A Geometric Bisection Algorithm for Automatic Page Layout

Alexandre Seki, Mauricio Klein, Ricardo Piccoli,
Isabel Manssour, João B. Oliveira
FACIN - Faculdade de Informática - PUCRS
{alexandre.seki, mauricio.klein, ricardo.piccoli}@cpph.pucrs.br,
{manssour, oliveira}@inf.pucrs.br

Abstract

Content distribution in a page designed for printing is a difficult task performed by human editors. There are applications and tools to help the editor in this task, but it is still a manual, iterative process that is made more difficult when content is distributed in columns, as is the case of newspapers. This work describes an algorithm to automatically distribute content in a printable page. Our implementation tries to use all the page, dividing it among the items to be placed according to their area.

1. Introduction

One of the problems in the design of magazines and newspapers is to distribute elements on a page. A designer is responsible for creating the page layout, distributing elements to achieve some degree of aesthetic quality. To reach a high-quality final result, that professional usually edits the layout several times, changing the news positions and format. There are tools to help the designer to format the page layout, but these tools do not generates the layout automatically and only support human intervention.

A typical bisection algorithm is an algorithm that repeatedly divides an interval in half and recursively processing both halves. Here, such an algorithm is used to distribute elements into a page, a task that will help many areas of digital publishing. Besides document designers, common users may also produce printable pages with a RSS feed content or pictures.

Jacobs *et al.* [3, 5] present an algorithm to generate layout automatically, using templates to guide the layout construction. Harrington *et al.* [4] propose a genetic algorithm to distribute elements on a page, but the computational cost of a genetic algorithm is very high for massive production. The distribution of photos on a page was researched by Atkins [1]. Another solution for this problem using genetic algorithms was presented by Geigel and Loui [2].

The main goal of this work is to describe and present results of an algorithm to automatically distribute elements into a page. This algorithm was developed as part of the APL (Automatic Page Layout) project. The current version is able to produce documents akin to newspapers covers. In our case, a user selects some news and pictures to be placed on the page and the algorithm distribute these elements. The remaining of the poster is organized as follows: the algorithm and the obtained results are described in Section 2; future work and conclusions are presented in Section 3.

2. The APL project

The main goal of the APL Project is to develop ways of distributing content automatically on a page. In our algorithm, the elements of a page are news, pictures and groups of such elements. News are composed by a title, the content text and optional attributes, such as author or the page number to the full news. Pictures are placeholders for an image or picture, and groups are used to aggregate them. For example, a group of two images and a piece of news will have its elements placed together, either side by side or one above the other, thus keeping elements organized.

The algorithm is based on a divide and conquer strategy: it receives the page width and height and a list of elements to be placed, and through consecutive bisections it divides the page area into regions, trying to allocate the required area of a news into this region. In our case, bisection is made by a horizontal or vertical line dividing a region into two new regions and, similarly, the list of elements are also divided. We can see an example of the defined regions for some news in Fig. 1a and 1b. In these figures, we have used the same news and only the number of columns was changed.

The algorithm tries to keep the elements in reading order (as given by the input sequence), from the top left to the bottom right. This strategy reduces the amount of possibilities to be examined and also reduces the surprise of the user, as news are not placed randomly. The first tests
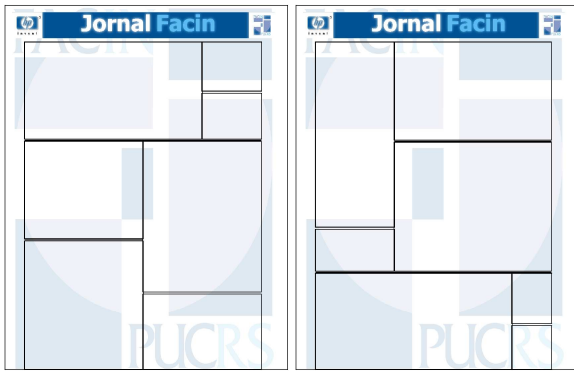
Figure 1. Specified regions in a page (a) with 4 and (b) with 6 columns.
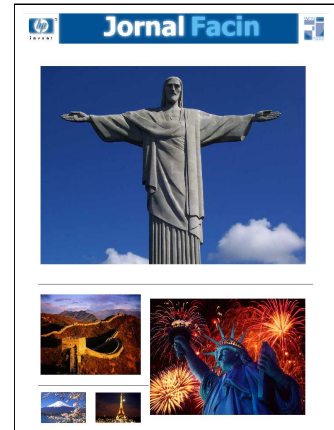
were executed, automatically generating newspaper covers, the content were pictures and news. We have high-quality results in a short processing time. Fig. 2a shows an example of a generated newspaper cover that has as input four news and three pictures.

Our algorithm can be used to produce pages from RSS feeds. In this case, most of news are short and have a link to the full news or other related pages. Another peculiarity of RSS feeds is that usually there are no pictures. An example of a cover generated from RSS feeds is shown in Fig. 2b. Finally, as in Fig. 3, our algorithm can also place pictures, images or photos on a page.

Figure 2. Examples of generated pages (a) from newspaper content and (b) from RSS feed.

The project is being developed using Java as the main programming language. The input file to the algorithm is an extension of XML obtained by converting RSS feeds and/or Wiki formats into our XML extension. The program can generate different outputs, such as PDF, LaTeX and dvi.

Figure 3. Generated page with pictures

3. Conclusions and Future Work

This poster presents a new algorithm to distribute content automatically in a printable page divided into columns, preserving reading order and taking into account the area of each news. Future works include finishing the prototype to make it available for tests and evaluation; extending the page layout from one to several pages and the development of a user interface for simpler use.

Acknowledgements

References

[1] C. B. Atkins. Adaptive photo collection page layout. In *ICIP*, pages 2897–2900, 2004.

[2] J. Geigel and A. C. P. Loui. Automatic page layout using genetic algorithms for electronic albuming. volume 4311, pages 79–90. SPIE, 2000.

[3] C. Jacobs, W. Li, E. Schrier, D. Bargeron, and D. Salesin. Adaptive grid-based document layout. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 838–847, New York, NY, USA, 2003. ACM.

[4] L. Purvis, S. Harrington, B. O'Sullivan, and E. C. Freuder. Creating personalized documents: an optimization approach. In *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, pages 68–77, New York, NY, USA, 2003. ACM.

[5] E. Schrier, M. Dontcheva, C. Jacobs, G. Wade, and D. Salesin. Adaptive layout for dynamically aggregated documents. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 99–108, New York, NY, USA, 2008. ACM.