Decapod: A flexible, low cost digitization solution for small and medium archives

Faisal Shafait[1], Michael Patrick Cutter[2], Joost van Beusekom[1], Syed Saqib Bukhari[2], Thomas M. Breuel[2]
[1]*Multimedia Analysis and Data Mining Competence Center*
German Research Center for Artificial Intelligence (DFKI)
Kaiserslautern, Germany
faisal.shafait@dfki.de, joost.van_beusekom@dfki.de
[2]*Image Understanding and Pattern Recognition Group*
Department of Computer Science
Univ. of Kaiserslautern, Germany
bukhari@iupr.com, cutter@iupr.com, tmb@cs.uni-kl.de

***Abstract*—Scholarly content needs to be online, and for much mass produced content, that migration has already happened. Unfortunately, the online presence of scholarly content is much more sporadic for long tail material such as small journals, original source materials in the humanities and social sciences, non-journal periodicals, and more. A large barrier to this content being available is the cost and complexity of setting up a digitization project for small and scattered collections coupled with a lack of revenue opportunities to recoup those costs. Collections with limited audiences and hence limited revenue opportunities are nonetheless often of considerable scholarly importance within their domains. The expense and difficulty of digitization presents a significant obstacle to making such paper archives available online. To address this problem, the Decapod project aims at providing a solution that is primarily suitable for small to medium paper archives with material that is rare or unique and is of sufficient interest that it warrants being made more widely available. This paper gives an overview of the project and presents its current status.**

***Keywords*-document capture, digitization, scanning, low cost book scanning**

I. Introduction

Document digitization is not easy. The whole process, from initial image capture to a useful output, is arcane with no guarantee of usable results. Though there has been an immense amount of high quality research in the document engineering field over the past two decades in both academia and industry, little of it has made it into real, deployed systems. Even after capture, in most cases the technology needed to convert the material is expensive and requires expert users to configure it, and to develop workflows to deal with the exceptions that inevitably occur. Existing digitization solutions are well suited for large digitization projects like [1], where expensive equipment and training personnel is economically feasible. However, for small projects these costs present a significant barrier.

To assemble a solution an institution must procure and assemble equipment, train operators, procure several pieces of software, and develop exception handling and QA processes and tools. All of these require specialized skills and knowledge that is not readily available. It is beyond the scope of the average institution, and it is expensive. The Decapod project targets just these institutions or collections, ones with modest budgets, with material that is unique or fragile and must remain on-site, either because it is being used locally or there are restrictions on it being removed. Such institutions do not have sufficient material to justify the high set up costs of the overseas solution despite the low unit costs. A capture process is needed that is fast, able to deal gently with diverse materials and resilient to operator error, paper quality, lighting variations and other factors.

Much of the scholarly material that would benefit from Decapod is complex in layout. Journals, with their multi-column layout, illustrations and complex lists and tables, auction catalogs, inventories and records, newspapers and news-sheets, manuscripts and so forth contain images, multiple columns or boxes. Moreover, many of these documents are old, fragile, discolored, and in archaic typefaces. If the material is bound then even flat-bed scanning produces distorted images. Off-the-shelf packages such as the OCR packages are not particularly good in dealing with complex layouts and historical documents. The correction process is particularly tedious. This is unlikely to change as the market for OCR is not large, and the investment of the surviving commercial companies such as Abbyy, Nuance (Scansoft) is more oriented towards the commercially more important goal of extending the languages covered than addressing the more esoteric layouts. (It should be noted that they are doing an excellent job of addressing the breadth of languages, where inexpensive software packages can OCR around 200 languages).

Decapod is focused on delivering an affordable and cost effective solution to permit high quality, minimal user intervention solutions to the capture and preparation of small to medium collections. We apply the technology advances (both hardware and software) of the recent decades to remove the usability, cost and quality barriers to such projects. This is now possible thanks to the existence of well understood software and algorithmic approaches to the digitization problem and the emergence of affordable high

Figure 1. A prototype scanning rig, consisting of standard tripod hardware and consumer digital cameras. The rig is portable and can be operated anywhere using a laptop computer.

resolution digital cameras. The project will deliver an out-of-the-box solution that allows local staff with modest training to easily capture their material and convert it to archive quality content suitable for deposition in online archives. The solution will deal with bound material that must be treated gently (and also, of course, single sheet material), and will trim the image down to the page boundaries and remove discolorations and other visual defects so as to deliver page images comparable to those from a flat bed scanner. Our proposed solution will accomplish the following:

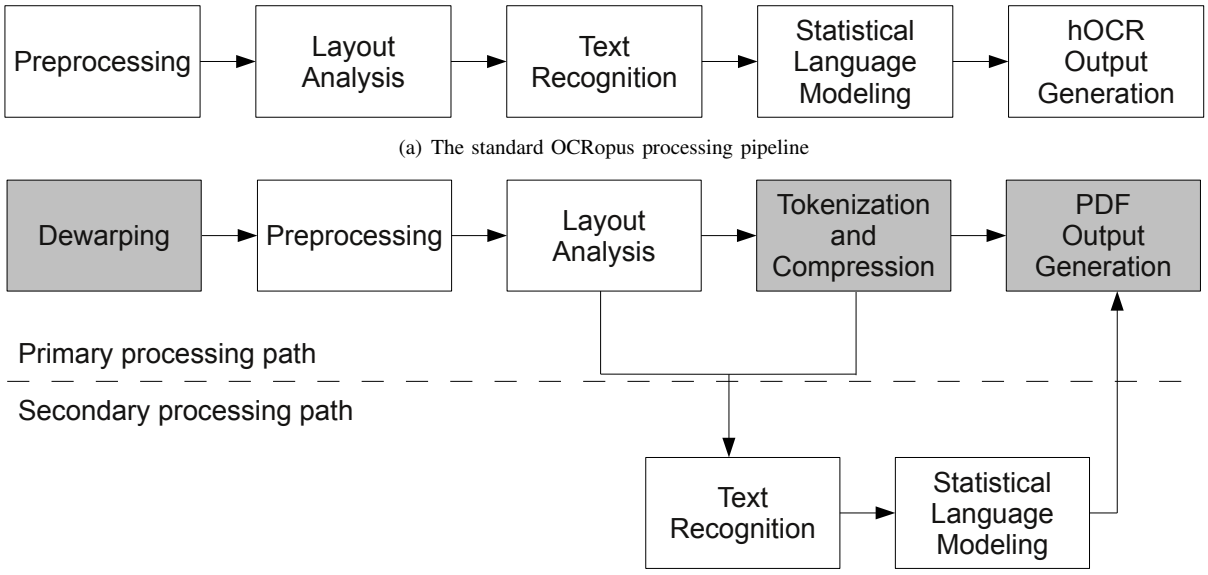
- Non-Destructive Scanning: The system will allow the non-destructive scanning of documents, journals, and bound volumes.
- Low Cost: Open source software, standard laptops, consumer-grade digital cameras.
- Competitive Quality: When used with a high-end digital camera and good lighting, the system will be capable of generating images of quality at least as good as that obtained by Google's scanning process.
- Portability: All system hardware components (cameras, tripods, laptop, etc.) will fit into a small suitcase.
- Usability by Non-Experts: The system will require minimal operator training and be usable by non-experts such as local staff and volunteers.
- Real-Time Scan Quality Control: Re-scans can be expensive or impossible; real-time scan quality control catches a high fraction of capture errors while the operator still has access to the document.

II. System Architecture

Decapod will deliver a complete solution for the capture of materials for which current digitization workflows are not appropriate. The deliverables will include software and suggested hardware configurations and hence allow the assembly of a complete system using off-the-shelf hardware components. A prototype of the system hardware is shown in Figure 1. The software components of the proposed system are:

- camera-based document capture using advanced computer vision algorithms to create "Scanner Equivalent" page images.
- A deeply user centered and easy-to-use document capture and quality control system based on state of the art document understanding technology that removes the need for most user interaction and simplifies the interaction when it is necessary.
- A high-quality scan-to-PDF conversion software that emits PDF/A with high fidelity (to the original) typefaces and embedded document layout information to permit reflow and text to speech.
- Integration of all software components into an end-to-end solution.

The overall flow of the system is a series of three steps. First there is the capture process, i.e. the creation of images of the pages from the physical material. The software demands at this point are primarily to ensure that the material is captured in its entirety and to sufficient quality. The next stage, which could take place later, is the generation of archive quality images and document structure information. The final stage is the generation of a usable output, which in this project is reflowable PDF/A documents. Several components from the OCRopus open source OCR system [2] will be employed in the system to achieve the final output. The relationship between the OCRopus system and the additional modules being developed as part of the Decapod project is shown in Figure 2. Different modules of the Decapod system are now described in more detail. Note that all of these modules have been / are being developed



(a) The standard OCRopus processing pipeline

(b) The Decapod book scanning pipeline (components being developed as part of the Decapod project are shown in gray)

Figure 2. The relationship between the OCRopus system and the additional modules being developed as part of the Decapod project.

as part of the Decapod project.

A. Document Capture

Documents are captured using a standard off-the-shelf consumer camera. The camera is connected to a USB port of the PC and `gphoto` library[1] is used to view live (low-resolution) video stream from the camera and to trigger image capture programmatically. We tested cameras from different brands for their support and stability with `gphoto` library. In our experience, Cannon cameras proved to be most stable (in terms of software crashes). Hence, we picked Cannon Powershot G10 due to its high resolution and relatively low cost. To increase throughput, we have also integrated support for a USB foot pedal to trigger image capture so that the user only has to turn pages with his hands while digitizing a book.

B. Dewarping

Once book pages are captured, they need to be dewarped for a better visual impression [3]. Dewarping module of the project is at a starting stage now. We plan to investigate approaches for monocular dewarping [4], stereo dewarping [5], [6] as well as dewarping using structured light [7] for their ease of setup, robustness, stability and output image quality.

C. Preprocessing / Layout Analysis

The flattened book pages returned by the dewaping module can be processed by typical modules for scanned pages like border noise removal [8], [9], skew correction [10], text/non-text segmentation [11], and layout analysis [12]. The text/non-text segmentation module is particularly more important in Decapod since we need to determine which connected components belong to text for font reconstruction. We have developed a multi-resolution morphology based method [13] within the Decapod project. Its main advantage over our previously published method [11] is that it does not require block segmentation prior to text/non-text classification. It is an extension of Bloomberg's text/image segmentation algorithm [14] that is specifically designed for text and halftone image separation. Bloomberg's method is simple and fast and performs well on text and halftone image segmentation, but it is unable to segment text and non-text components other than halftones, such as drawings, graphs, maps, etc. In our work, we introduced modifications to the original Bloomberg's algorithm for making it a general text and non-text image segmentation approach, where non-text components can be halftones, drawings, maps, or graphs.

D. Tokenization

Input to the tokenization algorithm is a text-only image in which non-text components have already been removed. The goal of tokenization is to cluster all the characters in a document into clusters containing the same character in the same font. These clusters are then called tokens. The tokenization is described in [15] in detail. A short summary will be presented here.

As we are interested in obtaining clusters with the same characters only, the label of each character obtained by OCR is used to cluster only characters together having the same label.

Inside this cluster it has then to be distinguished between different fonts. This is done by clustering characters that are

[1] http://www.gphoto.org/

visually similar into same clusters. As most font differences will show in the outline of the character - the main shape being the same for the most common fonts - the dissimilarity measure used for clustering will focus on the comparison of the outline of two characters.

First, the two characters are aligned to overlap their centroids. Second, the outline maps of both characters in a cluster are computed using morphological operations. Last, the dissimilarity is computed using the following formulas:

$$\text{error} = \sum_{x=0}^{W}\sum_{y=0}^{H} M(x,y) * \|T(x,y) - I(x,y)\| \quad (1)$$

$$\text{error}_I = \sum_{x=0}^{W}\sum_{y=0}^{H} M(x,y) * I(x,y) \quad (2)$$

$$\text{error}_T = \sum_{x=0}^{W}\sum_{y=0}^{H} M(x,y) * T(x,y) \quad (3)$$

$$\text{final error} = min(\frac{\text{error}}{\text{error}_I}, \frac{\text{error}}{\text{error}_T}) \quad (4)$$

where I is the candidate image, T is the token, and M is the mask. H and W represent the height and width of the token respectively. The mask M is obtained by morphological operations on the image of the character. First the image is dilated and inverted. Then this image is subtracted from a binary eroded image. The result of this last operation is combined with a thinned version of the image by a binary OR operation, finally arriving at the mask M.

If the overall error is lower than a given threshold, the new image I is clustered into the same cluster as T. In the other case, a new cluster is generated. Note that, due to the edge sensitive shape similarity metric, it is unlikely that different letters will be merged together.

The pairwise computation of the dissimilarity measure is computationally expensive. To reduce the amount of comparisons, a preliminary, inexpensive clustering is done based on the following features: height, width, and the number of holes present in the image of the character. Only characters where all of these features are identical, are compared to each other in the clustering.

E. Font Reconstruction

Font reconstruction is the inference of a mathematical representation of a digital font, given how it is typeset in a document image, which can then be used to reproduce the font in copies of the document or in new documents. The goal of the font reconstruction model is to capture all necessary characteristics of the original font in order to reproduce the original document in a visually faithful way.

The OCR system, OCRopus, outputs segmented and labeled letters, which become the input to our font reconstruction algorithm. After the token clustering phase the document can now be represented as a sequence of token IDs delimited by spaces. The co-occurrence of these tokens in words is the feature used to infer the candidate font groups. This is based on the assumption that a single latin word is almost always written in the same font. Therefore, for example if a token representing the letter 'a' co-occurs in multiple words then we assume those words were written in the same font. The candidate font selection method is further discussed in [15].

The next phase is to classify the font class of every letter within the document. This is achieved by exploiting locality and shape similarity to the candidate font alphabets. The probability of a token being classified as a particular font is determined by its spatial proximity of tokens which make up a respective font. The influence of font assignment of tokens decays until a maximum distance where then a simple nearest neighbor classifier is used for all remaining token font assignment. Details of the method can be found in [16]. Through our evaluation, we showed that this method is reasonably accurate on multi-font documents scanned at 300 DPI.

Once is a font group is identified, further processing is required in order to capture and output the font in reconstructed documents. We begin with the font group's alphabet of token prototypes and trace each of these prototypes by using `potrace`[2] - an open source polygon approximation tracing algorithm [17]. Since we want an accurate representation of the font, it is critical that the input to our tracing algorithm be at the maximum possible resolution. Since high resolution is already a requirement for high quality OCR this condition is met.

The input to `potrace` is the merged token prototype image (bitmap image) and the output in the vector outline (Bézier curves). `Potrace` approximates the bitmap using polygons to trace the outline of the bitmap. Peter Silinger describes the operation of `potrace` in four steps. In the first step the bitmap is decomposed into a sequence of paths between black and white areas in the image. In the second step the paths from the previous step are approximated by an optimal polygon. In the third step the polygons are transformed into an outline. The final step joins the outlines of each polygon to form Bézier curves representing the image.

Once a vectorized representation is available, the next step is to add parameters to each character in the font that control how it is rendered. This includes:

- **Relative size**: The size of each letter is a function of the aggregated statistics of the bounding box sizes of each instance of the letter classified as belonging to the same font group.
- **Baseline ratio**: Currently the system approximates how much of the letter should be placed above and below

[2]http://potrace.sourceforge.net/

the baseline by analyzing the baseline ratio of the same letter in another established font. This aspect of the system can be improved by finding the most similar corresponding font.

- **Left-right padding**: The amount of space between a letter and any other generic symbol can be inferred from either a corresponding font or extracted directly from the original document.
- **Kerning**: Kerning is the specific amount of space specified in some fonts between particular pairs of letters (like 'V' and 'A') to give a more visually pleasing effect. The Decapod system currently does not add specific kerning pairs. This could be added by measuring the spacing between pairs of letters within the document.

Often a document does not contain every letter of the alphabet in every font. Therefore, a distinction is to be made between reconstructing a font that can be used to recreate a specific document and reconstructing a portable font that can be used to author new documents in a similar form to the original. Ultimately, Decapod should achieve the former. To achieve the latter goal, one needs the ability to detect the most similar fonts to the newly reconstructed font in order to complete the alphabet of the font.

If there are more letters of the same font in a document, the corresponding tokens of a large number of these letters will be merged together during tokenization. Hence, result of tracing will be more visually appealing. Besides, the compression ratio will also increase with document size.

The font reconstruction algorithm is robust against possible outliers because letters that occur frequently are the ones that are the most likely to become part of the reconstructed font. After the initial candidate font selection phase all other tokens are classified as instances of one of the present fonts. These instances of a candidate font do not effect the shape of the font and are purely used as labels when reconstructing the document with reconstructed fonts.

F. PDF Generation

The PDF generation step converts the dewarped document images into different types of PDF. A representation of the processing pipeline for PDF generation is shown in Figure 3. Depending on the type of PDF that is wanted as output, different steps of the pipeline are run.

- Image only PDF: in this case, the dewarped images are converted into a single PDF. As no textual information is available this form is not searchable. However, this format needs no additional information and can thus even be used if no OCR and no font information is available. As input only the dewarped images are needed.
- Image with overlaid transparent text PDF: in this format, the recognized text will be overlaid transparently on the dewarped image, making the PDF searchable while maintaining the documents original appearance. As input the dewarped images and the character bounding boxes together with their label is needed.
- Tokenized PDF: instead of saving the page as a whole, tokenization is done on the connected components and only one character image per cluster is saved. This results in a searchable, lossy compressed version of the original input image. For this process OCR output and tokenization is needed. This type of PDF is merely an intermediate format and is not meant to be in the final system. Again, the dewarped images together with the OCR information serve as input.
- Font Reconstructed PDF: in this format, the extracted tokens are not saved as image inside the PDF, but they build the basis for generating a font out of the tokens. The generated fonts are then embedded in the PDF document.

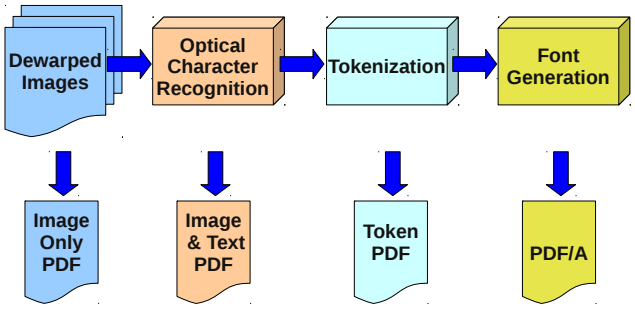


Figure 3. Overview of the PDF generation steps. Dewarped images can easily be converted into image only PDFs, whereas font reconstructed PDFs need all information of all intermediate processing steps.

The PDF generation uses the `ReportLab Toolkit` for PDF generation[3] and OCRopus [2] for performing OCR to create the underlying text layer. All data is stored in the *book structure*, a set of files and directories for each set of documents, as proposed by the OCRopus system.

III. Project Status

The Decapod software is available online[4]. The current version implements the full processing pipeline: starting with capturing of documents, loading and saving started projects, reordering the pages and export as PDF.

However, some important modules are still missing: dewarping is not yet available, instead only dewarped (e.g. flat bed scanned) images can be processed. The PDF generation has limited functionality. Currently, image-only PDF and image with overlaid transparent text layer PDF can be generated.

It is difficult to assess the overall performance of the Decapod system at this stage. In our view, the dewarping

[3]http://www.reportlab.com/
[4]http://code.google.com/p/decapod/

module will be the most crucial for delivering an overall good quality output. For tokenized and font reconstructed PDFs, it would be important to keep the token clustering errors low. We also plan to develop an automatic validation algorithm to verify the quality of the font reconstructed PDF. In this way, it will be possible to warn the user of a bad output quality or to automatically fall back to the image with overlaid transparent text PDF.

IV. Conclusion

The Decapod project investigates the use of low cost consumer electronics hardware (e.g. a standard tripod stand, consumer digitial cameras, and a PC) for setting up a digitization project at a small scale. Besides, software components are being developed to produce searchable PDFs that are visually similar to the original documents by having the same (reconstructed) fonts and layout. Our solution is open-source, easy to use, and will provide an out-of-the box method for in-situ digitization of small to medium archives where setting up a full production system such as that used by JSTOR or Google is not feasible.

References

[1] L. Vincent, "Google book search: Document understanding on a massive scale," in *Int. Conf. on Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007, pp. 819–823.

[2] T. M. Breuel, "The OCRopus open source OCR system," in *Proc. SPIE Document Recognition and Retrieval XV*, San Jose, CA, USA, Jan. 2008, pp. 0F1–0F15.

[3] F. Shafait and T. M. Breuel, "Document image dewarping contest," in *Proc. Int. Workshop on Camera-Based Document Analysis and Recognition*, Curitiba, Brazil, Sep. 2007, pp. 181–188.

[4] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Dewarping of document images using coupled-snakes," in *Proc. Int. Workshop on Camera-Based Document Analysis and Recognition*, Barcelona, Spain, Jul. 2009, pp. 34–41.

[5] A. Ulges, C. Lampert, and T. M. Breuel, "Document capture using stereo vision," in *Proc. ACM Symposium on Document Engineering*. ACM, 2004, pp. 198–200.

[6] A. Yamashita, A. Kawarago, T. Kaneko, and K. Miura, "Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system," in *Proc. Int. Conf. on Pattern Recognition*, 2004, pp. 482–485.

[7] M. Brown and W. Seales, "Document restoration using 3d shape: A general deskewing algorithm for arbitrarily warped documents," in *Proc. Int. Conf. on Computer Vision*, July 2001, pp. 367–374.

[8] F. Shafait and T. M. Breuel, "The effect of border noise on the performance of projection based page segmentation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 846–851, 2011.

[9] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel, "Document cleanup using page frame detection," *International Journal on Document Analysis and Recognition*, vol. 11, no. 2, pp. 81–96, 2008.

[10] J. van Beusekom, F. Shafait, and T. M. Breuel, "Combined orientation and skew detection using geometric text-line modeling," *International Journal on Document Analysis and Recognition*, vol. 13, no. 2, pp. 79–92, 2010.

[11] D. Keysers, F. Shafait, and T. M. Breuel, "Document image zone classification - a simple high-performance approach," in *2nd International Conference on Computer Vision Theory and Applications*, Barcelona, Spain, Mar. 2007, pp. 44–51.

[12] F. Shafait, D. Keysers, and T. M. Breuel, "Performance evaluation and benchmarking of six page segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 941–954, 2008.

[13] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Improved document image segmentation algorithm using multiresolution morphology," in *SPIE Document Recognition and Retrieval XVIII*, San Francisco, USA, Jan. 2011.

[14] D. S. Bloomberg, "Multiresolution morphological approach to document image analysis," in *Proc. Int. Conf. on Document Analysis and Recognition*, St. Malo, France, 1991, pp. 963–971.

[15] M. P. Cutter, J. van Beusekom, F. Shafait, and T. M. Breuel, "Unsupervised font reconstruction based on token co-occurrence," in *10th ACM Symposium on Document Engineering*, Manchester, UK, Sep. 2010.

[16] ——, "Font group identification using reconstructed fonts," in *SPIE Document Recognition and Retrieval XVIII*, San Francisco, USA, Jan. 2011.

[17] P. Selinger, "Potrace: a polygon-based tracing algorithm," in *http://potrace.sourceforge.net*, 2003.