Intelligent Content Fitting for Digital Publishing

Xiaofan Lin
Imaging Systems Laboratory
HP Laboratories Palo Alto
HPL-2005-160
September 12, 2005*

digital publishing, variable data printing, layout and composition, automatic image cropping, XSL-FO

One recurring problem in Variable Data Printing (VDP) is that the existing contents cannot satisfy the VDP task as-is. So there is a strong need for content fitting technologies to support high-value digital publishing applications, in which text and image are the two major types of contents. This paper presents meta-Autocrop framework for image fitting and TextFlex technology for text fitting. The meta-Autocrop framework supports multiple modes: fixed aspect-ratio mode, advice mode, and verification mode. The TextFlex technology supports non-rectangular text wrapping and paragraph-based line breaking. We also demonstrate how these content fitting technologies are utilized in the overall automated composition and layout system.

* Internal Accession Date Only
Published in and presented at the SPIE Conference on Digital Publishing, 15-19 January 2006, San Jose, CA, USA
Approved for External Publication

Intelligent Content Fitting for Digital Publishing

Xiaofan Lin

Hewlett-Packard Laboratories
1501 Page Mill Rd MS 1203, Palo Alto, CA 94304
Email: xiaofan.lin@hp.com

ABSRACT

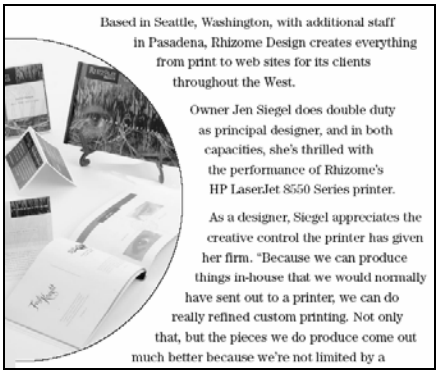
One recurring problem in Variable Data Printing (VDP) is that the existing contents cannot satisfy the VDP task as-is. So there is a strong need for content fitting technologies to support high-value digital publishing applications, in which text and image are the two major types of contents. This paper presents meta-Autocrop framework for image fitting and TextFlex technology for text fitting. The meta-Autocrop framework supports multiple modes: fixed aspect-ratio mode, advice mode, and verification mode. The TextFlex technology supports non-rectangular text wrapping and paragraph-based line breaking. We also demonstrate how these content fitting technologies are utilized in the overall automated composition and layout system.

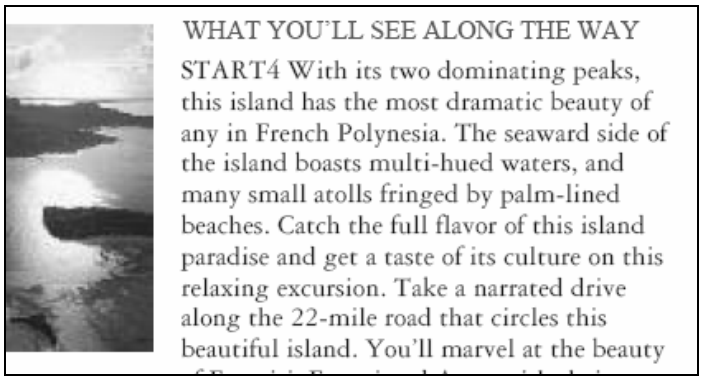
Keywords: digital publishing, variable data printing, layout and composition, automatic image cropping, XSL-FO

1. INTRODUCTION

Variable Data Printing (VDP) provides a huge opportunity for digital publishing and printing business. In addition to creating new contents specifically for the targeted VDP application, we also have several other venues to obtaining the contents: from the past campaigns, on the Internet (with appropriate copyright treatment), or from third parties. However, one recurring problem is that the existing contents cannot satisfy the VDP task as-is. For example, the aspect ratio of an image cannot fit into the copy hole of a layout; too much customized text cannot fit into a text block. So there is a strong need for content fitting technologies to support high-value digital publishing applications. As the two major types of variable data contents, text and image both need technologies that can automatically adapt them in order to achieve the best overall layout in VDP.

The basis of text fitting is a format to describe text flow. A number of layout description formats and related software systems exist for VDP. For example, XSL-FO is a widely accepted as an XML-based W3C standard [1] and suitable feature set for VDP. Besides, the Apache FOP [2] emerges as a popular implementation of parsing and rendering XSL-FO documents because of its open-source nature and reasonable quality. Such openness is XSL-FO/FOP's major advantage over commercial graphics design software such as Adobe InDesign and QuarkXpress and their proprietary formats. On the other hand, XSL-FO and FOP only support simple text placement within a rectangular container (see Figure 1(b)), and they do not provide several advanced text handling features needed in high-end VDP applications: text wrapping (placing text in a non-rectangular container around the boundary of an object, see Figure 1(a)), and paragraph-based line breaking (deciding how to break a paragraph into text lines using global optimization).





(a). Desired non-rectangular text wrapping (b). Rectangular text wrapping supported by XSL-FO and FOP

Figure 1: Mismatch between the desired layout features and existing formats and systems

The most common form of image fitting is Autocrop, which intelligently crops the images to satisfy the overall layout requirements. Most existing work on Autocrop utilizes variants of saliency maps to indicate the important regions of an image [3][4]. However, there is no published work on how to customize Autocrop to satisfy the requirements of automatic layout system in VDP.

This paper introduces novel technologies that can effectively solve the aforementioned problems in text and image content fitting. TextFlex is an advanced text handling adaptor to XSL-FO and FOP. With TextFlex, we can support complex layouts (see Figure 1(a)) in VDP applications on top of standard XSL-FO and FOP. TextFlex also features paragraph-based line breaking. Besides, meta-Autocrop framework is proposed to provide multi-mode Autocrop services for VDP. Autocrop results can be directly expressed constraints to drive automatic layout algorithms.

2. TEXTFLEX

In Section 1, we point out the lack of open text layout formats that support advanced text handling features. In theory, there are several alternatives to this problem: 1) Wait for future versions of XSL-FO with the desired features added and the new version of the system supporting the features; 2) Go with another format and system that support the features; and 3) Adapt the existing format and system. In reality, Option 1 is full of uncertainty and runs the risk of losing competition advantages. Option 2 requires fundamental system redesign and thus substantial development effort. Thus, Option 3 stands out as the most practical solution. Along this direction, we have introduced TextFlex technology. The key ideas are:

- Enhance the existing file format by adding necessary element/attribute types.
- Develop a preprocessor to parse the enhanced files. The elements/attributes supported in the existing file format/system will be passed to the output file stream as-is. When the preprocessor encounters the elements/attributes of the customized types, it will invoke special component to convert those elements/attributes into the elements/attributes supported by the existing format, and then insert them to the output file stream.
- The resulting output file will then be processed or rendered by existing systems.

As shown in Figure 4, we insert extend *textwrap* objects into ordinary XSL-FO documents. The *shapefile* attribute of points to an external copy hole shape description file that supports polygons, circles, and ellipses, etc. When the text handling adaptor encounters *textwrap* in the parsing process, it will retrieve the text content, font information, and copy hole shape. Figure 2 displays an example. Then the text is placed into the copy hole in several steps. The first step is to decompose each shape into a stack of rectangular blocks using computational geometry algorithms. Then the text in each paragraph is placed into the text line blocks. FOP's built-in line-breaking algorithm is a greedy line-by-line process, which attempts to put as much as possible text into each line. This algorithm is very simple, but the line-breaking results are far from satisfactory. As seen in Figure 3(a), it can result in very uneven right edges. In contrast, the paragraph-based line-breaking algorithm, which was first introduced by Donald Knuth in Tex [6], can produce much better results (see Figure 3(b)). The key idea of paragraph-based algorithm is to use dynamic programming to globally optimize some aesthetic cost function. Thus, the algorithm considers the effect on following lines when breaking the current line. TextFlex incorporates an improved version of Chung-chien Shan's C++ implementation [7]. Our improvement has added the support of variable widths of characters and lines. Depending on the applications, optionally font size may be adjusted to make the text fit into the bounding box. After that, individual text lines are rendered as separate text blocks in standard formats such as XSL-FO or SVG based on the calculated start positions and dimensions.

Based in Seattle, Washington, with additional staff in Pasadena, Rhizome Design creates everything from print to web sites for its clients throughout the West. Owner Jen Siegel does double duty as principal designer, and in both capacities, she's thrilled with the performance of …

(a). Text content to be laid out

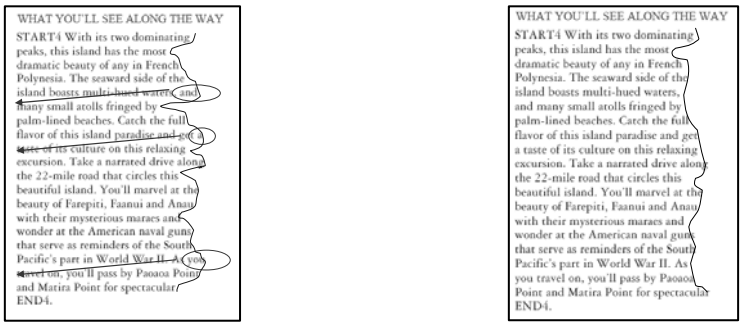
```
font-family2 Courier
font-family Frutiger-Roman
font-style normal
font-weight normal
font-size 10
letter-spacing 1
line-height 11.36
text-align justify
```

(b). Font specification

```
293 272
293 284
319 287
319 299
335 302
335 314
348 317
348 329
361 338
…
```

(c). Polygon description

Figure 2: Sample input files specified for <textwrap>



(a). Greedy algorithm (b). Paragraph-based algorithm

Figure 3: Comparison of greedy and paragraph-based line breaking

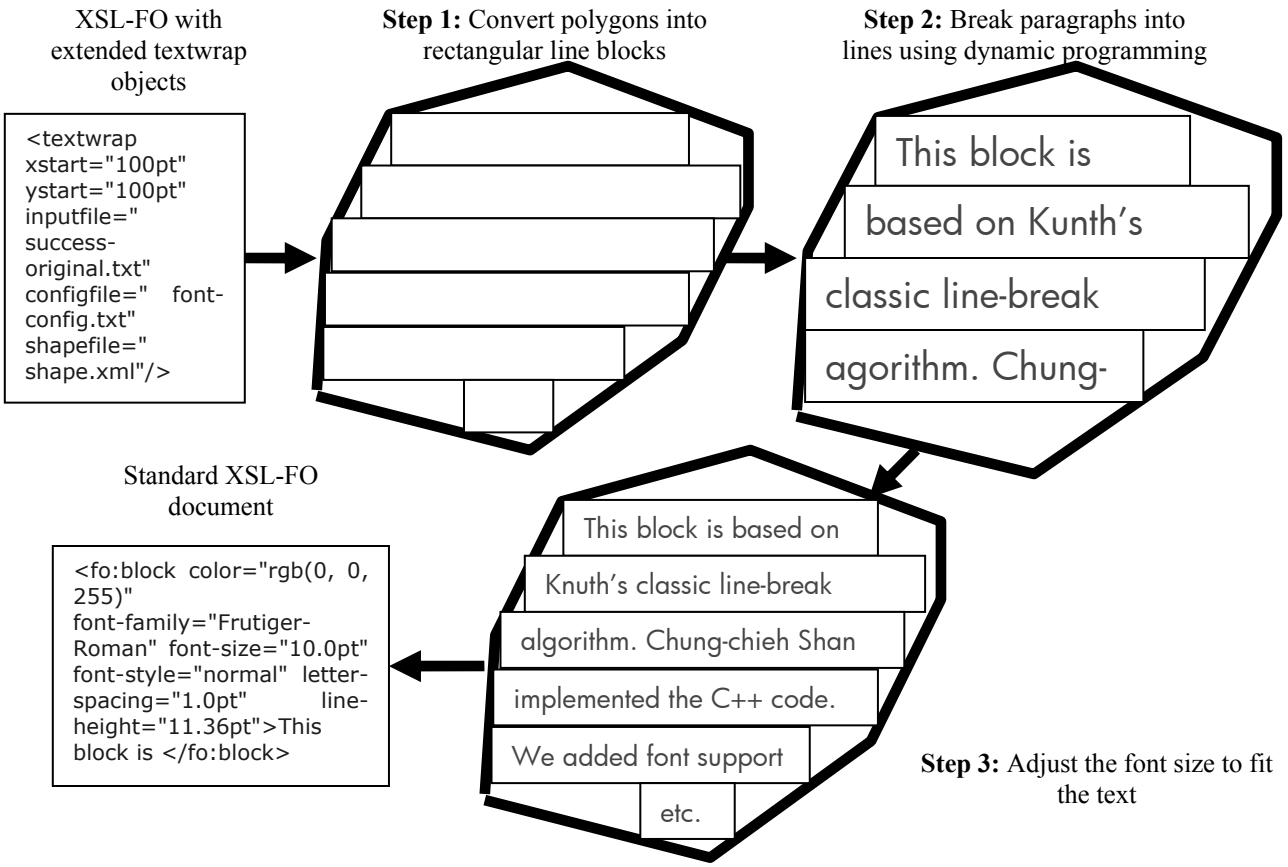


Figure 4: Text Placement in TextFlex

3. META-AUTOCROP FRAMEWORK

Although automatic cropping can be carried out using different image analysis algorithms, no single algorithm can work well for a wide variety of images. So we have incorporated a number of operational modes, each of which utilizes a different cropping algorithm, and then we can select the mode most suitable for a particular input image based on the image category or application type.

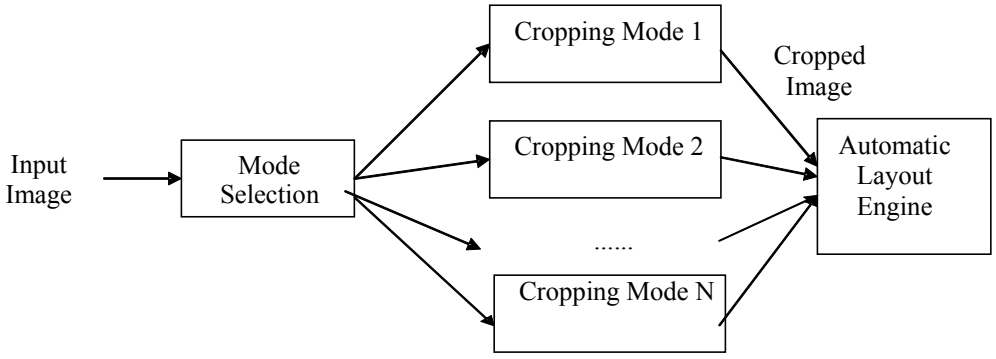


Figure 5: Meta-Autocrop framework

The above figure displays the configuration of the proposed meta-Autocrop framework. The input image is first sent into "Mode Selection" module (MS), which will decide the actual cropping mode to be invoked. MS makes decisions

based on a number of factors such as metadata associated with the images, the image category, and the feedback from the master application. The application can impose constraint on the cropping. For example, it may require that the crop comply with a specific aspect ratio.

To cater for different requirements of VDP, we have implemented the following Autocrop operational modes:

Table 1: Modes in meta-Autocrop framework

No	Mode	Description	Application
1	Fixed Aspect Ratio	Find crops that satisfy the specified aspect ratio	Deal with the copy hole constraint.
2	Advice Mode	Find the extent of cropping along the four borders	Integrated with constraint-based layout algorithms
3	Affine-transform Based Non-rectangular Cropping	Preprocess the image with affine transform and map the rectangular crops into the original coordinates to get non-rectangular crops	Offer more freedom of generating non-rectangular cropping

The Fixed Aspect Ratio Mode is designed and implemented by Phil Cheatle [4]. It is based on the clustering of saliency map. Figure 6 shows a typical example where the original image taken from a digital camera is cropped to fit into 6x4 photo paper.

(a) Original 4:3 digital photo (b) Cropped 3:2 picture

Figure 6: Example of Aspect Ratio Mode Autocrop

The Advice Mode decides the maximal extent of cropping along the four borders, and it gives the results at different risk levels: Conservative, Aggressive, and Ultra-aggressive. Table 2 lists the criteria and application domains of the three sub-modes. Figure 7 demonstrates the cropping results under the three sub-modes. It can be seen that Ultra-conservative Sub-mode works best for Figure 7(a). The other two sub-modes will cut off the tab of the package because the color of the tab is very close to the pure background. In contrast, Conservative Sub-mode does the best job on Figure 7(b) because the color clustering accurately merges the cloud into a single background area. The Ultra-conservative Sub-mode crops nothing at all in this case.

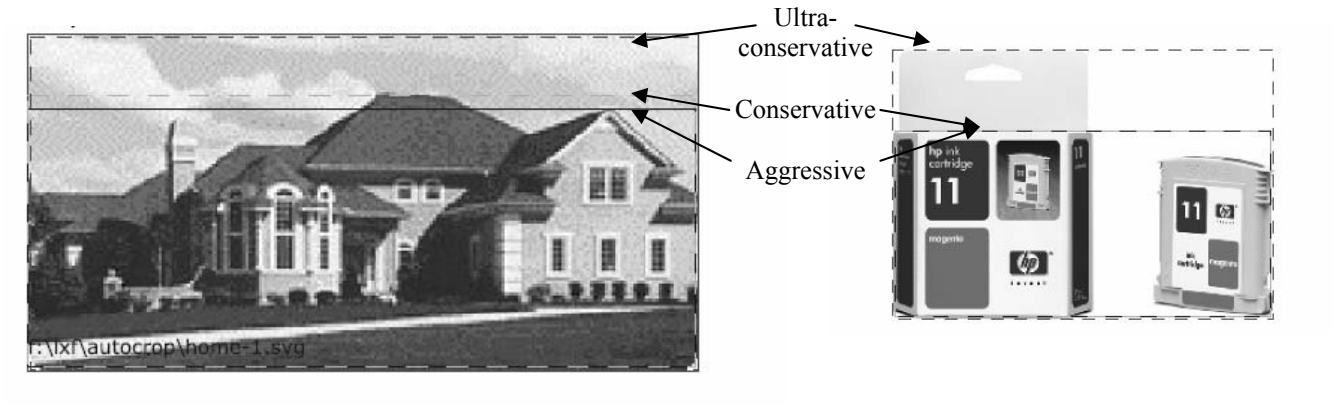
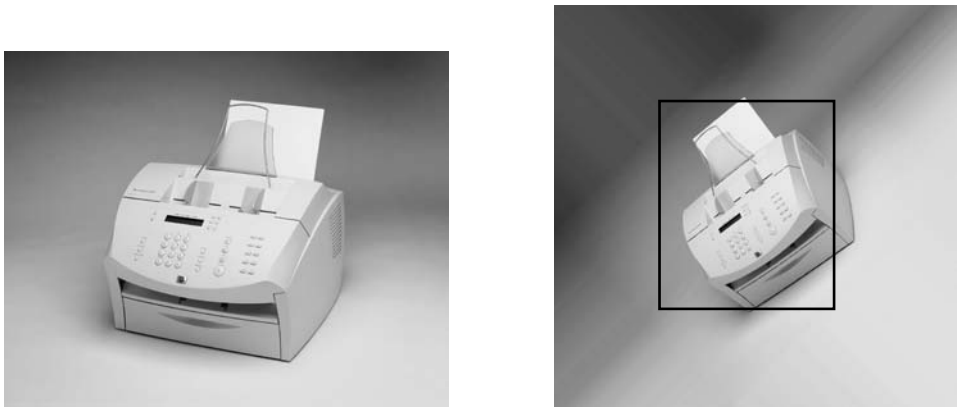


Figure 7: Results of different sub-modes under Advice Mode

Table 2: Sub-modes of Advice Mode

No	Sub-mode	Cropping algorithm	Suitable image category or application domain
1	Aggressive	Saliency map	Unedited natural photos with complicated background
2	Conservative	Color clustering	Objects with a relatively even background
3	Ultra-conservative	Color keying	Processed product pictures with very homogenous background

In many professional publishing applications, the copy hole can be of shapes other than upright rectangles. Thus, there exists a need to expand Autocrop to find non-rectangular shaped crops. In the affine-transform based non-rectangular cropping mode, an affine transform is designed according to the expected cropping shapes. Then this transform is applied to the original image. The resulted image is processed by conventional Autocrop methods to locate regular rectangular crops. Finally the crops are mapped back to the original coordinate system to get the crops of the desired shapes or orientations. In this way, we can obtain parallelogram-shaped crops. Figure 8 shows the results after different steps.



(a) (b)

(c)

Figure 8: Example of the non-rectangular cropping based on affine transform

(a). Original image (b). Distorted image with conventional Autocrop result (Red box)

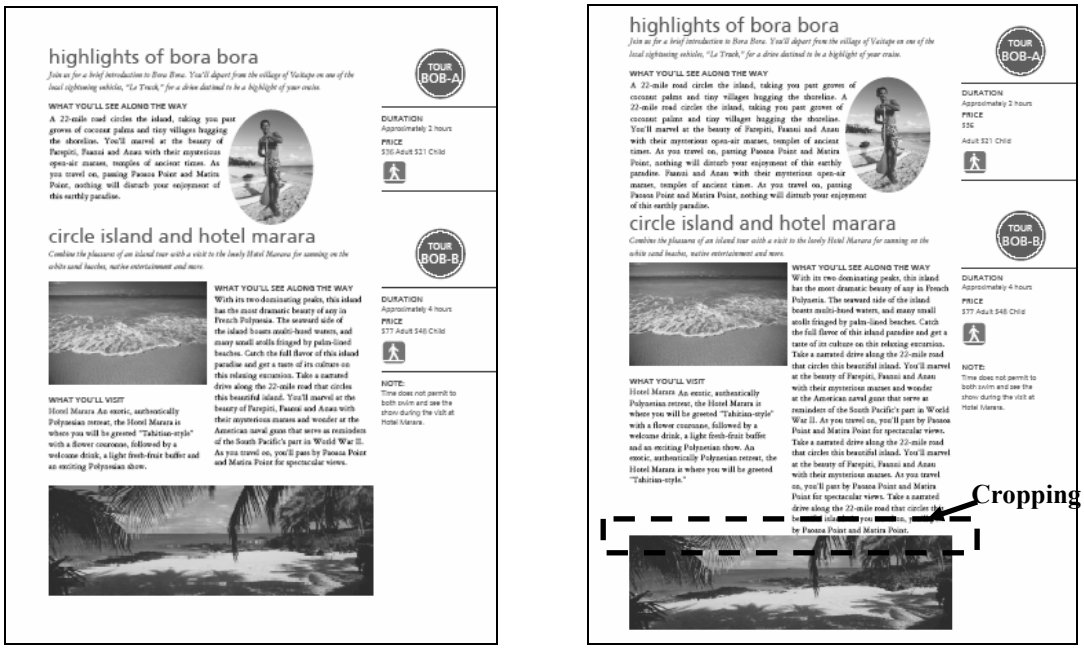
(c). Crop (Red box) on the original image

4. APPLICATIONS OF CONTENT FITTING TECHNOLOGY IN VDP

Figure 9: Role of content fitting in digital publishing pipleline

As shown in Figure 9, TextFlex and Autocrop have been integrated into HP Labs' end-to-to digital publishing pipeline. TextFlex has been used to support nonrectangular text wrapping. Autocrop can be directly integrated with the constraint-based Active Layout Engine (ALE) [9]. The advice mode results can be represented as linear constraints. For example, "<constraint rule="20 >=topoffset" strength="required" />" means that the top of an image cannot be cropped by more than 20 pixels. Our current implementation leverages the constraint description part of the CSVG language [8]. Taking advantage of the clipping path in SVG and Active Layout Template [9], we do not need to actually generate the cropped image files. The following SVG segment demonstrates this virtual cropping technique. The demo22.JPG enclosed inside the SCG corresponds to the image at the bottom of the document shown in Figure 10. When ALE sets topoffset to 20, we get the layout of Figure 10(b).

```
<svg xml:space="preserve" width="401" height="138">
<g>
<clipPath id="MyClip">
<rect style="stroke:green;fill:none" x="0" y="##topoffset" width="401" height="##+138-topoffset "/>
</clipPath>
</g>
<g  clip-path="url(#MyClip)">
<image xlink:href="demo22.JPG" id="red" x="0" y="0" width="401" height="138"/>
</g>
</svg>
```



(a). Original layout (b). After layout adjustment based on new text contents

Figure 10: Application of Autocrop and TextFlex in Active Layout Engine

5. CONCLUSIONS

Content fitting is a significant technology in the automated end-to-end VDP workflow. Working together with automatic layout technology, the creation side bottleneck can be relieved or even eliminated in particular applications. On the other hand, content fitting is a multidisciplinary area. Text fitting requires mathematical optimization, text composition, and computational geometry. Autocrop heavily relies on image processing techniques.

One interesting question about Autocrop is how reliable and robust it is compared with human beings. We have addressed this benchmarking question in another paper [5]. Besides, although TextFlex has already added valuable capabilities on top of standard XSL-FO, there are still more features to be supported, such as in-line text and sophisticated kerning. In addition, in order to utilize the content fitting technology in production environment, we have to deal with many challenges, including system optimization, distributed computing, and fault tolerance.

ACKNOWLEGEMENTS

The author is grateful to Phil Cheatle for providing the baseline Autocrop implementation, to Hui Chao for supplying the XSL-FO layout files, to Menaka Indrani and Elsa Durante for integrating the TextFlex technology into the overall system, and to Hector Santos for benchmarking the Autocrop algorihtms. Anna Durante, Gary Vondran, and Henry Sang have all championed this research from the start.

REFERENCES

1. http://www.w3.org/TR/xsl/
2. http://xml.apache.org/fop/index.html
3. B. Suh, H.B. Ling, B. B. Bederson, D. W. Jacobs, "Automatic Thumbnail Cropping and its Effectiveness," *UIST 2003*, Nov 2003
4. P. Cheatle, “Automated Cropping of Electronic Images”, US Patent Application 20020191861

5. H. J. Santos-Villalobos, X. Lin, “Benchmarking of Automated Image Cropping Technology,” HP Labs Technical Report, 2005
6. D. E. Knuth, :“Breaking Paragraphs into Lines,” *Software Practice and Experience*, Vol.11, pp 1119-1184, 1981
7. http://www.digitas.harvard.edu/cgi-bin/wiki/ken/FmtTwo
8. G. Badros, J. Tirtowidjojo, et al, “A Constraint Extension to Scalable Vector Graphics,” *Proceedings of Tenth International World Wide Web Conference*, May 2-5, 2001, Hong Kong
9. X. Lin, H. Chao, G. Nelson, E. Durante, “Active Document Versioning: From Layout Understanding to Adjustment,” submitted to *Document Recognition and Retrieval Conference XII*I, to be held in San Jose, January 2006