Image-Matching for Revision Detection in Printed Historical Documents

Joost van Beusekom[1], Faisal Shafait[2] and Thomas M. Breuel[1]

[1] Department of Computer Science, Technical University of Kaiserslautern
D-67663 Kaiserslautern, Germany
joost@iupr.dfki.de, tmb@informatik.uni-kl.de
[2] Image Understanding and Pattern Recognition (IUPR) research group
German Research Center for Artificial Intelligence (DFKI) GmbH
D-67663 Kaiserslautern, Germany
faisal@iupr.dfki.de

Abstract. In the research area of historical documents it is of high interest to reconstruct the process of the emergence of a historical typesetted document. Therefore, the chronological order of the different versions of a typesetted document has to be reconstructed. This is done by manually finding differences in two versions and then deciding on the order between these two versions. In this paper we present an approach to automate the search for differences in both images. This approach uses a globally optimal image matching technique to overlay both images and colors the differences accordingly. We also present a real-world application for this approach on digitized versions of a historical book.

We wish to thank Prof. Wolfgang Neuser from the Technical University of Kaiserslautern for the interesting problem he presented to us and also for the valuable data we obtained.

1 Introduction

For historians it is of interest to see how typesetted historical documents have evolved over different versions. At that time, printing a book was mostly a manual process: each letter of each page had to be typesetted manually and printing had also to be done manually. This allowed the typesetter to change characters or even words between the different printings of the book. These modifications allow today's historians to detect the chronological order of the printing of the books.

The process of chronologically sorting the versions starts with a very basic task: finding differences in the two versions. Currently this process has to be done manually: one person reads a version aloud and the other person checks whether the second version contains the same text or not.

This process is costly and time consuming. This first approach to automate the process would be the use of optical character recognition (OCR): however,

(a) Textual noise (b) Noise speckles (c) Missing parts

(d) Broken characters

Fig. 1. Examples for different types of defects in historical document images.

an OCR-based approach does not work, as current optical character recognition systems do not give reliable results on historical document images. One main problem is the missing support for old fonts. Furthermore, textual noise from the facing book page, presence of many speckles, missing parts of the page and broken characters present frequent and challenging image defects that will further reduce OCR performance. Examples can be found in Figure 1.

Therefore, as a first step to automate this process, we present a method for visualizing differences on a pixel basis in the two documents. The resulting image allows the operator to quickly find relevant differences.

Another strong constraint is that, in our case, only limited influence on the digitization process is possible: some versions are scanned from microfilm, others from paper-based copies. Most images are available in black and white only. This dramatically reduces the available methods for noise removal and quality improvement for degraded document image, as many methods dealing with historical documents work on grey-scale images.

Considering all these problems, we concluded that first of all a global matching between two versions of the same document image has to be established. Therefore, scale, rotation and translation parameters have to be found that allow matching both images. This matching can then be used in later steps to allow comparison of smaller regions or even characters or parts of characters.

Visualizing the differences in historical document images by image matching, as presented in this paper, is closely related to image registration. Many different approaches have been presented for various kinds of tasks: in the field of medical

images [MV98] solutions to many practical problems could be found. But also for many other applications, much work has been done. A good overview in this domain can be obtained in [Bro92] and [ZF03].

In the area of document image understanding, different methods for document image registration have been developed: Spitz et al. [Spi97] proposed a method for duplicate finding of document images by a text-based signature. Other methods use the OCR output for registration. Liang et al. [LDD06] propose a registration method used for mosaicing camera-based document images, where registration is done using PCA-based SIFT descriptors.

As using OCR is no option for historical document images, feature point matching, as done in [LDD06] is a promising way to solve the problem. But as our document images are available only in binarized form, other point of interest as well as other features and a more robust matching method have to be chosen.

In Section 2 our approach is explained. In Section 4 results are presented. Finally Section 5 concludes this paper. Unfortunately, due to the specialized nature of this problem and due to the absence of any ground-truthed dataset for this purpose, no quantitative evaluation could be done. An overview over the system can be found in Figure 2.

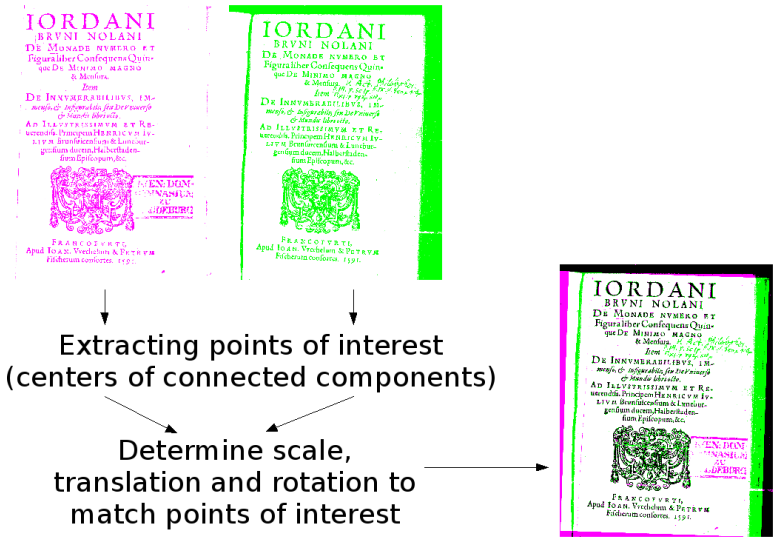


Fig. 2. System overview: first connected components are extracted as points of interest in both images. Then the optimal transformation given by scale, translation and rotation is computed to match the two sets of interest points. In the end, both images are overlaid.

2 Historical Document Image Matching

Given two images I and M. The goal is to find a set of parameters that allows to overlay both images as exact as possible. We define a perfect overlay as the set of parameters that matches the centers of all connected components of image I to the corresponding positions in image M, given that I is obtained by rotating, translating and scaling of M. Thus the quality function to optimize is the number of matching center points of the connected components. The 4 parameters that need to be found are the angle of rotation, the translation in horizontal and vertical direction and the scaling factor.

The method we used to determine the best parameters describing the transformation of the image I onto the image M is called RAST (Recognition by Adaptive Subdivision of Transformation Space) and was first presented by Breuel [Bre92]. This method is capable of finding globally optimal transformation parameters while avoiding to search the whole parameter space. This allows an optimal matching to be found and not, as currently done in many other procedures, only a locally optimal one.

RAST intelligently searches the whole parameter space $\mathcal{R} = [0, 2\pi) \times \mathbb{R}^2 \times \mathbb{R}_+^*$ for the globally optimal parameter set. The pseudo-code, taken from [Bre01], can be found in Figure 3.

The algorithm starts with enqueueing the whole parameter space (line 05). Then the region with the highest upper bound is taken from the queue (line 08). This is then subdivided into two parts (line 10 and 13). The two subregions are enqueued, if the upper bound for the quality for the subregion is higher than the currently best quality (line 19). Finally, if the remaining region is small enough, which strongly depends on the application, it is saved as possible solution.

Computing the upper bound of the quality of a parameter region is the main challenge. Given a parameter region, for each model point the possible target positions are computed and the bounding rectangle of these positions is extracted. This rectangle is used to determine if the point of interest of image M can be matched to a point of interest in image I. If this is the case, the quality is increased. Repeating this for all interest points in M leads to the upper bound for the quality. Computing this upper bound can be quite costly if the number of interest points is high. A more detailed description can be found in [Bre01].

To reduce the computing time needed to compute the upper bound for the quality, a pre-filtering step is added: instead of comparing all interest points of image M to all image interest points of image I, a pre-selection is done: only points that are "similar" are used as potential matches.

2.1 Filtering using Fourier Descriptors of the Boundary

Using Fourier descriptors to describe boundaries of connected components is a widely used method. Many examples of very different applications of this technique exist, e.g. for shape-based retrieval [ZL01]. As we want to match document images based on connected components positions, one would expect, that a connected component representing an "a" will be matched to another connected

```
01    Queue active;
02    Region result = nil;
03
04    void match() {
05      enqueue(active, initial_region);
06
07      while(not_empty(queue)) {
08        Region current = dequeue(active) ;
09
10        Region left = expand(split_left(current));
11        if (left != nil) enqueue(active, left);
12
13        Region right = expand(split_right(current));
14        if (right != nil) enqueue(active, right);
15      }
16    }
17
18    Region expand(Region r) {
19      if (quality(r) <= quality(result)) return nil;
20
21      if (region_is_small_enough(r)) {
22        result = r ;
23        return nil ;
24      }
25
26      return r ;
27    }
```

Fig. 3. Pseudo-Code for the RAST algorithm [Bre01].

component also representing an "a" and not to one representing an "x". Therefore, basing the filtering on features representing the contour of a connected component is a reasonable choice. Another advantage of the Fourier descriptors for the boundary is that they can be made less sensitive to noise by only considering the n first Fourier descriptors.

To obtain the Fourier descriptors of a connected component, the following steps have to be done:

- Step 1: Extraction of the boundary pixels. This is done using Pavlidis algorithm [Pav82]. A sequence of pixel positions is obtained.
- Step 2:"Conversion" of the contour to a sequence of complex values: a pixel position (x, y) is regarded as complex number $x + \hat{i}y$.
- Step 3: Perform the Fast Fourier Transform (FFT) on the complex number signal. The result is a sequence of complex numbers called "Fourier descriptors".

Depending on the starting position of the pixel sequence describing the contour, the Fourier descriptors change. In order to be invariant to the starting

position of the contour (this will happen frequently as images are rotated), the phase information is ignored and only the magnitude of the Fourier descriptor is used.

To define the similarity between two Fourier descriptor sequences, the mean of the differences of the magnitudes of the descriptors is taken.

For each model point only the n most similar image points are taken as possible matches, where $n = 25$ showed to be a reasonable value. The number n should be high enough to ensure that the correct match is also in the list. It should not be too high as it will increase the time needed to estimate the upper bound. An example for some components together with their most similar components can be found in Figure 4.

3 Implementation Details

To reduce the number of connected components a connected component based filtering step has been added, as the most interesting connected components are the one that represent some character. This removes components that are to small or too big compared to the mean component size.

For the similarity measure of the contours only the 64 first Fourier descriptors are used as they contain enough information to give a rough description of the contour. This number may vary depending on the resolution of the document image.

Finally, as the image size of the available images is about 3400×4400 pixels, reasonable initial parameter space is defined by -900 to $+900$ pixels for translation in horizontal direction, -600 to $+600$ pixels for translation in vertical direction, a scale factor reaching from 0.9 to 1.1 and a rotation angle from $-0.2rad$ to $0.2rad$. A wider initial search space is also possible but increases the memory and time needs for finding the optimal parameter set.

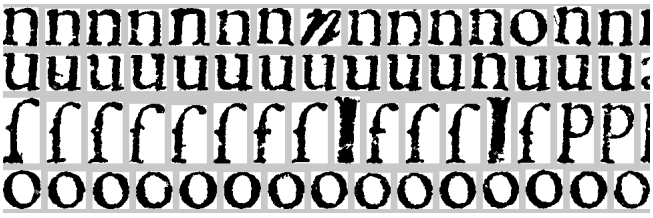


Fig. 4. Example of similarity based on Fourier descriptors of the boundary. The leftmost component in each row is a component from the image, the following are the most similar components from the model.

4 Results

We tested our method on 69 pages of the book "De monade numero et figura liber Consequens Quinque De Minimo Magno & Mensura item De innumerabilis immenso & infigurabili; seu De Universo & De Mundis libri octo" written by Bruno Giordano. Two versions of the 69 pages were available. As no ground-truth for the given document images is available and as, to our best knowledge, no publically available dataset with historical document images and ground truth is available, the only measure for success is visual inspection of the resulting images. This showed that the overlapping worked well in most cases. There are no examples where the matching was totally wrong. In some cases the overall matching was correct, but locally small discrepancies could be noticed. Example images where the matching worked well can be found in Figure 5.

In Figure 6 two examples of significant differences between the two versions of the book can be seen.

In a few images, there are parts of the page that do not fit as well as other parts. Most of these publicly matchings are due to distortions that can not be described by translation, rotation and scaling alone, as e.g. book curling. Examples can be seen in Figure 7.

A limited evaluation concerning the speed up factor obtained by using the highlighting method showed, that for two untrained persons reading the text aloud an checking for differences, about 5 minutes were needed to process a typical page (a part of the page can be seen in Figure 6 (a)). Checking the overlaid images to find the missing "e" took in mean about 1 minute. Although the number of tested persons is too small to be objective, one can conclude that this technique can speed up this process significantly.

5 Conclusion

In this paper we presented a first approach to automatically highlight differences in different versions of the same historical document. We used an globally optimal image matching technique allowing to find the optimal values for the scale, translation and rotation. Using these parameters, both images are overlaid, allowing the operator to identify quickly the differences between both versions.

As ground-truthed data is not yet available for this specialized problem, no quantitative evaluation could be done. However, we showed the usefulness of the highlighting approach for finding differences by measuring the time needed for a single person to find a word-level difference.

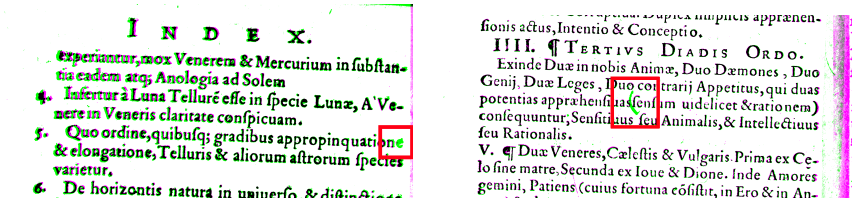
A main problem concerning this method is that it only is capable of matching images deformed by a similarity transform. This explains why curled pages are not matched perfectly (Figure 7). As dewarping curled pages is still an open problem, a local adaption of the obtained parameters on a connected component basis could be a good way to deal with this problem.

Furthermore, the current method is only applicable if the overall page is not changed by a small modification made in the text. This assumption does not

(a) (b) (c) (d)

Fig. 5. Examples of correct matches. In Figure (d) one of the two document images was severely degraded. Nevertheless the matching overlaid the two images well.

hold for modern documents, as often changing a word results in different line breaks and also a different number of lines. But for modern document images, in contrast to historical document images, OCR is in most cases reliable which allows a string-based comparing of two versions of a document.



(a) Example of a difference (b) Example of a difference

Fig. 6. Examples of resulting images containing differences. The differences are marked with a red rectangle.

References

[Bre92] T. M. Breuel. Fast Recognition using Adaptive Subdivisions of Transformation Space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 445–451, 1992.

[Bre01] Thomas M. Breuel. Implicit manipulation of constraint sets for geometric matching under 2d translation and rotation. In *Scandinavian Conference on Image Analysis (SCIA), Bergen, Norway*, 2001.

[Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, 1992.

[LDD06] Jian Liang, Daniel DeMenthon, and David Doermann. Camera-based document image mosaicing. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 476–479, Washington, DC, USA, 2006. IEEE Computer Society.

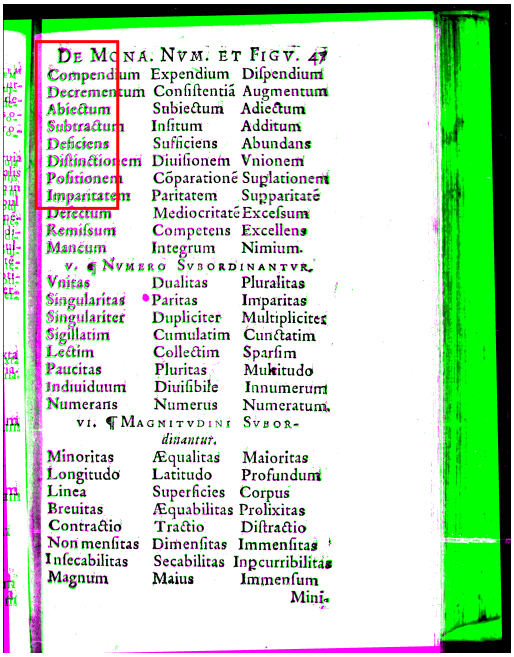
[MV98] J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, April 1998.

[Pav82] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, MD, 1982.

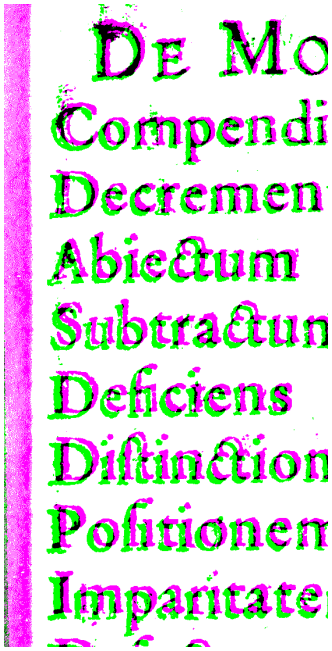
[Spi97] A. L. Spitz. Duplicate document detection. In L. M. Vincent and J. J. Hull, editors, *Proceedings SPIE*, volume 3027 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 88–94, March 1997.

[ZF03] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.

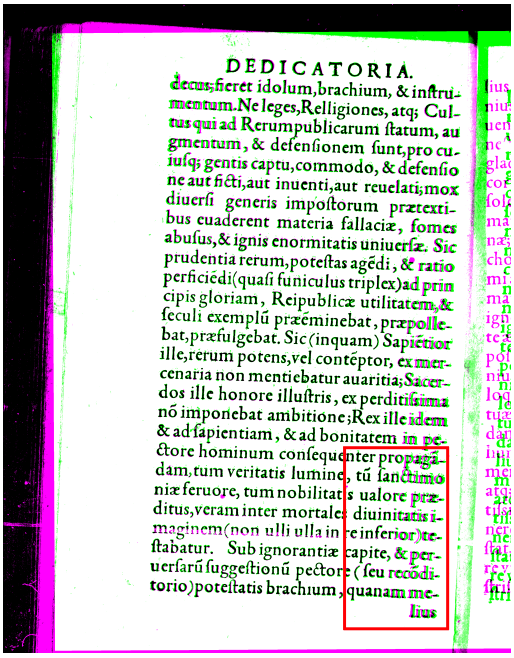
[ZL01] D. S. Zhang and G. Lu. A comparative study on shape retrieval using fourier descriptors with different shape signatures. In *Proc. of International Conference on Intelligent Multimedia and Distance Education (ICIMADE)*, pages 1–9, June 2001.



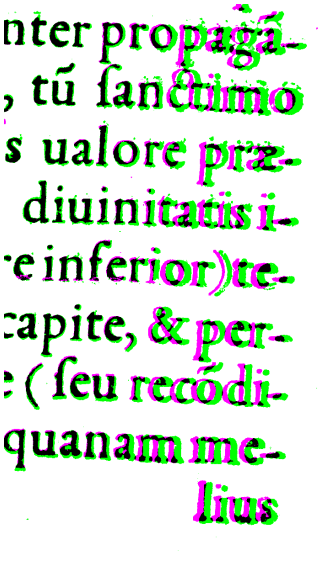
(a) Whole Page



(b) Detail



(c) Whole Page



(d) Detail

Fig. 7. Examples of resulting images that do not fit perfectly in all regions. This is due to non-similarity transforms on one of the images, e.g. book curling.