squares. This enables the user to define any possible region by joining all squares which are assumed to belong to the same glyph. The user can specify the lateral length of the squares in order to define the glyph at an arbitrarily fine scale. Fig. 6 shows the tessellation of the letter ‘p’ with a lateral side of 2 and 11 pixels on the left hand side and in the middle, respectively. The right hand side of Fig. 6 shows the usual approach, namely to divide a region into a few large subregions and to further divide a subregion into finer squares wherever it seems necessary to consider more details. As the mouse cursor is just to be moved over all regions which are to be connected, such tessellations can be easily transformed to coherent glyphs.

This is another example that shows the effective interplay of user and system. While a tessellation can be efficiently computed automatically for arbitrary regions, the user can easily combine those squares which are thought to pertain to the same glyph. Defining fine details by drawing corresponding paths through the regions would be much more intricate for the user. In this case, she would have to take care herself of the precise locations of any details. In contrast, with the help of the square separation method she has just to assign to each part its corresponding region, by hovering the mouse cursor over it.



Fig. 6 The tessellation of the letter ‘p’ at two different granularity levels. From left to right the squares have a lateral length of 2, 11, and intermixed, 2 and 11 pixels.

Note that the user interface provides a colour-coordinated display of adjacent regions, as shown in Figures 4, 5, and 6. The display iterates periodically over the colours yellow, red, green, and blue. Among others, this helps in identifying tiny pieces between adjacent

glyphs which have been assigned neither to any glyph nor to the background. The latter makes sense when those pieces represent noise.

5.3 Conclusion

The Diptychon method presented here has the advantage of working without the deployment of dictionaries or language models. It is therefore language-independent. Recursive region separation basically applies to quill created writings which often show thin parts in the transient zones between adjacent glyphs. In these cases, this algorithm will often yield satisfying region partitions. Otherwise, line separation and square separation will always be applicable.

6 A case study in glyph separation

A user independent evaluation consists in counting the number of interactions necessary in order to arrive at a proper separation of glyphs. As a prerequisite it is necessary to detect the text lines as described in section 4.4 and to adjust them as described in section 4.2.

6.1 Material

The methodology has been applied to two different document pages of a presumably single writing hand, that one of the 11th century chronicler Hugh of Flavigny [21]. In terms of palaeography, this writing hand is a late Carolingian Minuscule used in books.

Two pages of that chronicle are compared. The first one, page eleven, is almost at the beginning of the book and shows a neat and regular writing style. The second page, number 144, is much more irregular. Among others, it shows how text lines drop off towards their ends. Both pages have been deliberately chosen by a palaeographic researcher in order to provide two very different document pages. Samples of both document images are found in Fig. 7.

A second case study analyses three different handwritings from the 9th, 13th, and 18th century which are found in the IAM Historical Document Database [6, 8]. They exemplify handwritings in Latin, German, and English. This database is accessible to the public, and thus, the results can be taken as a reference with which others can compare themselves.

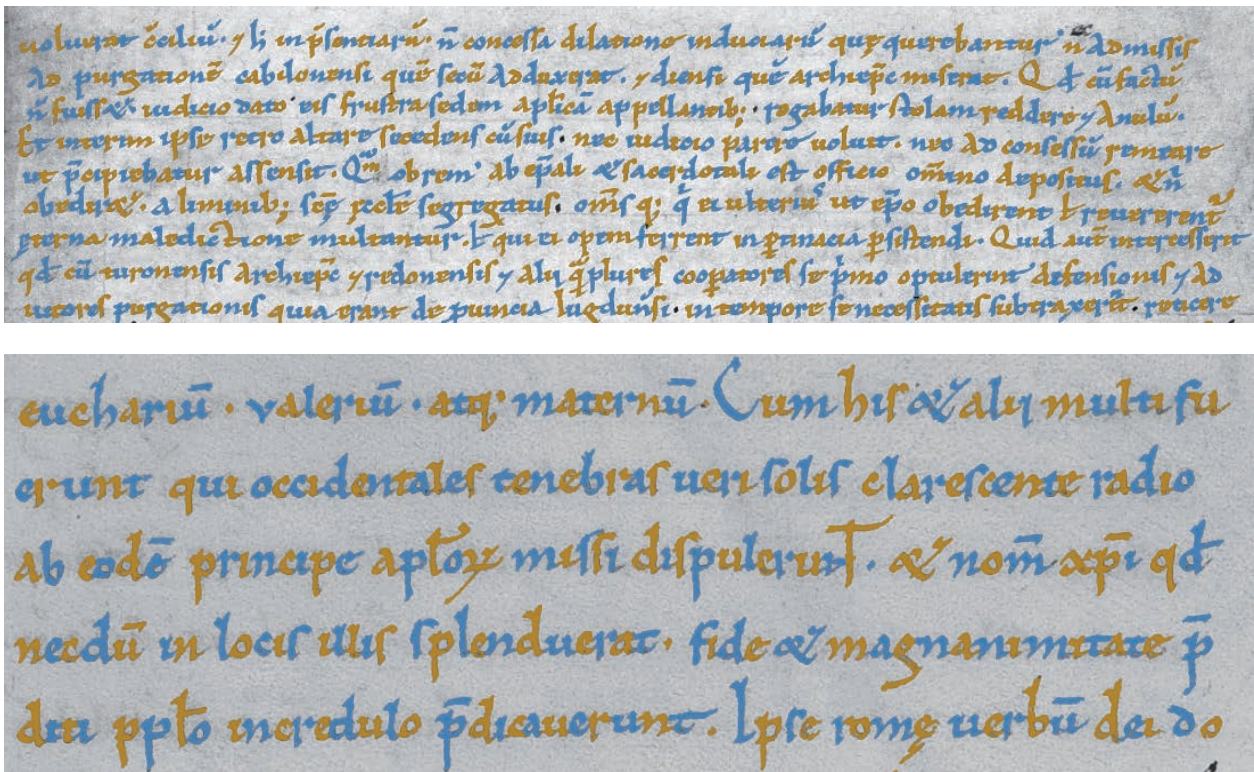


Fig. 7 Cutouts of documents A (bottom), p. 11 [21], and B (top), p. 144 [21], approximately at the same scale.

6.2 Methods to be evaluated

In each document at least the first thousand glyphs are to be properly separated. For this purpose, the recursive glyph separation algorithm, described in section 5.1, is applied to each text line. Corrections consist either in separating regions or in joining them. The number of necessary corrections are to be counted.

Both methods, recursive region separation and the joining of regions, are user independent in that it makes no difference where to click into regions or where to grasp them in order to separate or join regions, respectively. By contrast, the line separation algorithm can result in different glyph separations, however, the user can withdraw any interaction and repeat it until the glyphs are properly separated. In this sense, even the line separation interaction can be employed for a user independent evaluation.

The square separation interaction is omitted in this study. It is useful in order to deal with arbitrary complex handwritings. However, there are two reasons why the square separation method costs much more effort than the other separation interactions. At first, the user needs to determine an appropriate granularity level for tessellating a region in such a way that it is detailed enough. Secondly, the user has to join all tesserae which

connect to a proper glyph. This amounts to hover the mouse device over all those tiny tesserae consecutively which are part of the same glyph. The other separation interactions are either confined to a single mouse click or to hover the mouse over adjacent regions which are much larger than small tesserae, and therefore, can be grasped more easily.

6.3 The interaction ratio

It is the purpose to determine the effort to correct glyph separations of handwritten documents after the automatic glyph separation algorithm has processed a single text line. This effort is expressed by means of the ratio of interactions required and objects (glyphs) to be considered. It is referred to as the *interaction ratio*:

$$(6) \rho = \frac{\text{number of interactions}}{\text{number of objects}} \in \mathbb{R}_0^+$$

The lowest value of zero states that no interactions are required and the higher ρ the more interactions are necessary. A value below one means that less interactions are needed than there are objects which are to be taken into account, while a value of 1 means that in the average each glyph needs to be touched once, and if $\rho > 1$, that it needs to be touched more often.

6.4 Evaluation of all interaction methods

As expected it shows that there are many less interactions necessary in document A that exhibits a more regular writing style than document B (cf. Fig. 7). For document A it holds $\rho_A = 0.65$ and for document B it holds $\rho_B = 0.76$. The glyphs do not separate as clearly in document B as in document A. Note that there are complex cases in which two or more interactions are necessary to correct a single glyph. The values 0.65 and 0.76 are just averages.

For document A, 8% of the glyphs have been divided by recursive region separation, while almost twice as many glyphs, namely 15%, had to be split into two parts by means of the line separation method. The latter allows the user to more precisely indicate where a region is to be broken up. It requires only a little bit more effort for the user than the other method. For recursive region separation the according region just needs to be selected somewhere, while the line separation method requires the user to draw a line which indicates the path along which the region is to be broken up. For document B, there are 13% of the glyphs for which recursive region separation had been applied and 23% of the glyphs for which line separation was required.

Interestingly, it shows that more joining operations are necessary in the case of the more regular handwriting. This is the case due to the higher ratio of fragments that are automatically determined by the preprocessing algorithm. Over-segmentation is higher for document A. From the correction interactions in document A 64% are joining operations, in document B there are 53% such operations. For both documents they dominate the modes of interactions.

Although the text lines in document B are quite close to each other, there were only seven interactions necessary for separating descenders from ascenders of following text lines. For document A there are only two such occurrences. They have all been resolved by line separation. Table 1 summarises the main results.

Table 1 1032 and 1025 glyphs have been analysed for documents A and B, respectively. The number of interactions have been determined: region separation (RS), line separation (LS), and joining. The sum of interactions and the interaction ratio ρ are also given.

	RS	LS	Joining	Σ	ρ
A	83	155	433	671	0.65
B	133	233	417	783	0.76

In Table 1 the interaction ratios are given for the entire documents. In order to analyse how ρ differs across single documents, this value has also been computed for each text line separately. It is not only assumed that this value is lower for the text lines of document A but that it is similar among text lines within the same document. The upper part of Table 2 shows the results. The small standard deviations σ_ρ in both cases support the assumption that the text lines within the documents are similar in the way how glyphs connect. Interestingly, the standard deviation for the more regular document is higher than for the other document. But this is in accordance with what has been mentioned above about the over-segmentation of document A.

Table 2 Range, mean, and standard deviations for documents A and B after having computed the interaction ratio for all interactions for each text line separately; in the two bottom rows confined to both separation methods.

	ρ_{Min}	ρ_{Max}	μ_ρ	σ_ρ
A	0.43	0.91	0.66	0.12
B	0.57	0.90	0.77	0.10
A'	0.11	0.33	0.23	0.06
B'	0.16	0.57	0.36	0.12

6.5 Evaluation of the separation methods

There are two reasons why to reevaluate the data by only looking at the separation methods. First, they cost more effort than the joining operations in that the user has to decide which separation method she wishes to apply and because line separation needs some carefulness by deciding where to separate a region, while neither recursive region separation nor the joining of regions leave any decisions to the user. Secondly, an over-segmentation of document A requires a large number of joining operations and leads to the wrong impression that document A is more complex than document B, since the total interaction ratio makes no difference between separation and joining operations.

The analysis of the glyph separation problem, however, should mainly take into account the number of actual separation methods to be applied. Focusing on the separation methods it is expected that document A shows a lower interaction effort than document B.

The lower part of Table 2 shows the results. Clearly, the interaction effort is lower for document A. Its mean is 0.23 and the mean for document B is 0.36. The difference is lower for the previous evaluation shown in the upper part of the table. Moreover, the standard

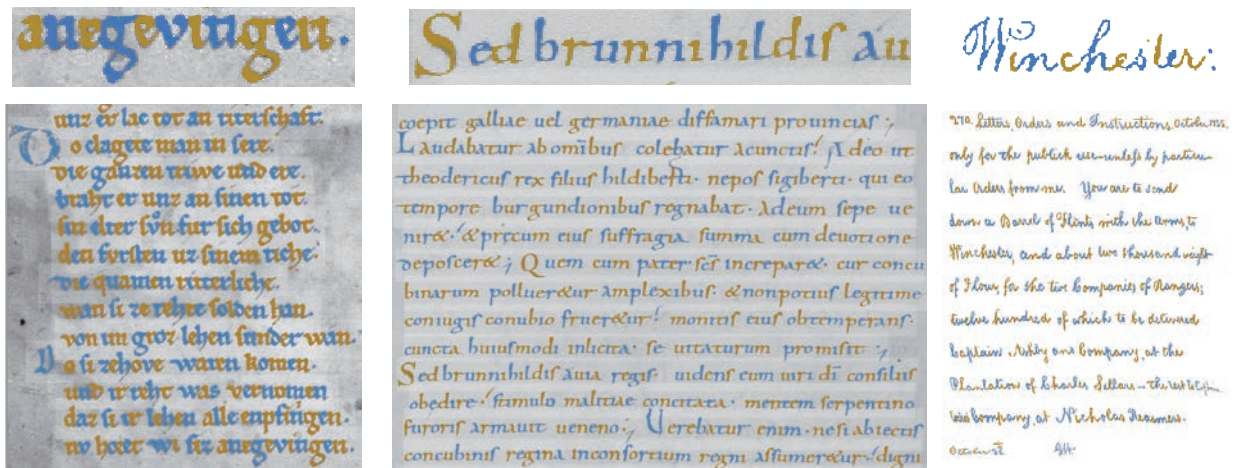


Fig. 8 Excerpts of the documents of the IAM Historical Document Database; from left to right: documents D, C, and E. The changing colours of the glyphs show how they got separated. At the top there are enlarged cutouts for each handwriting.

deviation is now lower for document A, showing how the effort for the joining operation offered a disputable comparison of the interaction ratio applied to both documents. In both cases, the upper bound for the number of necessary interactions clearly dropped down.

6.6 The IAM Historical Document Database

There are three different handwritings contained in this database. They have been analysed in the same way as documents A and B:

- C: The manuscript images of the Codex Sangallensis 562, St. Gallen, Stiftsbibliothek, 9th century, Latin writing, page 3 [8].
- D: The Abbey Library of Saint Gall, Cod. 857, 13th century, German writing, page 6, column a [6].
- E: The Library of Congress, George Washington Papers, Series 2, Letterbook 1, 18th century, English writing, page 270 [8].

Fig. 8 shows samples of those documents. The evaluation indicates that in document D, which is shown on the left hand side of that figure, the separation of glyphs is most difficult. There are in the average 1.44 operations necessary in order to correct a glyph of document D, as denoted by the interaction ratio shown in Table 3. The effort for corrections regarding document C is close to the interaction ratios of documents A and B. Document E shows the most recent handwriting of the analysed data. Its interaction ratio lies somewhere in the middle of the interaction ratios of the older handwritings.

Table 3 1123, 1064, and 1015 glyphs have been analysed for documents C, D, and E, respectively. The number of interactions have been determined: region separation (RS), line separation (LS), and joining. The sum of interactions and the interaction ratio ρ are also given.

	RS	LS	Joining	Σ	ρ
C	150	203	490	843	0.75
D	479	348	704	1531	1.44
E	247	91	564	902	0.89

The joining operations are dominant for all three documents as for documents A and B, followed by line separation for document C. Only for documents D and E the local region separation algorithm has been more often applied than line separation.

6.7 Discussion

In the presented study all handwritings have been analysed with the same parameter settings. This concerns the removal of noise in the beginning, the binarisation parameters, as well as the region separation parameters that define the acceptable thickness of glyph transitions and the minimum size of region fragments to be created when getting down to small pieces during recursion. In other words, the results show the performance of the algorithm when it is not optimised for a given handwriting. The trained user, however, can adapt the parameter settings for a specific handwriting, or even during the processing of a single document, to different parts of the same document. This again shows the extent of the anytime anywhere paradigm which also allows the adjustments of parameters to single regions,

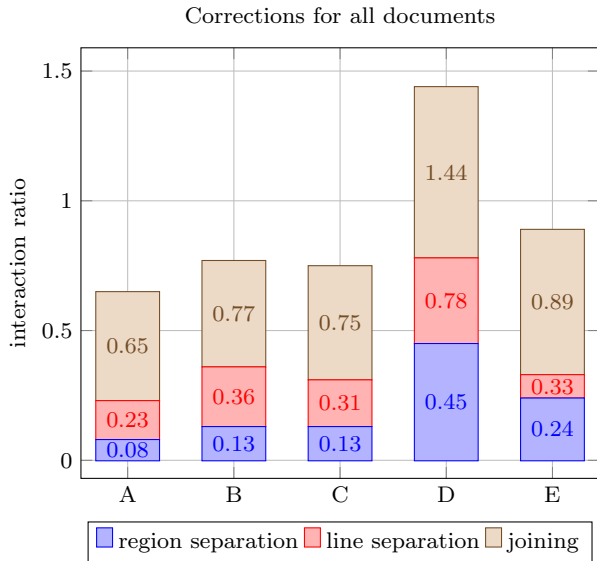


Fig. 9 For all five documents the interaction ratios are accumulated for all three modes of correction. For instance, the stacked bar for document A reads: there are 0.08 region separations, 0.23 region plus line separations, and 0.65 total operations per glyph in the average, including joining operations. The heights of the three component bars show the contributions of the three interaction methods.

depending on the quality of any location on a document image. For old documents this becomes of particular importance due to their degraded quality.

Fig. 9 summarises the interaction ratios. It shows that the interaction ratio is quite similar for all documents with the exception of document D that costs the most effort. Each stacked bar, as a whole, coincides with the according value for ρ , as each of the three components relates to the average effort per glyph. But note that the numbers contained within the bars are accumulated. In this way, the separation effort alone can be read of that diagram, as motivated in section 6.5: The lowest effort is 0.23 separations per glyph for document A while the highest is 0.78 separations per glyph for document D.

All correction methods are confined to either click somewhere into a region (for region separation), to hover the mouse cursor over two or more regions (in order to join regions), or to indicate the start point and end point of a line (line separation). Therefore, the temporal effort of those interactions is rather low. For all documents the time required in order to correct a whole text line has been measured. As the text lines for document D are quite short, two text lines with 25 glyphs each have been taken. In the other cases, the number of glyphs contained in the text lines range from 45 to 77. However, the time required for a single glyph has been determined by the ratio of time and number of glyphs

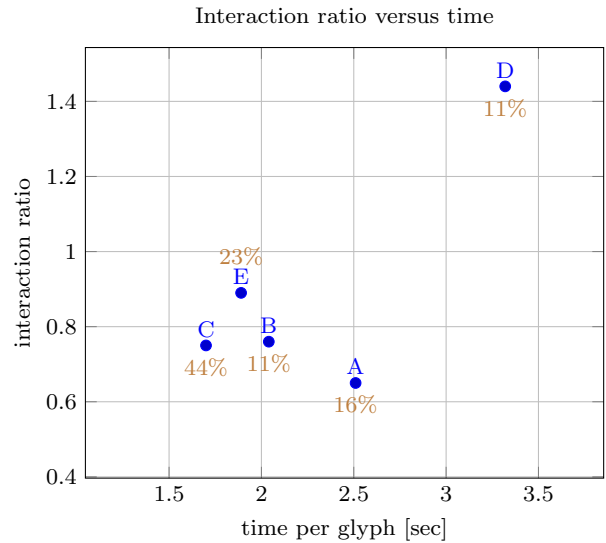


Fig. 10 For all five documents the interaction ratio is related to the average time spend on correcting glyph separations. The numbers next to the data points of the documents indicate the percentage amount of correct glyphs which have been found by the recursive region separation algorithm, before having made any corrections.

in a text line. For all documents the temporal range lies between $1.70 \frac{\text{sec}}{\text{glyph}}$ and $3.32 \frac{\text{sec}}{\text{glyph}}$.

In order to compare the required time with the interaction effort ρ both quantities are related in Fig. 10. It turns out that the temporal effort per glyph is quite similar for all documents, though Fig. 10 suggests that document D is much more difficult to handle (in accordance with Fig. 9). However, there is only a difference less than a second to the temporal effort for document A. As expected, the higher the interaction effort the higher the temporal effort, though this is just a rough tendency with small deviations for documents A and B. A clear linear dependence can hardly be expected, given the small temporal differences.

The only meaningful conclusion that could be drawn is that in the average there is an effort of approximately two seconds per glyph. For a text line with 25 glyphs (such as in Document D) this amounts to an overall effort of around one minute for the whole text line and for long text lines with 77 glyphs (as for Document B) the effort lies around two and a half minutes. This refers to the expertise of a trained user.

7 General discussion

When sticking to the image processing pipeline as a one-way road, the user is forced to analyse documents in a strictly artificial way. By contrast, the interactive

anytime anywhere document analysis paradigm put forward in this paper, allows the user a natural way of dealing with documents – a paradigm probably applicable in other domains as well. A preprocessing step can be locally applied whenever necessary, as can be the adjustment of text lines or the separation of glyphs in a word. These steps can be performed for a whole document page at a single stroke or for parts of it separately in an arbitrary order. The palaeographic researcher is carefully inspecting documents while trying to read the handwriting and while analysing its characteristics. In doing so, she can immediately apply any image processing functions in order to neatly extract and separate all glyphs, which are needed for a standardised and detailed document statistic.

This approach complements other interaction methodologies which either require a formal model of documents to be processed or which are confined to interactions within single processing steps. The advantage of the anytime anywhere analysis paradigm is that it does not make any assumptions about document structures, contents, and languages. It is applicable to any kinds of document images and bears resemblance to commercial picture editing software except that it is optimised and restricted solely to document image processing. A drawback is that it is less effective than interaction methodologies that deploy models of document structures or languages, and hence, provide more accurate suggestions to the user. Another distinction to other methodologies is that the user can directly apply interaction methods whenever she thinks that automatic methods would not provide useful suggestions, instead of letting the system deal with particular complex cases which would be difficult to adjust afterwards. In such cases the roles are reversed: The user edits difficult parts of the document, while the system continues to process that document afterwards.

In particular, a number of interactive methods have been introduced that enable the separation of glyphs for arbitrary complex handwritings. The results are useful for both palaeographic research and for generating ground truth data sets, for example for classification evaluations based on characters. The interaction ratio allows the quantification of the effort that is necessary to manage different documents. As far as the assumption is valid that a handwriting style is distinguished by the way how well its glyphs can be separated, the interaction ratio also applies as a measure of similarity. But this needs to be investigated in future work more carefully.

That letters can always be meaningfully separated into individual glyphs is doubtful. Some handwritings are highly connected and lack clear boundaries between

glyphs. Indeed, some schools of palaeography reject the validity of glyph based comparisons and look instead at the broader context of whole words or characteristics like the movements of the pen. But even if the boundaries between glyphs cannot always be determined, frequently there is at least a subset of glyphs with clear boundaries. Such subsets can be used for a glyph based comparison by means of shape features [9]. Additionally, a glyph based transcription enables the search for any character strings, even if the boundaries between glyphs have only been determined inaccurately [29]. Eventually, having separated only a subset of all glyphs the transcription of that glyph subset could significantly help the user in transcribing poorly legible handwritings, since the Diptychon system presents all recognised glyphs in the context of all illegible word fragments.

8 Summary

Algorithms for the separation of glyphs of handwritings have been presented, together with a new interaction paradigm that provides a flexible editing tool for historic documents. It includes image enhancement, figure-ground segmentation, text line detection, glyph separation, and transcription methods that can be applied in any way desired, in order to deal with whole documents or parts of it.

An emphasis is put on the glyph separation problem for which several different methods are made available to the user. The only alternative for those methods is the employment of conventional picture-editing software which requires the fully manual separation of glyphs. In this sense, the suggested methods provide a significant improvement. The underlying philosophy of the approach derives from the observation that fully-automated systems are not always successful or sometimes even not necessarily desirable. Instead, a methodology is put forward that seeks to bring together the advantages of automatic methods and skills of human experts.

Acknowledgements The described methods have been realised within the Diptychon system which has been implemented to a large extent by Jan-Hendrik Worch. We are thankful for his valuable contributions to this project. Moreover, we acknowledge the support of Prof. Dr. Michael Menzel, Lehrstuhl für Mittelalterliche Geschichte, Humboldt-Universität zu Berlin, and Prof. Dr. Michael Lawo, Centre for Computing and Communication Technologies, University of Bremen. Finally, we appreciate the valuable feedback provided by the reviewers.

References

1. Brockhaus: Graphem. In: Der neue Brockhaus, vol. 2, p. 436. F. A. Brockhaus Wiesbaden (1979)
2. Casey, R.G., Lecolinet, E.: A survey of methods and strategies in character segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(7), 690–706 (1996)
3. Chazalon, J., Couasnon, B.: Iterative analysis of document collections enables efficient human-initiated interaction. In: C. Viard-Gaudin, R. Zanibbi (eds.) *Document Recognition and Retrieval XIX - DRR 2012*, 19th Document Recognition and Retrieval Conference, part of the IS&T-SPIE Electronic Imaging Symposium, Burlingame, California, USA, January 25, 2012, Proceedings, *SPIE Proceedings*, vol. 8297. SPIE (2012)
4. Chazalon, J., Couasnon, B., Lemaitre, A.: Iterative analysis of pages in document collections for efficient user interaction. In: *International Conference on Document Analysis and Recognition, ICDAR 2011*, Beijing, China, September 18–21, 2011, pp. 503–507. IEEE (2011)
5. Clavier, E., Masini, G., Delalandre, M., Rigamonti, M., Tombre, K., Gardes, J.: Docmining: A cooperative platform for heterogeneous document interpretation according to user-defined scenarios. In: J. Lladós, Y.B. Kwon (eds.) *Graphics Recognition, Recent Advances and Perspectives*, 5th International Workshop, GREC 2003, Barcelona, Spain, July 30–31, 2003, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 3088, pp. 13–24. Springer (2004)
6. Fischer, A., Frinken, V., Fornés, A., Bunke, H.: Transcription alignment of latin manuscripts using hidden markov models. In: *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing, HIP '11*, pp. 29–36. ACM, New York, NY, USA (2011)
7. Fischer, A., Indermühle, E., Bunke, H., Viehhauser, G., Stolz, M.: Ground truth creation for handwriting recognition in historical documents. In: D.S. Doermann, V. Govindaraju, D.P. Lopresti, P. Natarajan (eds.) *The Ninth IAPR International Workshop on Document Analysis Systems, DAS 2010*, June 9–11, 2010, Boston, Massachusetts, USA, ACM International Conference Proceeding Series, pp. 3–10. ACM (2010)
8. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character hmms. *Pattern Recogn. Lett.* **33**(7), 934–942 (2012)
9. Gottfried, B.: Qualitative Similarity Measures - The Case of Two-Dimensional Outlines. *Computer Vision and Image Understanding* **110**(1), 117–133 (2008)
10. Gottfried, B., Meyer-Lerbs, L.: Towards the processing of historic documents. In: Bernadi, R. et al. (ed.) *Advanced technologies for digital libraries.*, *LNCS*, vol. 6699, pp. 15–28. Springer (2011)
11. He, L., Chao, Y., Suzuki, K.: A run-based two-scan labeling algorithm. *IEEE Transactions on Image Processing* **17**(5), 749–756 (2008)
12. Hofmeister, W., Hofmeister-Winter, A.: Schriftzüge unter der High-Tech-Lupe. Theoretische Grundlagen und erste praktische Ergebnisse des Grazer Pilotprojekts DAMaLS. In: *Internationales Jahrbuch für Editionswissenschaft*, vol. 22, pp. 90–117 (2008)
13. Kansal, H., Sanyal, S., Gupta, D.: Dewarping and deskewing of a document using affine transformation. In: A. Ranchordas, H. Araújo (eds.) *VISAPP (2)*, pp. 73–78. INSTICC Press (2009)
14. Kim, G., Govindaraju, V., Srihari, S.N.: An architecture for handwritten text recognition systems. *IJDAR* **2**(1), 37–44 (1999)
15. Lebourgeois, F., Emptoz, H.: DEBORA: Digital AccEss to BOoks of the RenAissance. *IJDAR* **9**(2–4), 193–221 (2007)
16. Lowe, K.A.: From quill to t-pen: Palaeography, editing and their e-futures. *Literature Compass* **9**(12), 1004–1009 (2012)
17. Moalla, I., Lebourgeois, F., Emptoz, H., Alimi, A.M.: Contribution to the discrimination of the medieval manuscript texts: Application in the palaeography. In: H. Bunke, A.L. Spitz (eds.) *Document Analysis Systems, LNCS*, vol. 3872, pp. 25–37. Springer (2006)
18. Nagy, G.: Twenty years of document image analysis in pami. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 38–62 (2000)
19. Ouwayed, N., Belaïd, A.: A general approach for multi-oriented text line extraction of handwritten documents. *IJDAR* **15**(4), 297–314 (2012)
20. Peck, A.: *Beginning GIMP: From Novice to Professional*. Apress Inc. (2006)
21. Philipps: fol. 11v and fol. 144r (1870). Staatsbibliothek zu Berlin, Preußischer Kulturbesitz
22. Plötz, T., Fink, G.A.: Markov models for offline handwriting recognition: a survey. *IJDAR* **12**(4), 269–298 (2009)
23. Ramel, J.Y., Sidère, N., Rayar, F.: Interactive layout analysis, content extraction and transcription of historical printed books using pattern redundancy analysis. *Literary and Linguistic Computing* **28**(2), 301–314 (2013)
24. Romero, V., Toselli, A.H., Rodríguez, L., Vidal, E.: Computer assisted transcription for ancient text images. In: M.S. Kamel, A.C. Campilho (eds.) *ICIAR, LNCS*, vol. 4633, pp. 1182–1193. Springer (2007)
25. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. *Pattern Recognition* **33**, 225–236 (2000)
26. Serrano, N., Gimnez, A., Civera, J., Sanchis, A., Juan, A.: Interactive handwriting recognition with limited user effort. *IJDAR* pp. 1–13 (2013)
27. Serrano, N., Tarazón, L., Pérez, D., Terrades, O.R., Juan, A.: The gidoc prototype. In: A.L.N. Fred (ed.) *Pattern Recognition in Information Systems, Proceedings of the 10th International Workshop on Pattern Recognition in Information Systems, PRIS 2010*, In conjunction with ICEIS 2010, Funchal, Madeira, Portugal, June 2010, pp. 82–89. SciTePress (2010)
28. Smith, R.: A simple and efficient skew detection algorithm via text row accumulation. In: *ICDAR*, pp. 1145–1148. IEEE Computer Society (1995)
29. Worch, J.H., Lawo, M., Gottfried, B.: Glyph spotting for mediaeval handwritings by template matching. In: C. Concolato, P. Schmitz (eds.) *ACM Symposium on Document Engineering*, pp. 213–216. ACM (2012)
30. Wüthrich, M., Liwicki, M., Fischer, A., Indermühle, E., Bunke, H., Viehhauser, G., Stolz, M.: Language model integration for the recognition of handwritten medieval documents. In: *ICDAR*, pp. 211–215. IEEE Computer Society (2009)
31. Yacoub, S.M., Saxena, V., Sami, S.N.: Perfectdoc: A ground truthing environment for complex documents. In: *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005)*, 29 August - 1 September 2005, Seoul, Korea, pp. 452–457. IEEE Computer Society (2005)