Page Frame Segmentation for Contextual Advertising in Print on Demand Books

Hanning Zhou
Amazon Inc.
701 5th Ave., Seattle, WA, 98104
hanzhou@amazon.com

Zongyi Liu
Amazon Inc.
701 5th Ave., Seattle, WA, 98104
joeliu@amazon.com

Abstract

The Web, TV and print publishing are the top three advertisement media. With the technology breakthroughs in digital printing, scanning and data processing, print on demand emerges into a multi-billion dollar business that is due to revolutionize the traditional publishing industry.

With the rapid growth of print-on-demand, a whole new advertising media is born, that is, print on demand books. We discuss some unique characteristic of this new media that enables user-targeted contextual advertising, which is more effective than traditional print media.

In this paper, we first proposed a contextual-matching based advertising model for print on demand books, and then concentrate on using the page frame segmentation algorithm to solve the problem of where to insert the selected contextual ads in print on demand books. Our page segmentation algorithm shows significant improvement over existing methods in its robustness against noisy background.

1. Introduction

On Nov. 19, 2007, just a week after the book ”Print is Dead” [9] was published, Amazon announced the Kindle [1] e-book reader, which marks one of the turning points of the mass market book business. In the same year, industry tracker Bowker reports traditional publishing is flat and business titles declined 12 percent to 7,650. Meanwhile, the newspaper business are also facing unprecedented challenges from the digital media [4] [5].

Is print really dead? Yes and no. As the demand for mass-market publications becomes more volatile, the book vendors across the whole spectrum from traditional publishing houses to self-publishers, are turning to a disruptive printing technology called *print on demand* (POD). Infotrend forecasts the US print-on-demand market will approach $79 billion in 2012 [6]. Major book retailers are already fulfilling a significant portion of book purchase orders through POD. For instance, as of March 2009, over 1 million titles on Amazon are printed on demand. The POD revenue grows 68% YoY. Two of every 100 books sold on Amazon are printed on demand.

Compared to traditional offset printing, POD has the following advantages:

1. Fixed cost per copy, irrespective of the size of the order
2. Large selection remains available without physical inventory
3. No waste from unsold products

By integrating the above advantages with an online retail platform, Amazon extended POD beyond its niche market of self-publishing books, into the much broader mainstream market of the front-listing books. As the audience base of POD books grows, it’s eminent to start exploring the advertising sponsored business model in the new media of Print-on-Demand books.

To put the size of the potential advertising market for print on demand into perspective, we quote the following market outlook from the ICON group research reports. In 2009, the world wide publishing advertising market is estimated to be $328.4 billion [17], while that of the internet search ads market is estimated to be $23.5 billion [16], and that of the online display ads is $14.4 billion [14], and that of the online video ads is $32.9 billion [15]. As print on demand is becoming the next generation of print publishing media, it has enormous advertising potential.

Compared to the traditional publishing media, the POD books as an advertising media has the following unique characteristics:

1. The book is manufactured for a particular customer. Therefore it is possible to customize the ads in the book to target the specific book buyer.
2. Every book printed is guaranteed to have at least one reader, with known user profile. Therefore we can measure the volume of ads impressions generated, in each group of users.

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

3. Cost per 1000 impression is tied to printing cost, which is fixed, regardless of the scale of the campaign. Therefore it is easy for the advertisers to control the budget of their campaigns.

To fully take advantage of these characteristics of the POD media, we propose a user-targeted contextual advertising model, the details of which will be described in Section 2.

In Section 3, we describe a page frame segmentation algorithm to identify the margin of the book page using document image analysis. In Section 4, we compare the performance of our page frame segmentation algorithm against the state of the art page segmentation algorithms and show its robustness in noisy image. In Section 5, we show some example pages with contextual ads inserted to illustrate the final printed product.

The rest of this paper concentrates on solving the page layout analysis problem, which will enable us to identify where to insert the ads.

2. User-targeted Contextual Advertising in POD Books

Contextual Advertising has been a heated area of research in the field of online advertising [11][8]. We borrow ideas from the existing cosine distance based matching approach, and added user profile as additional features to make the contextual ads targeted to the intended audience. Below is the process of generating a POD book with user-targeted contextual advertisement.

As an offline preprocess, we extract the following features from each POD book: BISAC subject code of the book, keywords in the description of the book, user profile of previous buyers of the book, user profile of the readers of the same author of the book, available ads space in the book; and translate them to a feature vector to store in our book meta-data repository.

When the advertisers or their agents decide to launch a new campaign through POD ads, we ask them to specify keywords, target audience, display size and a high-resolution image of their ads, and store them in our ads repository.

When a customer orders a POD book, we find all the ads in the repository whose target audience matches the customer's user profile. We also find all the ads in the repository that is within a certain cosine distance from the meta-data feature vector of the POD book.

Finally, we insert the matched ads into the available spaces in the book and print out the customized copy of the book and ship it to the customer.

When inserting the ads, we need to find the available spaces in a book where we can insert ads. The most straightforward way is to insert the whole page ads at the very beginning or end of the books. A more elaborate way is to insert the whole page ads between chapters. The most sophisticated approach is to insert the ads at the margin of the book page where the matching keywords appear. This approach involves the following steps:

1. Run OCR over the entire book to extract text and the bounding boxes of the words.

2. Match the keywords of the ads with the text in each page to determine which page to insert the ads.

3. Resize the original ads image to fit the margin of the book.

It's important to accurately detect the margin of each page, because it will determine how much available space we can use for displaying ads. Section 3 describes our page frame segmentation algorithm.

3. Book Page Frame Segmentation

Our page frame segmentation algorithm consists of two parts: (i) an initial segmentation that binarizes the input images, searches text-lines and images inside the pages, and removes noises; and (ii) a page frame detector that groups text-lines and images using their vertical projection profile histogram, and dynamically estimates the page boundary.

3.1. Initial Segmentation

The initial segmentation starts with the image binarization. For color input images, we first convert it to grayscale, and then compute the histogram. We fit the histogram into a single peak, and classify pixels falling into the histogram peaks as background, and the rests as foreground. This is based on the observation that background pixels consist of the majority of the image.

After removing background, we deskewed the image using W. Postl's method [18] implemented in the Leptonica library [2]. Then we detect text-lines and images/halftones regions using the page segmentation algorithm implemented in the Leptonica library. In brief, it is a seed filling algorithm that first performs the morphological opening to find the seeds in an image, and then uses iterative closing to connect horizontally/vertically aligned components, and then searches text-lines and images using the seeds locations.

The segmentation result is shown in the top image of Fig. 1. We can see that most of the textlines are correctly detected. However, some text-lines are under-segmented, e.g., several lines near "Workstations" at the middle right of the page are mistakenly grouped into one text line. This is due to the error of the closing operations in the text-line mask

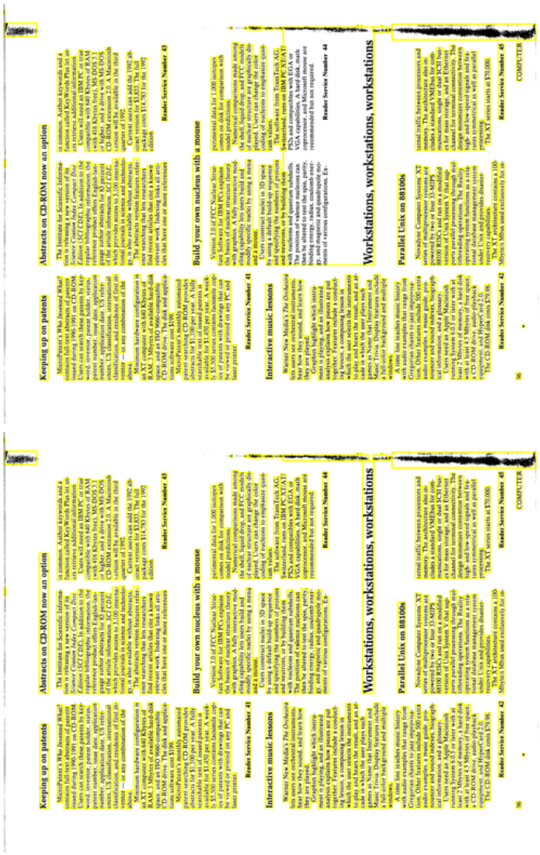
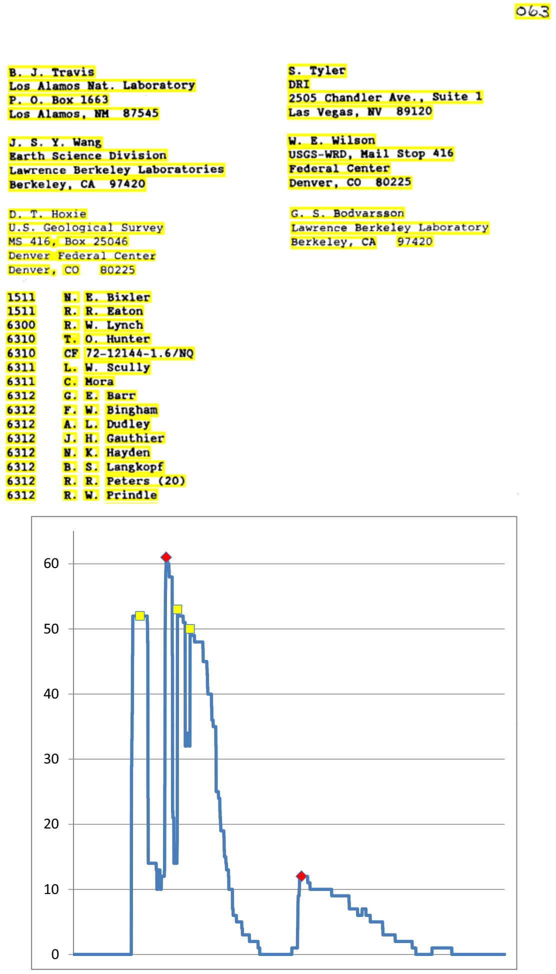


Figure 1. Two sample images of the initial segmentation results. The top one is without noise removal. The bottom one is with noise removal. The detected textlines are highlighted with yellow bounding boxes.

creation that connects these lines. To address this problem, we perform morphologically opening for every connected component in the text-line mask, which is able to split under-segmented text-lines parallel to each other. But this method does not work for the "T-shape" under-cut, i.e., a horizontal text-line connected to a vertical text-line. Usually this "T-shape" error can be detected by computing the horizontal/vertical foreground histogram for each component in the text-line mask image, and checking if there is a *sharp* peak. In our algorithm, a sharp peak is defined to be an area with width less than one third of the component width, and with top value at least four times greater than the average value of the non-peak areas. And A "T-shape" error can be fixed by cutting the mask component either horizontally or vertically, based on the peak locations. The bottom image of Fig 1 shows the initial segmentation result after the noise removal, where the problem of under-segmentating text-line in the top image is resolved.



063

B. J. Travis
Los Alamos Nat. Laboratory
P. O. Box 1663
Los Alamos, NM 87545

J. S. Y. Wang
Earth Science Division
Lawrence Berkeley Laboratories
Berkeley, CA 97420

D. T. Hoxie
U.S. Geological Survey
MS 416, Box 25046
Denver Federal Center
Denver, CO 80225

S. Tyler
DRI
2505 Chandler Ave., Suite 1
Las Vegas, NV 89120

W. E. Wilson
USGS-WRD, Mail Stop 416
Federal Center
Denver, CO 80225

G. S. Bodvarsson
Lawrence Berkeley Laboratory
Berkeley, CA 97420

1511	N. E. Bixler
1511	R. R. Eaton
6300	R. W. Lynch
6310	T. O. Hunter
6310	CF 72-12144-1.6/NQ
6311	L. W. Scully
6311	C. Mora
6312	G. E. Barr
6312	F. W. Bingham
6312	A. L. Dudley
6312	J. H. Gauthier
6312	N. K. Hayden
6312	B. S. Langkopf
6312	R. R. Peters (20)
6312	R. W. Prindle

Figure 2. A sample image with segmented text-lines highlighted with yellow bounding boxes, and its histogram of the vertical projection profiles, where the local maxima are marked as red ◇, and the peaks are marked as yellow □.

3.2. Page Content Grouping and Frame Detection

The page frame detection is accomplished by identifying the column layout of images in our algorithm. First, we compute the histogram of the vertical projection profile [12]:

$$H_i = \sum T_i + \sum I_i \tag{1}$$

where H_i is the histogram value at column i, $\sum T_i$ and $\sum I_i$ are the number of text-lines and images at column i, respectively. Note that H is built on the text-line/image level instead of the pixel level, so that it is less sensitive to the variations like font size, character spacing or scanning noises.

We then identify the column layout by searching the peak areas in H. However, we need be careful because one

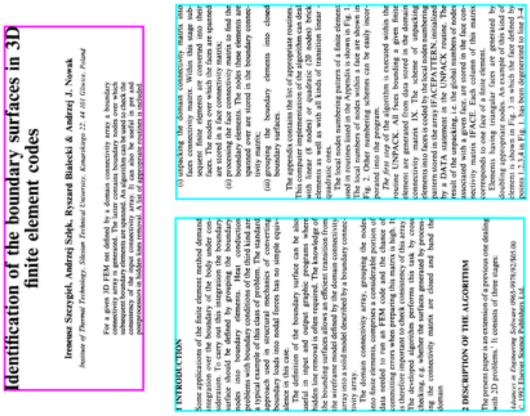


Figure 3. An example of grouping the detected blocks, where the text-lines classified into groups are circled with the light blue bounding boxes, and those do not belong to any group are circled with a red bounding box.

column might have more texts than the other(s) so that it has several peaks in H, such as the example shown in Fig. 2, where the left column has actually three peaks. To avoid the over-segmentation problem, we use the local maximum instead, with the bin size set to $1/8$ of the image width, based on our observation that a page usually has no more than four columns. Specifically, one group G is created for every local maximum in H, with the boundary G_L and G_R set to cover the corresponding peak area. And a text-line/image is inserted into a group if *at least half of its width is inside the group area*. Then we update the G_L and G_R so that they exactly cover all group components, i.e., the text-lines and the images of G. Note that not all the segmented block are classified into groups. Some of them, like the header lines that cross several local maxima areas, do not belong to any group. Fig. 3 lists the grouping result on a sample image, where the text-lines in each column are classified into one group and circled with a light blue bounding box, and the header lines are not grouped and circled with a red bounding box.

Next, we select groups and filter out those likely to be noises due to bad scanning. The selection starts with creating a group set GS, and then iteratively inserting groups that are similar to each other:

$$G_i \in GS \text{ if } \forall_{G_j \in GS} \overline{S(G_i, G_j)} > Th \qquad (2)$$

In our algorithm, the group similarity value is computed from the ratio between the mean width of the k-widest text-lines/images in a group ($L(G)$):

$$S(G_i, G_j) = \frac{min(L(G_i), L(G_j))}{max(L(G_i), L(G_j))} \qquad (3)$$

On the other hand, we still need an initialization for

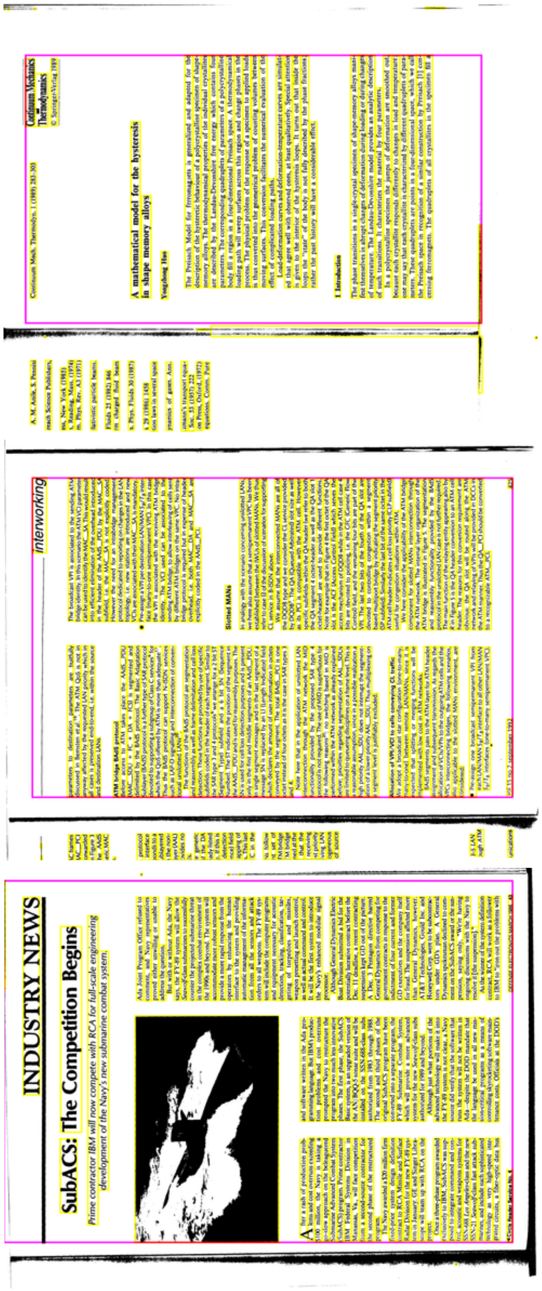


Figure 4. The grouping results on sample images of one column page (top), two columns page (middle), and three columns page (bottom); the detected text-lines are circled with yellow bounding boxes, and the created groups are circled with red bounding boxes;

Eq. 2, where we choose the group with the largest L value.

We discard all segmented blocks classified into a group that is not clustered into GS since they are most likely to be noises. And then we repeat the process, starting from building the histogram of the vertical projection profiles of text-lines/images, until no group is filtered any more. So now we can estimate the vertical page frame using the leftmost and rightmost boundaries of the selected groups. Fig. 4 lists the grouping results of three sample images of one column page, two columns page, and three columns page. And we see that all of them have the page contents correctly estimated.

The grouping algorithm described above separates the page content from the side border noises, which are the most encountered errors in scanned books. On the other hand, we still need a method to estimate the top and bottom boundary of a page frame, where we proposed an online training classifier in [7]. In brief, the classifier automatically labels the text-lines in an image as *high confidence* or *low confidence*, then it learns features from the high confidence text-lines, and applies them to filter out the low confidence text-lines that are likely to be noises.

4. Experiments

4.1. Performance Evaluation of Our Document Image Segmentation Algorithm

We first evaluate our algorithm on the popular UW-III database [10] consisting of 1600 images of different qualities, which has been widely used by the DAR communities to measure the performances of the document image segmentation algorithms [19, 20, 13]. And we compute the performances using the two metrics employed in the Faisal Shafait *et. al*'s work [20]:

- **Area Overlap**: it uses the ratio between the area intersection and the area union of the detected page frame (F_d) and the ground truth page frame (F_g):

$$M_A = \frac{2|F_d \bigcap F_g|}{|F_d| + |F_g|} \tag{4}$$

- **Component Classification**: it defines the components inside the detected page frame (F_d) as "positive", and those outside F_d as "negative". And it quantifies the performance as the *true positive rate* (M_{TP}) and the *true negative rate* (M_{TN}):

$$M_{TP} = \frac{NC_{P \in F_g}}{(NC_P + NC_N) \in F_g} \quad , \quad M_{TN} = \frac{NC_{N \in \widetilde{F_g}}}{(NC_P + NC_N) \in \widetilde{F_g}} \tag{5}$$

Fig. 5 compares the performances of our algorithm with two state of art page segmentation algorithms: IUPR [20] and Scansoft [3]. It shows that our algorithm achieves 96% area overlap rate, which is the best among the three of them. And it has 91.2% true negative rate, a substantial improvement over the other two that are below 75%. In addition, it also has 98.5% true positive rate that is as competitive as the others.

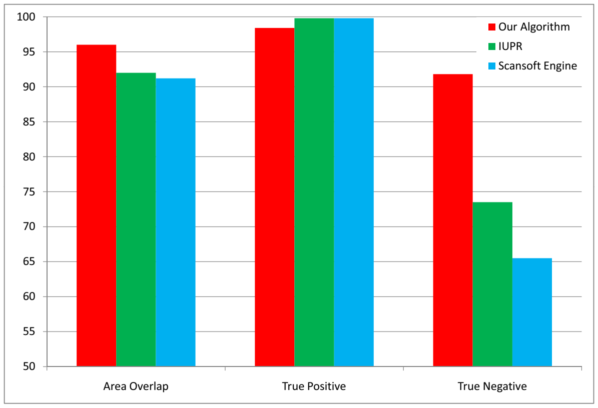


Figure 5. The page frame level segmentation performance comparison between our algorithm, IUPR [20], and Scansoft [3], on the UW-III database with 1600 images; all the metrics are reported in percentage (%).

The results above show that our algorithm is ideal for segmenting pages with noisy background. This is especially valuable for print-on-demand books, because most of our books were scanned copies with various degrees of background noise.

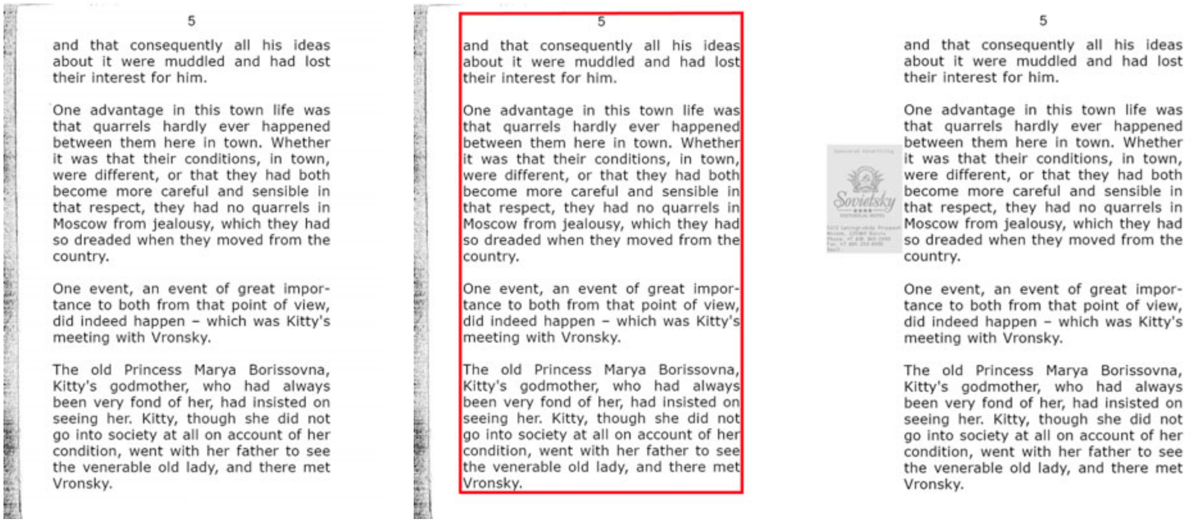
5. Examples of Formatted Page with Advertisement Insertion

For illustration purpose, we collected several advertisement with text and graphics, and inserted them into the margin of the cleaned pages. Fig. 6 shows example results.

6. Conclusions and Future Study

This paper introduced the new media of print-on-demand books and proposed a user-targeted contextual advertising model in this media. One of the key enabling factor of print-on-demand advertising is understanding the layout of the book pages and identifying where we can insert advertisement. We described our book page segmentation algorithm in Section 3, and compare its accuracy against that of the two state of the art page segmentation systems IUPR [20] and Scansoft [3] using the UW-III data set. The result shows we achieves significantly better accuracy in detecting background noise, while maintain similar accuracy in detecting normal page content.

Print on demand adver1tising is a whole new application which faces many technically intriguing challenges. This paper mainly addresses the problems of contextual matching and book page segmentation. In the future, we will work on page re-formatting as well as using mobile-device and printed barcode to provide additional online feedback chan-



5

and that consequently all his ideas about it were muddled and had lost their interest for him.

One advantage in this town life was that quarrels hardly ever happened between them here in town. Whether it was that their conditions, in town, were different, or that they had both become more careful and sensible in that respect, they had no quarrels in Moscow from jealousy, which they had so dreaded when they moved from the country.

One event, an event of great importance to both from that point of view, did indeed happen – which was Kitty's meeting with Vronsky.

The old Princess Marya Borissovna, Kitty's godmother, who had always been very fond of her, had insisted on seeing her. Kitty, though she did not go into society at all on account of her condition, went with her father to see the venerable old lady, and there met Vronsky.

5

and that consequently all his ideas about it were muddled and had lost their interest for him.

One advantage in this town life was that quarrels hardly ever happened between them here in town. Whether it was that their conditions, in town, were different, or that they had both become more careful and sensible in that respect, they had no quarrels in Moscow from jealousy, which they had so dreaded when they moved from the country.

One event, an event of great importance to both from that point of view, did indeed happen – which was Kitty's meeting with Vronsky.

The old Princess Marya Borissovna, Kitty's godmother, who had always been very fond of her, had insisted on seeing her. Kitty, though she did not go into society at all on account of her condition, went with her father to see the venerable old lady, and there met Vronsky.

5

and that consequently all his ideas about it were muddled and had lost their interest for him.

One advantage in this town life was that quarrels hardly ever happened between them here in town. Whether it was that their conditions, in town, were different, or that they had both become more careful and sensible in that respect, they had no quarrels in Moscow from jealousy, which they had so dreaded when they moved from the country.

One event, an event of great importance to both from that point of view, did indeed happen – which was Kitty's meeting with Vronsky.

The old Princess Marya Borissovna, Kitty's godmother, who had always been very fond of her, had insisted on seeing her. Kitty, though she did not go into society at all on account of her condition, went with her father to see the venerable old lady, and there met Vronsky.

(a) (b) (c)

Figure 6. Examples of inserting advertisements to print on demand books: (a) the original input image (b) The detected page frame highlighted with a red bounding box, and (c) the cleaned output image after page alignment and advertisement insertion.

nel.

References

[1] Amazon kindle. http://en.wikipedia.org/wiki/Kindle, 2007. 1

[2] Leptonica web pages. http://www.leptonica.com/, 2008. 2

[3] Scansoft web pages. http://www.nuance.com/, 2009. 5

[4] Seattle P-I publishes its last edition. http://www.seattlepi.com/business/403793_piclosure17.html, 2009. 1

[5] NY Times publisher manages transition from print to internet. http://www.haaretz.com/hasen/spages/822775.html, Feb 8, 2007. 1

[6] Infotrends forcasts us print on demand market. http://www.seyboldreport.com/infotrends-forcasts-growth-us-print-demand-market, Nov 6, 2008. 1

[7] Anonymous. A semi-supervised learning framework for document layout analysis. In *Submitted to the 12th International Conference on Computer Vision*, Sep. 2009. 5

[8] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel. A semantic approach to contextual advertising. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 559–566, New York, NY, USA, 2007. ACM. 2

[9] J. Gomez. *Print Is Dead: Books in our Digital Age*. Palgrave Macmillan, November 13, 2007. 1

[10] I. Guyon, R. M. Haralick, J. J. Hull, and I. T. Phillips. Data sets for ocr and document image understanding research. In *In Proceedings of the SPIE - Document Recognition*, number IV, pages 779–799, 1997. 5

[11] A. Lacerda, M. Cristo, M. A. Goncalves, W. Fan, N. Ziviani, and B. Ribeiro-Neto. Learning to advertise. In *Proceedings of the 29th International ACM SIGIR Conference*, pages 549–556, New York, NY, 2006. 2

[12] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. In *Computer*, volume 25, pages 10–22, July 1992. 3

[13] O. Okun, D. Doermann, and M. Pietikainen. Page segmentation and zone classification: The state of the art. Technical Report 37, University of Maryland, College Park, Nov. 1999. 5

[14] P. M. Parker. *The 2009-2014 World Outlook for Online Display Advertising*. ICON Group, December 2, 2008. 1

[15] P. M. Parker. *The 2009-2014 World Outlook for Online Video Advertising Services*. ICON Group, September 20, 2008. 1

[16] P. M. Parker. *The 2009-2014 World Outlook for Paid Internet Search Advertising*. ICON Group, September 27, 2008. 1

[17] P. M. Parker. *The 2009-2014 World Outlook for Publishing Advertising*. ICON Group, September 27, 2008. 1

[18] W. Postl. U.S. patent, Num. 4723297, 1988. 2

[19] F. Shafait, D. Keysers, and T. M. Breuel. Performance evaluation and benchmarking of six page segmentation algorithms. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 30, pages 941–954, June 2008. 5

[20] F. Shafait, J. van Beusekom, D. Keysers, and T. M. Breuel. Page frame detection for marginal noise removal from scanned documents. In *Proceedings of Image Analysis*, volume 4522, pages 651–660, 2007. 5