Script-Agnostic Reflow of Text in Document Images

Saurabh Panjwani
Bell Labs India
saurabh.panjwani@gmail.com

Abhinav Uppal
IIT Delhi
abhinavuppal88@gmail.com

Edward Cutrell
Microsoft Research India
cutrell@microsoft.com

ABSTRACT
Reading text from document images can be difficult on mobile devices due to the limited screen width available on them. While there exist solutions for reflowing Latin-script texts on such devices, these solutions do not work well for images of other scripts or combinations of scripts, since they rely on script-specific characteristics or OCR. We present a technique that reflows text in document images in a manner that is agnostic to the script used to compose them. Our technique achieved over 95% segmentation accuracy for a corpus of 139 images containing text in 4 genetically-distant languages—English, Hindi, Kannada and Arabic. A preliminary user study with a prototype implementation of the technique provided evidence of some of its usability benefits.

Author Keywords
Document images, segmentation, reflow, script-agnostic.

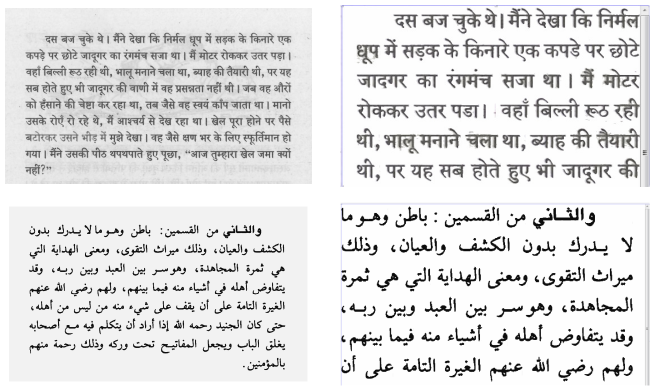
ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces (interaction styles, prototyping). I7.5. Document and Text Processing: Document Capture.

INTRODUCTION

In several practical scenarios, electronic documents can be viewed only as images and access to the underlying raw text is either difficult or infeasible. One example of such a scenario is of reading text from digital libraries like Google Books. Another example is that of collective viewing of a digital presentation made using a document camera [12].

One issue that can hinder readability in such scenarios is the screen width available to the user. A researcher reading scanned pages from Google Books on a phone or an eBook reader may find it frustrating if he is forced to scroll back and forth for every line of text. Students in a large class reading text from a projector screen may have a hard time if the displayed text lines do not fit the screen or are otherwise made too narrow to maintain legibility. A suitable user interface for such scenarios would be one which allows text to be enlarged and shrunk while also being reflowed into a fixed screen width. (See figure 1.)

Although the problem of reflowing text in document images has been studied in the past [1], such work has largely focused on documents containing English text. The resulting tools are inapplicable to non-English documents and those written in non-Latin scripts, in particular. There are several ongoing efforts today on digitizing books in Asian and other non-Western languages [11,3] and simultaneously, the number and variety of electronic reading devices available to users is on the rise worldwide. Enabling universal and convenient access to cross-language digital libraries raises the need for reflow-capable readers that operate on different scripts; indeed, such a need has been expressed by some mobile phone users already[1].



दस बज चुके थे। मैंने देखा कि निर्मल धूप में सड़क के किनारे एक कपड़े पर छोटे जादूगर का रंगमंच सजा था। मैं मोटर रोककर उतर पड़ा। वहाँ बिल्ली रूठ रही थी, भालू मनाने चला था, ब्याह की तैयारी थी, पर यह सब होते हुए भी जादूगर की वाणी में वह प्रसन्नता नहीं थी। जब वह औरों को हँसाने की चेष्टा कर रहा था, तब जैसे वह स्वयं काँप जाता था। मानो उसके रोएँ रो रहे थे, मैं आश्चर्य से देख रहा था। खेल पूरा होने पर पैसे बटोरकर उसने भीड़ में मुझे देखा। वह जैसे क्षण भर के लिए स्फूर्तिमान हो गया। मैंने उसकी पीठ थपथपाते हुए पूछा, "आज तुम्हारा खेल जमा क्यों नहीं?"

दस बज चुके थे। मैंने देखा कि निर्मल
धूप में सड़क के किनारे एक कपडे पर छोटे
जादगर का रंगमंच सजा था। मैं मोटर
रोककर उतर पडा। वहाँ बिल्ली रूठ रही
थी, भालू मनाने चला था, ब्याह की तैयारी
थी, पर यह सब होते हुए भी जादूगर की

والثـاني من القسمين: باطن وهـو ما لا يـدرك بدون الكشف والعيان، وذلك ميراث التقوى، ومعنى الهداية التي هي ثمرة المجاهدة، وهوسـر بين العبد وبين ربـه، وقد يتفاوض أهله في أشياء منه فيما بينهم، ولهم رضي الله عنهم الغيرة التامة على أن يقف على شيء منه من ليس من أهله، حتى كان الجنيد رحمه الله إذا أراد أن يتكلم فيه مـع أصحابه يغلق الباب ويجعل المفاتيـح تحت وركه وذلك رحمة منهم بالمؤمنين.

والثـاني من القسمين : باطن وهـو ما
لا يـدرك بدون الكشف والعيان، وذلك
ميراث التقوى، ومعنى الهداية التي هي ثمرة
المجاهدة، وهوسـر بين العبد وبين ربـه،
وقد يتفاوض أهله في أشياء منه فيما بينهم،
ولهم رضي الله عنهم الغيرة التامة على أن

Figure 1. Document images in Hindi (top-left) and Arabic (bottom-left) and their corresponding reflowed versions. Hindi is written from left to right, Arabic from right to left.

In this paper, we present a technique to reflow text stored in document images in a manner that is independent of the script used to create them. Our technique is simple and at its heart, it involves segmenting blocks of text in a document image into individual words without implementing any character recognition. While segmenting text, we address the possible presence of diacritics and tolerate a moderate amount of variability in inter- and intra-word spacing, particularly capturing cases of zero intra-word spaces, a characteristic of some Asian scripts. Addressing such possibilities raises interesting challenges unmet by previous reflow techniques.

MobileHCI 2011, Aug 30–Sept 2, 2011, Stockholm, Sweden.
Copyright 2011 ACM 978-1-4503-0541-9/11/08-09....$10.00.

[1] See this online discussion forum, for example: *http://www.mobileread.com/forums/showthread.php?t=61633, 2009.*

Our technique works for scripts which use white spaces to delimit words and lines; this includes scripts of most major language families including, in particular, the Indo-European, Semitic and Dravidian families. As a proof of concept, we have tested a prototype implementation on text from 4 languages: English (a West-Germanic language), Hindi (an Indo-Aryan language and the most-widely spoken one in India), Kannada (a Dravidian language from South India) and Arabic (a Semitic language spoken in West Asia and Africa). Results from our initial experiments are promising: for a sample of 139 document images, our technique achieved over 99% accuracy for English and Hindi and over 95% for the other two languages.

While these accuracy rates could potentially be improved by tweaking our technique for individual scripts, we believe there is value in persisting with a script-agnostic approach. First, it results in a simpler user interface – a reflow-enabled reader that requires no parameter input from the user (not even the script name!). The second and more important reason is that it addresses the case of multi-lingual text. The use of English words in non-Western text is prevalent practice and even in our sample corpus, we find several instances of English words immersed in largely non-English text. We also find cases of English documents containing non-Latin script (e.g., language-learning books) and of Indian documents that use multiple Indian scripts. Text reflow for such documents must necessarily be agnostic to the underlying script.

RELATED WORK

The issue of reflowing text for facilitating reading on small screens has arisen multiple times in the literature [6, 7]. For the case of reading Western-language text on small screens, it is well established that viewing automatically reflowed text (in document images or otherwise) is preferred by users over navigating horizontal pan/scroll interfaces [7]. In fact, text reading studies for mobile devices often use reflowed text as the baseline presentation format [9]. What is missing in the literature is a tool that implements reflow for *document images* and does this *cross-linguistically*. This is the problem we address in the current paper.

For English, there are reflow-capable document readers already available in the market [10] but most of the commercial tools rely on OCR for implementing reflow. However, good OCR tools are not yet available for Asian languages like Hindi and Kannada and there is some evidence (from users of web forums) that reflow technologies for English do not work with such languages. In the academic literature, there is some work on OCR-less reflow but largely, for English only [1]. Some assumptions made by [1], like the presence of inter-character spacing and the absence of diacritics, are violated by non-Latin scripts. To the best of our knowledge, ours is the first work which attempts text reflow for Indic, Arabic and Latin scripts using a single algorithm.

There is much literature on document image segmentation for both Latin and non-Latin scripts. Surprisingly, though, there is little work on techniques to segment document images in a script-agnostic manner. Ittner and Baird build a language-free document image analysis tool [5] but they rule out Hindi and Arabic because of the way these languages treat intra-word spaces. Accuracy of their tool is not known for any script; in fact, there is subsequent work [2] which shows that their technique (and of others, e.g., [8]) performs poorly on Indic scripts. There has been some work on script-independent *line* segmentation of *hand-written* text [4] but accuracy measures are again not known, and applicability to Indic/Arabic text is unclear. Besides, line segmentation, in general, is insufficient for text reflow.

THE ALGORITHM

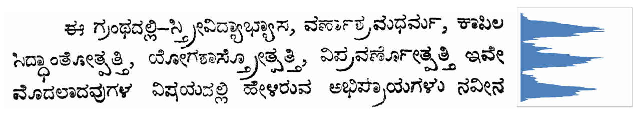
Our technique works well for a class of scripts which we refer to as *segmentable* scripts. Such scripts have distinct white spaces separating adjacent lines and adjacent words and if they admit spaces within words then the width of every such space is strictly less than that of every inter-word space. For simplicity, we assume that text is written in straight horizontal lines and read from top to bottom, although it seems possible to remove this assumption by automatically learning line orientation. We rule out logographic scripts like Chinese which do not delimit words by spaces; such scripts require domain knowledge for segmentation. (Our definition of segmentable scripts is similar to that of Manhattan scripts/layouts [5], except that (a) we permit zero intra-word spacing and the use of dangling diacritics; (b) we do not consider vertical orientations.)

The technique operates on high-resolution document images such as those obtained by digitally scanning paper documents (300+ dpi). In the current version, we make a few other assumptions about the input image: it contains only text, the text has nearly-consistent font size and inter-line and inter-word spacing, the image is binarized (has only black and white pixels) and there is no skew or shear. There are known script-agnostic approaches to address some of these issues [5], although some others (e.g., use of variable font size and spacing) are hard to address even for single scripts.

There are two main components of our algorithm: a *segmentation engine* which identifies words within the text and a *reflow process* which reorganizes the identified words to fit a particular line-width. For segmentation, we use a top-down approach based on projection profiles: first, compute the black-pixel density for each row of pixels in the image and identify rows with zero density; these define the line boundaries. Then, for each identified line (region between 2 line boundaries), compute the black-pixel density for each column of pixels and identify "large" groups of consecutive columns with zero density; these define word boundaries within that line. Implementing this approach accurately and script-agnostically, however, raises some non-trivial issues.

Line Identification Issues

First, black pixels in the space between lines can cause pairs of lines to be treated as one line. This could either be because of noise but, more importantly, due to the use of diacritics which extend downwards or upwards from a text line. Figure 2 shows an extract of Kannada text from our corpus with heavy use of such diacritics; the horizontal projection of this extract has no rows with zero black-pixel density. It is for this reason that segmentation algorithms for Kannada (and other Dravidian languages) tend not to use projection profiles but instead adopt a bottom-up segmentation approach [2]. Bottom-up approaches, though, are difficult to apply cross-linguistically, so we stick with the projection profiles approach but modify it to search for *minimas* in row pixel densities, instead of absolute zeroes. This introduces a slight risk of line over-segmentation but we counter the risk using suitable thresholds to separate inter-line minimas from intra-line ones. (A multiplicative threshold of 0.1 sufficed in our tests.)



ಈ ಗ್ರಂಥದಲ್ಲಿ—ಸ್ತ್ರೀವಿದ್ಯಾಭ್ಯಾಸ, ವರ್ಣಾಶ್ರಮಧರ್ಮ, ಕಾಪಿಲ
ಸಿದ್ಧಾಂತೋತ್ಪತ್ತಿ, ಯೋಗಶಾಸ್ತ್ರೋತ್ಪತ್ತಿ, ವಿಪ್ರವರ್ಣೋತ್ಪತ್ತಿ ಇವೇ
ಮೊದಲಾದವುಗಳ ವಿಷಯದಲ್ಲಿ ಹೇಳಿರುವ ಅಭಿಪ್ರಾಯಗಳು ನವೀನ

Figure 2. A splice of a Kannada document image with its horizontal projection profile. There are no regions of zero pixel density in the profile but there are several minimas.

Sometimes, diacritics could be completely detached from associated words and could get treated as separate lines by our segmentor. We detect such cases by relying on two characteristics of diacritic-only lines: they are much shorter than true text lines and contain regions of contiguous white columns. Using appropriate thresholds, we detect these lines and merge them with nearby text lines.

Word Identification Issues

Once text lines have been identified, the key issue is to identify inter-word spaces in each line. Doing this script-agnostically is difficult since some scripts admit intra-word spaces while others (e.g., Hindi) don't. For the former scripts, the distribution of white spaces in each line is bimodal; for the latter, it is unimodal. Our goal is to address both possibilities *simultaneously* and we do it as follows:

1. *Let w(i) be the width of the i^{th} narrowest white space in the line.*
2. *Identify the smallest i, say i^*, for which the ratio w(i)/w(i-1) exceeds a threshold t and for which the slope of function w(.) is locally maximized at i.*
3. *Discard all spaces of width smaller than $w(i^*)$.*

In our experiments, $t = 1.4$ emerged as an effective choice. Later, we also check that the i^* selected in the above manner is the closest match (in terms of width) to the i^* of other lines and adjust its value if this is not the case; this helps address corner cases like lines with single words and those with scanty inter-word spacing. In the end, we use a contour-walking technique to expand word boundaries where needed; this reduces line boundary errors (e.g., diacritic spillovers).

The Reflow Process

Once words have been identified in a document image, reflowing them to fit a given width is relatively easy. We adapt the dynamic-programming approach used by LaTex [13] to make it work on document images. Three tricky issues arise when doing this: (1) *word alignment* – when a word is moved from one line into the preceding or succeeding line, it needs to be aligned with the words in that line. For this, we identify the highest pixel-density row for every text line – call it the "center" of that line and of each contained word – and for positioning a word in a fresh line we align the word's center with the line's center; (2) *flow direction* – some scripts are written left to right, some not. Flow direction can be learnt with reasonable accuracy by locating forced line breaks (although in our current prototype, we let the user specify it); (3) *hyphenation* – in some scripts like English, it is common to break words at the end of a line using a hyphen. So, if a new word is to be placed at the end of a hyphen-ending line, it could *follow* the hyphen (treat it as part of the word) or it could *replace* it (treat it as a word-breaker). The choice seems hard to make without doing text recognition and so we treat hyphen-ending lines just as ordinary lines in our current system, hoping it will not hurt usability in practice.

ACCURACY

We tested the accuracy of our segmentation engine using a corpus of 139 document images containing printed text – English (41 images, 7 documents), Hindi (51 images, 6 documents), Kannada (40 images, 8 documents) and Arabic (7 images, 2 documents). The English, Hindi, and Kannada images were obtained from the Digital Library of India [3], the Arabic images through a web search. Documents were font-diverse within each language, including one case of type-written text. Twelve documents had bilingual text (7 English-Hindi, 1 Kannada-English, 4 Kannada-Sanskrit). On average, there were 221.9 words per document. All Arabic documents used contemporary scripting (figure 1); traditional Arabic scripts are harder to segment using a generic top-down technique like ours.

Language	***Line Errors***		***Word Errors***	
	Merge	*Split*	*Merge*	*Split*
English	*0*	*0*	*0.37*	*0.07*
Hindi	*0*	*0*	*0.73*	*0.09*
Kannada	*0.26*	*0*	*3.87*	*0.42*
Arabic	*0*	*0*	*3.74*	*0.10*

Table 1. Error rates for our prototype implementation (in %).

Table 1 depicts our accuracy results, which were computed by manual inspection of the engine's output. We encountered very few line identification errors: zero line splits and only a 0.26% rate of line mergers in the case of Kannada (none for other languages); these errors were largely due to dangling diacritics, which characterize Kannada. Word identification was less accurate. We noted up to a 3.9% rate of word mergers, which were either due to poor print quality or high variability in inter-word spacing (e.g., in Arabic, due to the cursive nature of text, some words tend to be closer to each other than normal, sometimes as close as characters). Word splits were still scarce: at most 0.42% per script. These were largely due to print erasures and sometimes due to stray deviations in intra-word spacing. The overall segmentation accuracy of our technique is close to that of the best script-specific segmentation algorithms: in Kannada, we achieve 95.4% accuracy and the best accuracy rate for Dravidian languages is 98.5% [2]. (We remark that [2] achieves this accuracy using a semi-automated technique which is applied to text drawn from a single document; our Kannada test set, on the other hand, was derived from 8 documents.) For Arabic, the best known accuracy rate is 97% [6] and our technique currently achieves about 96.1%.

USABILITY

To test usability, we ran a preliminary user study for a smartphone implementation of our technique. There were 2 goals of the study: (a) compare usability of our technique with that of reflow-lacking readers across different scripts, and (b) assess the impact of reflow errors on readability. Twelve bilingual subjects (6 male, 6 female, mean age: 29) participated in our study. All subjects were experienced phone users and 9 reported to have used smartphones in the past for reading long texts. Subjects were fluent in English and Hindi; fluency in English was consistently reported to be greater. During the study, subjects read 2 scanned texts in each of these languages (a within-subjects design) using a PDA with Windows Mobile 5.0. One text was read using the image viewer available for this platform and the other using our reflow-capable reader. (The former required both horizontal and vertical scrolling; the latter only vertical strokes.) For each language, the texts were extracted from a single source and were identical in word count. Text presentation was balanced against order and condition.

Subjects read aloud in all conditions and answered 4 multiple-choice questions on each text after reading it. We observed a significant difference in reading speed between the two conditions for both English ($t(11) = 10.9$, $p<0.001$) and Hindi ($t(11) = 4.9$, $p<0.001$). The difference was greater for English than for Hindi, plausibly due to the gap in the subjects' fluency in the languages. No significant differences in comprehension were observed, indicating that the increased speed of reading did not reduce attention.

All subjects stated a preference for the reflow-capable reader over the reflow-lacking one, and across languages. Some reported to have noted irregularities while reading the reflown texts (7 noted word split errors, 5 noted a reflown line-ending hyphen and 1 noted word misalignment, all correctly so), but none felt these issues were significant enough to induce a switch to the default viewer.

Language	*Condition*	***Reading speed***		***Comprehension***	
		Mean	*Std.dev*	*Mean*	*Std.dev*
English	No reflow	*85.29*	*10.94*	*75*	*26.11*
	Reflow	*148.33*	*26.06*	*77.08*	*12.87*
Hindi	No reflow	*83.67*	*26.54*	*72.91*	*19.82*
	Reflow	*103.27*	*30.32*	*72.91*	*32.78*

Table 2. Reading speed and comprehension data for default (reflow-lacking) image viewer and our reflow-enabled reader. Reading speed is in words/minute and comprehension in %.

CONCLUSION

With libraries across the world being digitized, there is a growing need for electronic reading software that can flexibly address variations in document structure and composition and, in particular, those in the underlying script. In this paper, we have addressed a key process in electronic document reading, namely automated reflow of text in document images. We demonstrated that it is possible to execute this process in a script-agnostic manner without relying on OCR. The technique developed by us works reasonably well for four scripts; future work will strengthen accuracy rates for these (and possibly, other) scripts and incorporate other important elements of layout analysis like text segregation and block identification. Once completed, the prototype will be made available publicly.

REFERENCES

1. Breuel, T. Reflowable document images for the Web. *In Proc. WDA 2003, the 2nd International Workshop on Web Document Analysis*, (2003).
2. Dasigi, P., Jain, R., and Jawahar, C.V. Document Image Segmentation as a Spectral Partitioning Problem. *In Proc. ICVGIP '08, 6th Indian Conference on Computer Vision, Graphics and Image Processing*, (2008).
3. Digital Library of India. *http://www.new.dli.ernet.in*.
4. Du, X., Pan, W., and Bui, T. Text line segmentation in handwritten documents using Mumford-Shah model. *ICFHR '08*, (2008), 253-258.
5. Ittner, D.J. and Baird, H. Language-Free Layout Analysis. *In Proc. ICDAR '93*, (1993), 336-340.
6. Lee, Y., Pepineni, K., Roukos, S., Emam, O., and Hassan, H. Language Model Based Arabic Word Segmentation. *ACL '03*, (2003), 399-406.
7. Muter, P. Interface Design and Optimization of Reading of Continuous Text. *Cognitive Aspects of Electronic Text Processing*, H. van Oostendorp and S. de Mul (Eds.) (1996).
8. Nagy, G., Seth, S., and Viswanathan, M. A Prototype Document Image Analysis System for Technical Journals. *Computer 25*, 1992, 10-22.
9. Öquist, G. and Lundin, K. Eye Movement Study of Reading Text on a Mobile Phone using Paging, Scrolling, Leading and RSVP. *In Proc. MUM '07, 6th International Conference on Mobile and Ubiquitous Multimedia*, (2007).
10. Repligo Reader. *http://www.cerience.com/products/reader*.
11. The Million Book Project. *http://www.ulib.org*.
12. The Visualiser Forum. *http://www.visualiserforum.org/*.
13. Word Wrap. *http://en.wikipedia.org/wiki/Word_wrap*.