Hybrid Web-page Segmentation and Block Extraction for Small Screen Terminals

Shefali Singhal
Assistant Professor IT Department, Manav Rachna International University, Faridabad
Sec-43, Aravalli Hills Faridabad-121006

Neha Garg
Assistant Professor CSE Department, Manav Rachna International University, Faridabad
Sec-43, Aravalli Hills Faridabad-121006

ABSTRACT
Web page representation is a topic of concern for small screen devices, like, mobile, palm, etc. In a web-page, bulk of irrelevant data including advertisements and other noisy information's create access inconvenience. Web page segmentation is a technique which resolves this problem by logically dividing a web page into segments. These segments can be created by using DOM (Document Object Model) and VIPS (Visual Page Segmentation) techniques. In this paper, a hybrid method of web page segmentation has been designed using combination of DOM method and VIPS algorithm for developing segments from a web page. Here both the structural and visual aspects of a web page to create a segment have been considered. A segment is such a basic unit of web page which cannot be further divided. This is done by processing a web page through a BLOCK CREATION ALGORITHM which is discussed further.

General Terms
DOM method and VIPS Algorithm.

KEYWORDS
Web page segmentation, Block Extraction Algorithms and System Architecture.

INTRODUCTION
Web page segmentation is a solution towards all type of modifications in a web page whether it belongs to any type of format or structure. However, there are many methods which were previously proposed to extract segments from a web page. These searches resulted into either structural segments or visual segments. These segments are independent and can be created using different algorithms.

In this paper the Document Object Model (DOM) tree representation of web pages is analysed, in order to identify their segments. On these segments Vision Based Page Segmentation (VIPS) algorithm is applied to get blocks in a very refined manner.

This paper is organized as follows: The related work has been discussed in section Related Work. In section Background, background details are overviewed. Further in next section, the proposed hybrid web page segmentation method has been detailed out. Block extraction algorithm is presented in next section. Finally we draw conclusions and present our future work in Conclusion and Future Work section.

RELATED WORK
DOM and VIPS are two basic methods, which are analyzed and discussed for web page segmentation. Other than this, there are techniques developed on the basis of structure, content, vision or HTML source code.

One of the ways of page segmentation is based on the layout of a web page. For example, a web page can be separated into 5 regions: top, down, left, right and centre. But the drawback of this method is that such a layout template can not be suggested for all type of web pages. Furthermore, such segmentation method will be too rough to illustrate connection among the regions. However, there are few algorithms that make use of content or link information in order to segment a web Page. Embley [3] discovered record boundaries within a page using heuristic rules. Bar-Yossef [11] solved the template detection problem and helped in its various applications.

Other than these two kinds of methods, many researchers have considered web page division, based on the type of tags. Some of the applicable tags include <P> (paragraph), <Table> (table), <H> (Heading), etc. Lin [5] discovered informative contents of a web page by calculating entropy value of each feature and distinguish informative content blocks from it.

Kaasinen [4] separates a web page using tags such as <P> and <Table> for adaptive content delivery. VIPS [2] proposed by D.Cai is a page segmentation algorithm based on html visual clues. This paper presents an automatic top-down, tag-tree independent approach to detect web content structure.

BACKGROUND
The Document Object Model (DOM) provides a standardized set of objects that define the structure of HTML and XML documents (Robie 1998[1]). The DOM is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. In order to build a DOM tree from a document, a user agent must first parse the source HTML or XML file. This involves reading all of the tags and text and constructing the node structure based on the contents of each tag. Once the tree is in memory, it can be modified by the application or used for other purposes such as information extraction or rendering of the document. The logical structure of a DOM has a hierarchical nature; hence, it resembles a tree. But, as Robie (1998) states, the DOM does not require documents to be implemented as a tree.

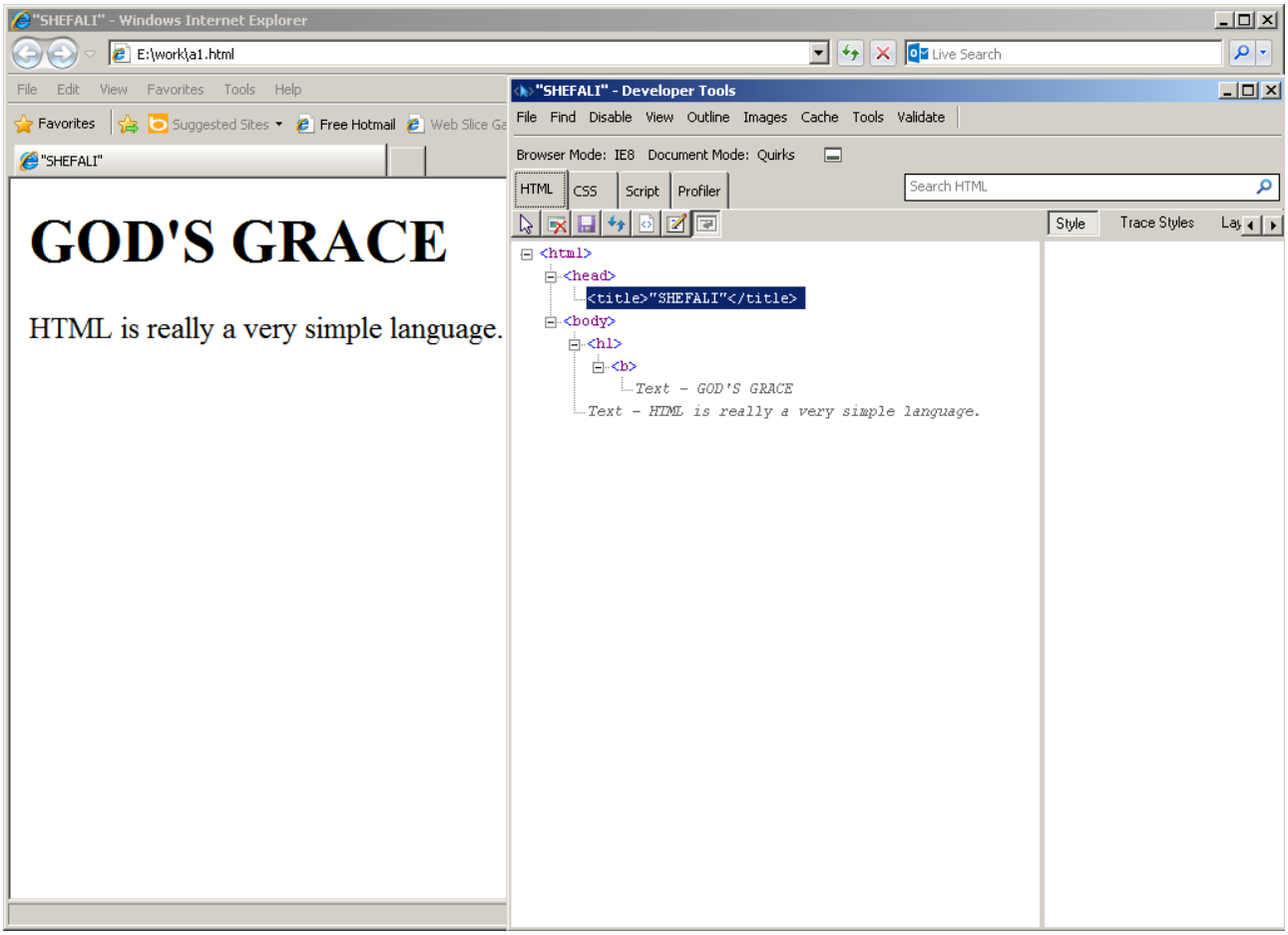

Fig. 1 The background window is HTML document which is rendered in explorer and the front window on right side shows Web Developer Toolbar which displays much of the information as DOM Inspector

Another approach is vision based page segmentation (VIPS) algorithm [2], which utilizes the fact that semantically related contents are often grouped together. Using this concept the entire page is divided into different regions using implicit or explicit visual separators such as images, lines, font sizes, blank areas, etc.

HYBRID WEB PAGE SEGMENTATION

Dom Tree Extractor

In order to analyze a web page for content extraction, the web page is first passed through an HTML parser which corrects the HTML code for any syntactic error and creates informative items of web page in the form of tokens. These tokens are ready to get arranged like a tree. Here the term "tree" is used when referring to the arrangement of those tokens which can be reached by using "tree-walking" methods. DOM technique is applied on these tokens which result into DOM tree. The nodes of the tree represent the various types of content in a document. This process accomplishes the steps of structural analysis and structural decomposition of the web page.

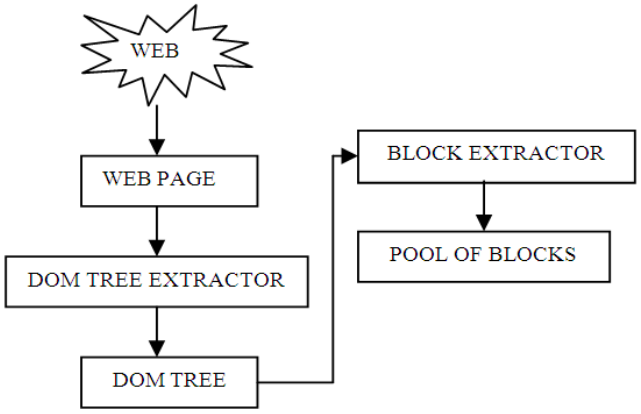

Fig. 2 System Architecture

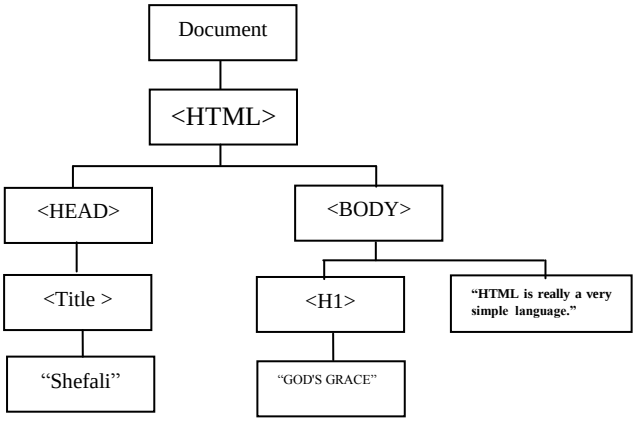

Fig. 3 DOM Tree of HTML Page shown in fig. 1

Block Extraction

In the sense of human perception, it is always the case that people view a web page as different semantic objects rather than a single object. Actually, when a web page is presented to the user, the spatial and visual clues can help the user to unconsciously divide the web page into several semantic parts. Therefore, it is possible to automatically segment the web page by using the spatial and visual clues.

DOM tree nodes require more refinement in order to create blocks. Block is the leaf node of the resulting tree which cannot be further divided and it purely segments the HTML source code.

The proposed algorithm makes full use of page layout feature: it first extracts all the suitable nodes from the html DOM tree. Each node of the Dom tree is segmented and analyzed on the basis of certain rules in order to create single unit blocks from it. For each big block, the same segmentation process is carried out recursively until sufficiently small blocks are resulted, which satisfies all the rules and can't be further divided.

Rules to be followed in the block extraction phase:

1) Divide a node if the DOM node has only one valid child and the child is not a text node

2) Divide a DOM node if one of the child nodes of the DOM node is line-break node.

3) Divide a DOM node if one of the child nodes of the DOM node has HTML tag <HR>.

4) Divide a DOM node if the background color of this node is different from one of its children's, but the child node with different background color will not be divided in this round.

Pool Creation

Once the blocks have been extracted, they all are collected in a single queue, which we say as pool of blocks. Now it is easier to search for blocks and to further work on them.

Block creation algorithm.

After discussing system architecture, it is clear that one has to start from construction of a DOM tree for the web page by using the proposed method given by Jing Wang [6].

Here, the function definition of SourceHtmlToBlock() contains two parameters as input. Node parameter is a particular node of previously constructed DOM tree at a particular level and Subtree parameter indicates the sub-tree below the selected node in DOM tree. This node contains a part of web page, segmented on the basis of Document Object Model (DOM). These two parameters are passed through the converter() function, which converts the part of web page contained in Node into its HTML source code, and stores it in another character variable, Sourcecode. Further a condition is applied on the selected node that if a node has a child then check further conditions of which if any one is true then divide the node else declare the node as indivisible unit and give the name of a block and store it. Those sub conditions are: child is not a text node or child is a line break node or child has <HR>tag or child has different background color. Finally it returns a pool of blocks.

```
Algorithm SourceHtmlToBlock (Node, Subtree)
{
Sourcecode =converter (node, Subtree);
If (node has a child)
{
   If (child is not a text node || child is a line break node || child has <HR>tag || child has different bgcolor)
            And put them into a queue then
           Divide the node for each child;
   }
Else
Declare the node as a block and store it;
}
```

Fig. 4 Algorithm for Block Extraction

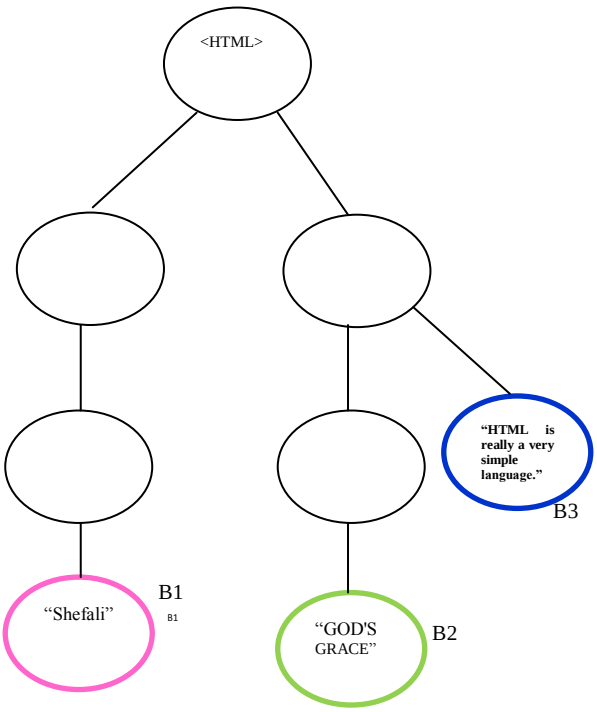

Fig. 5 Resulting blocks after applying proposed algorithm

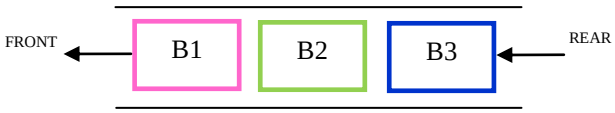

Fig. 6 A queue representing pool of blocks

CONCLUSION AND FUTURE WORK

In this paper a block extraction algorithm has been proposed which processes each node of the DOM tree of the web page to extract a block or segment which cannot be further divided. It separates the relevant information's and assembles coherent regions at one place. On comparison with the existing methods, proposed method considers both the structural and visual aspects of a web page which led to a good result. However, more visual clues can be added in this algorithm as it is containing very few constraints during the division process.

These extracted blocks can work as boon for many applications. For example, it can be used by browsers to extract relevant information and assemble similar content blocks at one place. It can be also helpful for web access in mobile phones where users are fed up with extra banners and widgets. It is also useful for researches done in the area of clustering. So this work can be further extended to prove other results.

REFERENCES

[1] Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson, Lauren Wood, Document Object Model (DOM) Technical Reports, In Proceedings of W3C Recommendation, 1 October 1998.

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma, VIPS: a Vision-based Page Segmentation Algorithm, In Proceedings of Microsoft Research, Microsoft CorporationOne Microsoft Way Redmond, WA 98052 Nov. 1, 2003.

[3] Embley, D. W., Jiang, Y., and Ng, Y.-K., Record-boundary discovery in Web documents, In Proceedings of the 1999 ACM SIGMOD international conference on Management of data, Philadelphia PA, 1999, pp. 467-478.

[4] Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S., and Laakko, T., Two Approaches to Bringing Internet Services to WAP Devices, In Proceedings of 9th International World-Wide Web Conference, 2000, pp. 231-246.

[5] Lin, S.-H. and Ho, J.-M., Discovering Informative Content Blocks from Web Documents, In Proceedings of ACM SIGKDD'02, 2002.

[6] Fangju Wang, Jing Li, Hooman Homayounfar, A space efficient XML DOM parser, Department of Computing and Information Science, University of Guelph, Guelph, Ont., Canada, Jan. 2007.

[7] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[8] J. Breckling, Ed., the Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[9] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[10] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.

[11] Bar-Yossef, Z. and Rajagopalan, S., Template Detection via Data Mining and its Applications, In Proceedings of the 11th International World Wide Web Conference (WWW2002), 2002.

[12] Rahman, A., Alam, H., and Hartono, R., Content Extraction from HTML Documents, In Proceedings of the First International Workshop on Web Document Analysis (WDA2001), 2001.