ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

World Wide Web: Web Page Segmentation

Bosubabu Sambana, MCA, M.Tech
Assistant Professor,
Dept of Computer Science & Engineering,
Simhadhri Engineering College, Visakhapatnam, AP-531001, India.

Abstract:

The World Wide Web contains more and more information each day and it is highly demanded, that search engine shows the most relevant result available. But it has to overcome a great amount of irrelevant information. This paper summarizes effort focused on filtering the content of web pages by segmenting the page to smaller pieces and filtering the irrelevant ones. This research area in a very important part of wider effort in content extraction, not just for its benefits to search engines, but also for ability to transform the content e.g. for special devices.

Keywords:
TF-IDF, World wide Web, Semantic Relatedness, Keyword Extraction, Word net,

1. Introduction:

For last several years there is a growing trend in sharing a vast amount of information on the web. However with all this information new problem arises. Parsing and indexing it becomes a serious challenge. One of the main issues on the web is that the information contained on web sites is often mixed with irrelevant content, which might mislead automatic crawlers on the semantics of the page. A family of methods focused on how to deal with this issue has been developed and this paper brings summary of different approaches these methods are based on. The goal of these algorithms is **to** find segments of the web page, which contain the relevant information. They receive data and meta data of a single web page as input and they output a content structure, usually in a form of semantic tree.The advantage of this family of methods is that they are designed to work on a single page (unlike Template detection methods). That also implies the main disadvantage – scaling. Especially for visual methods processing a large number of web pages can become a serious problem.

2. BACKGROUND:

2.1. World Wide Web: World wide Web (later renamed to **Nexus** to avoid confusion between the software and the World Wide Web) is the first web browser. and editor and now discontinued. When it was written, World Wide Web was the only way to view the Web.

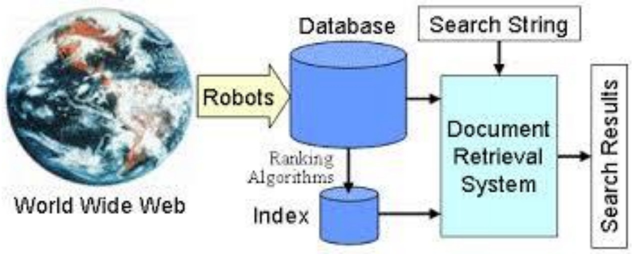


Figure 1: Basic View of World Wide Web

2.2. INTERNET:

The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. Information that travels over the Internet does so via a variety of languages known as protocols.

Figure 2: Internet

2.3 Security : Information Security Threats are using the following

a) Internet Cryptography Techniques
b) Transport Layer Security
c) Application Layer Security
d) Server Proxies and Firewalls
e) Week resources in web sites/Hosts

2.4 Authentication Protocol: a legal user can access its own data fields, only the authorized partial or entire data fields can be identified by the legal user, and any forged or tampered data fields cannot deceive the legal user.

3. RELATEDWORK:

3.1. Literature survey

It involved many researching previous studies that were conducted in the area of

3.2 VISUAL PAGE SEGMENTATION:

This family of methods is based on an approach with simple concept, but rather complex computing demands. The concept is to segment the web page as a user would segment it if he was looking at it. Some methods in this family even rely on user-assisted learning. Those methods however are not the subject of this paper, we will focus only on fully automatic ones. These have to simulate user view of the web page, which means a page has to be rendered either to an actual picture or at least to some kind of relevant internal representation of the visual information contained on the web page.

3. 2.1 VIPS ALGORITHM:

Vision-based Page Segmentation algorithm (VIPS) is a basic algorithm in a family of visual based methods [3].The algorithm uses one essential term: Degree of Coherence or simply DoC It is a measure of visual coherence defined for each block. It can be represented by any number (integer or real), but it must grow with visual consistency of the block. Also a parent can never have greater DoC than its children in block hierarchy tree.

The algorithm segments page in three steps:

(1) Extracting visual blocks,

(2) Detecting separators between extracted blocks and

(3) Detecting content structure based on results of previous two steps.

Extracting visual blocks consists of a top-down tree walking through the DOM tree1. The walk is iterative – in each iteration a new node representing visual block is detected in the DOM tree. After this detection a decision is made (based on certain properties like color, size . . .). Whether the block shall be recursively segmented further or not. For each detected block which isn’t segmented further a DoC is set according to its visual coherence. The second step is separator detection. Separator is defined as horizontal or vertical line or rather rectangular area which doesn’t intersect any of previously detected blocks.

The algorithm is initialized by a single separator covering the whole page. Separators are always detected for a particular level of visual block tree. Then for each block we perform a detection of its relation to each existing separator**:** if the block is fully covered by area of a separator, divide this separator into two if the block partially intersects a separator, shrink that separator, so the intersection is eliminated if the block fully intersects with a separator (i.e. covers entire height of a horizontal separator or entire width of vertical separator), remove the separator entirely The algorithm is finished by removing separators at the edges of document.

After we have all separators, we assign the weights based on visual difference of adjacent blocks. Note that this algorithm produces either only vertical or only horizontal separators. However the page has to be segmented in both directions. Here it is important to realize that the whole algorithm is done recursively2. That means that the page can be separated only in one direction, but any of its child nodes can be separated on the other direction.The final step of VIPS is content structure construction. In this step we iterate through a list of pre-piously found separators and merge visual blocks adjacent to them. It's important to merge blocks adjacent to separators with the smallest weight first. Before merging we have to check whether blocks meet granularity requirement. If they do, there is no need for merging them. The granularity requirement is of course represented by PDoC and the general rule for meeting the granularity requirement is that DoC > PDoC. If the block doesn't meet the requirement, we return to step one with root node being that visual block.

This algorithm shows overall better results than its predecessors, but it has some shortcomings as well. Two were described in : In some cases direct division of a visual block is impossible and utilization of virtual blocks is required. This can have negative impact on further processing, because blocks are not really present in the document. Resulting tree represents page segmentation but some information such as mutual position of blocks is missing. That information might be useful for improving algorithm's results.

- Of course additional visual information such as background color or font properties is required.
- In previous paragraph we stated that each of the detected blocks can be recursively segmented further – in that case the algorithm is invoked from the beginning, but the root node in the new run is replaced by node corresponding to the segmented block

3.3. OTHER VISUAL ALGORITHMS:

Burget and Rudolfová in introduced a bottom-up method based on some concepts introduced by VIPS. The algorithm he described has a goal to deal with VIPS shortcomings described in section 2.1. It creates a tree of visual areas in four steps:

(1) Creating temporary tree of boxes,

(2) Finding boxes which represent standalone visual areas,

(3) Detecting continuous areas and

(4) Finding significant areas.

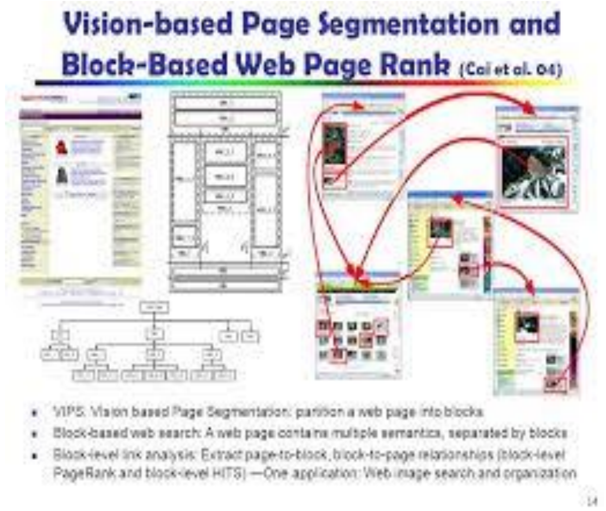
The box is a basic compound of the web page – it is defined as a rectangular area with defined position and width/height and containing either another set of boxes or a content of the web page. The first step creates a tree of these boxes. First we need to identify them, then we can build the tree. The tree basically defines nesting of the boxes in other boxes. There are three possibilities of a relation between two boxes:

A) Boxes don't intersect: they are not related.
B) One box is completely nested in another: the larger one is considered a parent.

Boxes have a partial intersection:

This is detected by mutual position of boxes' corners. The parent box is then extended to contain the whole child box. The whole page is representing a root node of the box tree. In the tree only two rules apply: parent box contains all child boxes and all child nodes on the same level of the tree mustn't have intersection with each other. In the second step we create a tree of visual areas by merging visually same boxes from step one. The final tree of visual areas corresponds with the tree of visual boxes, but its nodes have the above described feature – all are visually distinct from each other.The third step brings more merging. This time visually similar nodes (e.g. adjacent paragraphs of text) are merged into one continuous block.

In this step information about mutual block position is required. In the fourth step we look for significant areas by finding optical separators like borders, lines or just big spaces between blocks. This method has couple advantages over VIPS. One of them is particularly useful for our purposes – it is strictly focused on visual information rather than the DOM tree, which can bring better results e.g. for absolute positioning on the web page.



Vision-based Page Segmentation and Block-Based Web Page Rank (Cai et al. 04)

- VIPS: Vision based Page Segmentation: partition a web page into blocks
- Block-based web search: A web page contains multiple semantics, separated by blocks
- Block-level link analysis: Extract page-to-block, block-to-page relationships (block-level PageRank and block-level HITS) —One application: Web image search and organization

Fgure.3

4. Existing System:

HTML BASED SEGMENTATION METHODS:

These methods are based on analyzing web page without the need for rendering it. That means selected approach is either based on inspecting the HTML code directly or (more often) traversing the DOM tree and evaluating information gathered from it. They are usually much faster than visual-based methods. Quality and speed of these methods is given by used heuristics. Below are described some examples of algorithms which illustrate what heuristics can be used. [1]

3.1 TEXT BASED:

DOM or Document Object Model is a tree structure representing a nesting of elements within the page and it is often used for traversing the web page. Method introduced in [4] uses very crude heuristics. The segmentation part of the method is unusable nowadays, because it considers table data cells as content blocks (modern web pages don’t use table layout). For this reason we shall omit it and focus on classification part of the method. The method classifies segments into two categories:

A) Informational and
B) Redundant.

The entropy of content features (represented by meaningful words) is used as heuristics. A matrix of all features and their occurrence on all pages (in all content blocks) is created and the probability of particular feature being in particular content block is calculated. The final step is to calculate entropy of the block from all of its features. If the entropy reaches threshold level, the block is considered redundant. Otherwise it is considered informative. There are some other methods utilizing features of the text itself and building heuristics on them. An example of such can be a method proposed by Laber et al. [5].

The method is based on calculating link density and statistical F1 score of all blocks. The basic assumption is that for every page there is one block with high link density and high F1 – this block is then marked as the main content.The algorithm itself just performs DFS traversal of the DOM tree and it is trying to find a leaf node which contains defined amount of characters and has an ancestor which passes conditions defined in [5] (text amount, link density, non-presence of siblings with similar properties).

3.2 DOM BASED:

The last group of segmentation methods is based on general traversal of the DOM tree and identifying the content with usage of various heuristics. Some of related works might not even be directly solving the segmentation by traversing the DOM tree.

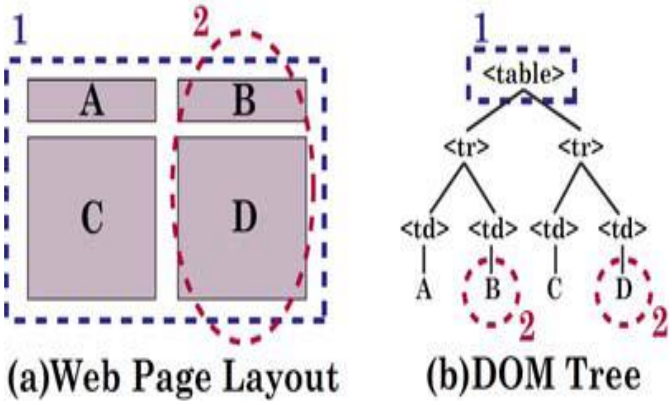


Figure.4: Problem of DOM-based segmentation.

5. PROPOSED WORK:

The main objective of this work is to propose a solve using this algorithms and the proposed VIPS algorithm takes advantage of visual cues to obtain the vision-based content structure of a web page and thus successfully bridges the gap between the DOM structure and the semantic structure. The page is partitioned based on visual separators and structured as a hierarchy. This semantic hierarchy is consistent with human perception to some extent. VIPS is also very efficient. Since we trace down the DOM structure for visual block extraction and do not analyze every basic DOM node, the algorithm is totally top-down.

WISH algorithm:

In [2] Hong et al. introduce their technique for traversing the DOM tree and selecting relevant content. They target so called data records which are pieces of a web page which are repeating themselves, but with a different content. An example of such record can be a category or search results listing. Their algorithm is divided in several steps. In the first step, they extract content candidate nodes using BFS-based algorithm. Data records are defined as tags on the same level of DOM tree, containing repetitive children sequences and having similar parent. The parent is denoted as data region. In case no nodes on a particular level of BFS satisfy the definition, the next tree level is inspected. Output of the first stage is a list of all data regions identified on the page. Other stages only filter results the algorithm gained in the first stage. Following observations are used for filtering heuristics:

- Relative to the whole page, data records have large size
- Data Records are usually repeated more than three times on a page
- A regular expression can be devised for description of data record. Since all data records share the same template, it will apply on all of them
- Data Records usually consist of a small amount of HTML tags

After the list of data regions is filtered, every data region has to be assigned its relevancy score. The scoring function described in [6] determines the size of area taken by data records by counting characters and images each data record contain. Elements representing free space are taken into account as well. Data region with the best score is considered to be the main content of the page. This algorithm is the best example of how can different heuristics be used for page segmentation and classification.

Iterated dividing and shrinking:

Based on web pages features, our algorithm is composed of three steps – web to image, image pre-processing and dividing.

Features of Web page:

By analyzing the visual characteristics of web page, it is easy to find out that there are; following two general features.

- Web page is composed by blocks; every visible element in HTML is displayed in its rectangle area. Labels, images, tables even flashes are all possessed of the basic parameters: width and length. So with a rectangle area, we can locate any visible element in web pages.

- The visible elements of web page are separated by background space; If there is no space between two visible elements, they may be considered as one by people, which does not handicap our understanding from visual aspect. Based on those two features, we can divide the web page into sub-images (basic blocks) according to the dividing zones which are the spaces between blocks.

Web page pre-processing:

First, web pages are converted into images, and then we get the binary images from detecting the edges of these web images. The dominant browsers such as IE and Firefox provide the APIs to save web page as image. Also there are several commercial programs like View Page (http://viewpage.maxthon.com/) Html to jpg that can do the same job. In this paper, we make use of the system library function (Microsoft.mshtml.dll) to fulfill convert from html to image. This system library is used to parse the HTML, and is employed by IE, Outlook.

The initial web images are always complex and usually in RGB format, so we will simplify them by boundary detecting. Canny is employed for its validity. Compared with Roberts, Prewitt and Sobel, Canny algorithm enjoys the virtues of higher positioning precision and better signal to noise ratio. From the results of experiments. we can find that the Canny algorithm is much more suit for web page image preprocessing. The result of the pre-processing is a binary image with 0 for background and 1for boundary.

Dividing and shrinking:

The sub-images are used to locate the blocks in web page. Based on features of web page we have summarized in , any element in a web page can be constricted in a rectangle area that is represented by a sub-image.

VIPS-Algorithms Experiments:

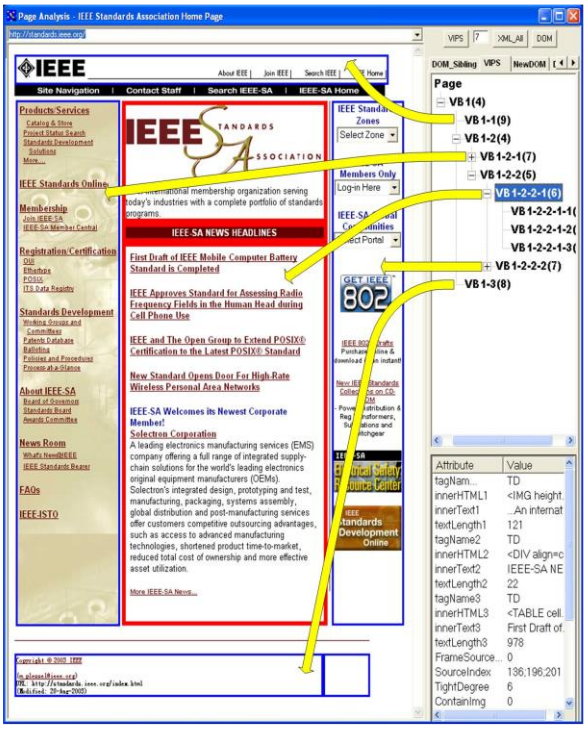
In this section, we will first show two example pages, comparing our VIPS result and the DOM tree structure. Then we provide some performance evaluation of our proposed VIPS algorithm based on a large collection of web pages from Yahoo. We also conduct experiments to evaluate how the algorithm can be used to enhance information retrieval on the Web.

Simple examples of web page segmentation:

In this sub-section, several simple examples of web page segmentation are presented to give people an intuition how our VIPS algorithm works. In the meantime, we show the DOM tree of these pages. We can clearly find that we cannot get the right structure of the blocks only based on naïve DOM tree. Moreover, it is hard for us to decide which node conveys semantic meanings and where we should stop in different applications.

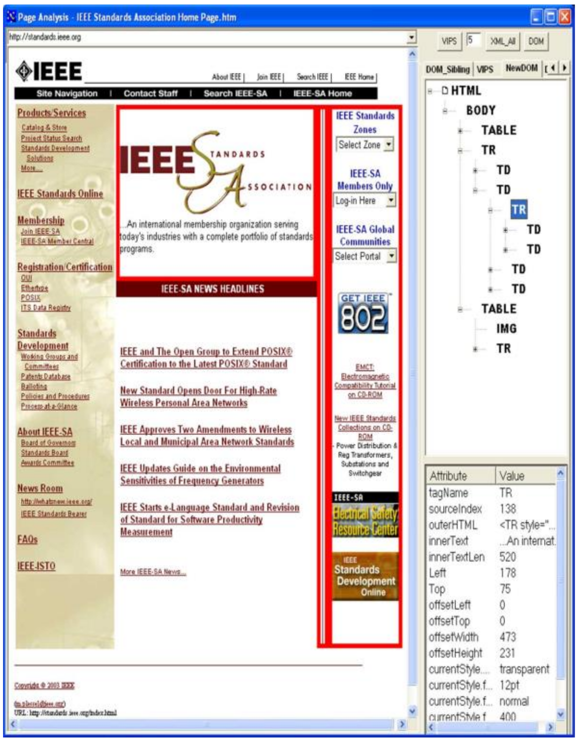
Figure.5 shows our VIPS result on a sample page1. The left part shows the page with different regions (different visual blocks in VIPS algorithms) marked with rectangles. The block marked with red rectangle is the block VB1-2-2-1. The upper right part shows the vision-based content structure of this page, while the lower one shows some statistics of selected visual block. From the right VIPS tree, we can know the hierarchical structure of different blocks. In different applications, we can control the partition granularity by setting PDoC, as shown in the northeast corner.

The DoC value of each block is shown behind the node name (in parenthesis) in the right VIPS tree. Although from page layout we see that VB1-2-1, VB1-2-2-1 and VB1-2-2-1 are parallel, in our heuristic rules the separator between VB1-2-1 and VB1-2-2-1 will get higher weight than the separator between VB1-2-2-1 and VB1-2-2-1 because of the different background colors. So VB1-2-2-1 and VB1-2-2-2 will be merged to VB1-2-2, paralleled to VB1-2-1.[2]



Attribute	Value
tagNam...	TD
innerHTML1	<IMG height.
innerText1	...An internat
textLength1	121
tagName2	TD
innerHTML2	<DIV align=c
innerText2	IEEE-SA NE
textLength2	22
tagName3	TD
innerHTML3	<TABLE cell.
innerText3	First Draft of.
textLength3	978
FrameSource...	0
SourceIndex	136;196;201
TightDegree	6
ContainImg	0

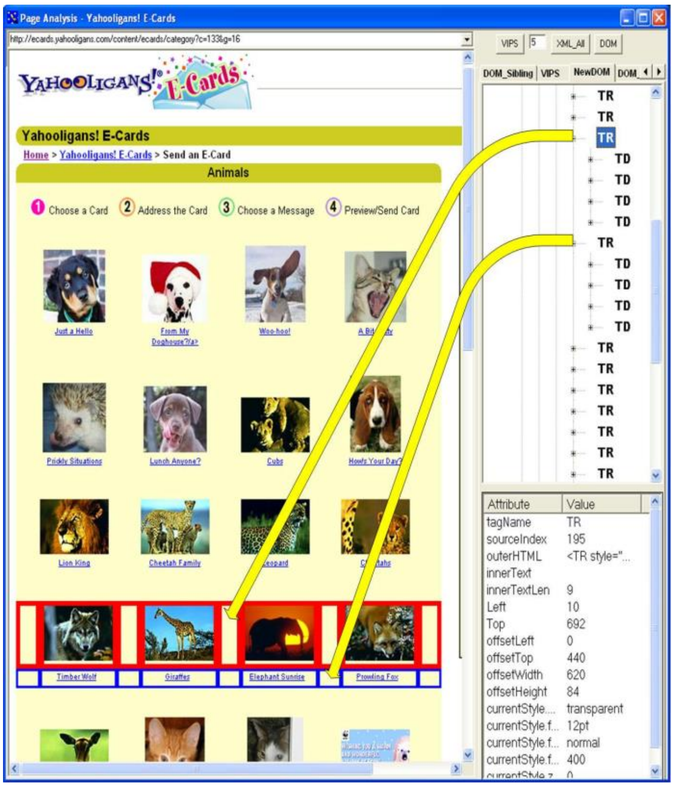
Figure.5: VIPS segmentation result of an IEEE page



Attribute	Value
tagName	TR
sourceIndex	138
outerHTML	<TR style="...
innerText	...An internat.
innerTextLen	520
Left	178
Top	75
offsetLeft	0
offsetTop	0
offsetWidth	473
offsetHeight	231
currentStyle....	transparent
currentStyle.f...	12pt
currentStyle.f...	normal
currentStyle.f	400

Figure .6: DOM tree Partition result of an IEEE page

For comparison, we show the DOM tree and its corresponding blocks in Figure 6.We can see that the area marked with red line is a <TR> node (with three <TD> children). From visual perspective, these <TD> nodes should not be grouped together, but we cannot get this information from DOM tree structure, while in VIPS this problem can be solved with the spatial and visual information of these blocks. We got the right content structure using our VIPS algorithm. Take another example web page1.

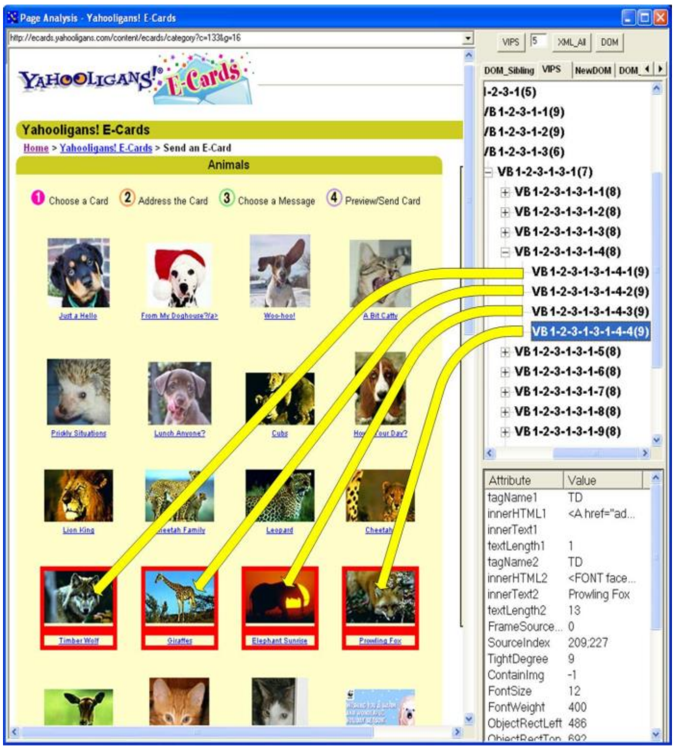


Attribute	Value
tagName	TR
sourceIndex	195
outerHTML	<TR style="...
innerText	
innerTextLen	9
Left	10
Top	692
offsetLeft	0
offsetTop	440
offsetWidth	620
offsetHeight	84
currentStyle....	transparent
currentStyle.f...	12pt
currentStyle.f...	normal
currentStyle.f...	400
currentStyle.z	0

Figure 7: DOM tree struct

We show the DOM tree structure in Figure-7 and VIPS segmentation result in Figure-8.For comparison, we show the DOM tree and its corresponding blocks in Figure 6.We can see that the area marked with red line is a <TR> node (with three <TD> children). From visual perspective, these <TD> nodes should not be grouped together, but we cannot get this information from DOM tree structure, while in VIPS this problem can be solved with the spatial and visual information of these blocks.

We got the right content structure using our VIPS algorithm. Take another example web page1. We show the DOM tree structure in Figure 7 and VIPS segmentation result in Figure 8. In DOM tree structure, the images and the texts are belonging to different <TR> nodes. It is hard to decide the exact description text of the image. VIPS result clearly reveals the semantic relationship of the images and their surrounding texts.



```
-2-3-1(5)
/B 1-2-3-1-1(9)
/B 1-2-3-1-2(9)
/B 1-2-3-1-3(6)
⊟ VB 1-2-3-1-3-1(7)
    ⊞ VB 1-2-3-1-3-1-1(8)
    ⊞ VB 1-2-3-1-3-1-2(8)
    ⊞ VB 1-2-3-1-3-1-3(8)
    ⊟ VB 1-2-3-1-3-1-4(8)
        VB 1-2-3-1-3-1-4-1(9)
        VB 1-2-3-1-3-1-4-2(9)
        VB 1-2-3-1-3-1-4-3(9)
        VB 1-2-3-1-3-1-4-4(9)
    ⊞ VB 1-2-3-1-3-1-5(8)
    ⊞ VB 1-2-3-1-3-1-6(8)
    ⊞ VB 1-2-3-1-3-1-7(8)
    ⊞ VB 1-2-3-1-3-1-8(8)
    ⊞ VB 1-2-3-1-3-1-9(8)
```

Attribute	Value
tagName1	TD
innerHTML1	<A href="ad...
innerText1	
textLength1	1
tagName2	TD
innerHTML2	<FONT face...
innerText2	Prowling Fox
textLength2	13
FrameSource...	0
SourceIndex	209;227
TightDegree	9
ContainImg	-1
FontSize	12
FontWeight	400
ObjectRectLeft	486
ObjectRectTop	692

Figure .8: VIPS Partition result

We can use these surrounding texts to represent the images and used this text representation in a web image search system. From these examples, we can see that our VIPS algorithm can successfully identify the relationships among different blocks in the web page, while DOM structure fails. Moreover, VIPS algorithm assigns a DoC value to each node in vision-based content structure, which is critical in determining where to stop in different applications.

Performance of VIPS algorithm:

To evaluate the performance of VIPS algorithm, we select 600 web pages from popular sites listed in 14 main categories of Yahoo! directory (http://www.yahoo.com). The VIPS algorithm is run on all the pages and the results are assessed by several individuals. Five volunteers are asked to judge the results. Table 3 shows the result.

Human judgment	user1	user2	user3	user4	user5	All
Perfect	299	307	374	310	377	1667
Satisfactory	231	266	193	237	197	1124
Fair	64	23	29	48	20	184
Bad	6	4	4	5	6	25

Table 1: valuation of the VIPS algorithm

As can be seen, 1667+1124=2791 (93%) pages have their semantic content structures correctly detected. For those “fair” pages, the hierarchies are incorrectly constructed because they contain insufficient visual cues to separate blocks while people get the structure by understanding the semantic meaning of the blocks. For the “bad” cases, one major reason is that the browser (i.e. Internet Explorer in our experiments) provides wrong position information so that our algorithm cannot get the correct content structure. Another reason is that, for several pages the images (e.g., a very thin image representing a line) are used to divide different content blocks. Our algorithm currently cannot handle this situation.

Experiments on web information retrieval:

Query expansion is an efficient way to improve the performance of information retrieval. The quality of expansion terms is heavily affected by the top-ranked documents. Noises and multi-topics are the two major negative factors for expansion term selection in the web context. Since our VIPS algorithm an group semantically related content into a block, the term correlations within a segment will be much higher than those in other parts of a web page. With improved term correlations, high-quality expansion terms can be extracted from segments and used to improve information retrieval performance.

We choose Okapi as the retrieval system and WT10g in TREC-9 and TREC 2001 Web Tracks as the data set. WT10g contains 1.69 million pages and amounts to about 10G. The 50 queries from TREC 2001 Web Track are used as the query set and only the TOPIC field for retrieval, and use Okapi's BM2500 as the weight function and set $k1 = 1.2$, $k3 = 1000$, $b = 0.75$, and $avdl = 61200$. The baseline is 16.55% in our experiments. An initial list of ranked web pages is obtained by using any traditional information retrieval methods. Then we apply different page segmentation algorithms (including our VIPS algorithm with PDoC (6) and a naïve DOM-based approach) to the top 80 pages and get the set of candidate segments. The most relevant (e.g. top 20) segments from this candidate set are used to select expansion terms. These selected terms are used to construct a new expanded query to retrieve the final results.We compared our method with the traditional pseudo-relevance feedback algorithm using whole document and a naïve segmentation method based on DOM tree, which are briefly described below:

- Our Vision-based approach (denoted as VIPS): The PDoC is set to 6. To reduce the effect of tiny blocks, blocks less than 10 words are removed. The top 80 pages returned by the initial retrieval phase are segmented to form the candidate segment set.
- Simple DOM-based approach (denoted as DOMPS): We iterate the DOM tree for some structural tags such as TITLE, P, TABLE, UL and H1~H6. If there are no more structural tags within the current structural tag, a block is constructed and identified by this tag. Free text between two tags is also treated as a special block. Similar to VIPS, tiny blocks less than 10 words are also removed, and the candidate segments are chosen from the top 80 pages returned by the initial retrieval phase.
- Traditional full document approach (denoted as FULLDOC): The traditional pseudo-relevance feedback based on the whole web page is implemented for a comparison purpose. The experimental result is shown in Table 4 and Figure 14.

Number of Segments	Baseline (%)	FULLDOC (%)	DOMPS (%)	VIPS (%)
3	16.55	17.56 (+6.10)	17.94 (+8.40)	18.01 (+8.82)
5		17.46 (+5.50)	18.15 (+9.67)	19.39 (+17.16)
10		**19.10** **(+15.41)**	18.05 (+9.06)	19.92 (+20.36)
20		17.89 (+8.10)	19.24 (+16.25)	20.98 (+26.77)
30		17.40 (+5.14)	19.32(+16.74)	19.68
40		15.50 (-6.34)	19.57 (+18.25)	17.24 (+4.17)
50		13.82 (-16.50)	**19.67** **(+18.85)**	16.63 (+0.48)

60		14.40 (-12.99)	18.58 (+12.27)	16.37 (-1.09)

Table 2: Performance Comparison of Query Expansion Using Different Page Segmentation Methods.

As can be seen, the average retrieval precision can be improved after partitioning pages into blocks, no matter which segmentation algorithm is used. In the case of FULLDOC, the maximal average precision is 19.10% when the top 10 documents are used to expand the query. DOMPS obtains 19.67% when the top 50 blocks are used, a little better than FULLDOC. VIPS gets the best result 20.98% when the top 20 blocks are used and achieves 26.77% improvement. Document based query expansion FULLDOC uses all the terms within the top documents for expansion. Since the baseline is very low, many of top ranked documents are actually irrelevant and there are many terms coming from irrelevant topics. These cause the retrieval performance relatively low although better than the baseline. For the same reason, the average precision drops quickly.

DOM based approach DOMPS does not obtain a significant improvement compared with FULLDOC, partly because the segmentation is too detailed. The segments are usually too short to cover complete information about a single semantic. This is a major limitation of segmentation based on naïve DOM tree which we addressed before. In many cases, good expansion terms are within the previous or proceeding blocks, but are missed because those blocks are not ranked high enough to be selected in pseudo-relevance feedback. Compared with DOMPS, our VIPS algorithm considers more visual information and is more likely to obtain a semantic partition of a web page. Therefore, better expansion terms can be extracted and better performance can be achieved. About 27% performance improvement on the Web Track dataset was achieved.

The experiments clearly show that vision-based web page content structure is very helpful to detect and filter out noisy and irrelevant information. Thus better expansion terms can be selected to improve retrieval performance.

6.RESEARCH WORK:

Future Research:

There are several areas which might be convenient for future research because there wasn't too much research done in them. The first such area is combination of visual algorithm with DOM and text based heuristics. Although all visual methods still work with a DOM tree (at least to some extent), they rely heavily on computed visual information. These algorithms can be accelerated by adding various non-visual heuristics. Their target would be to avoid (+18.91) computing of visual features, which is the most expensive part of visual segmentation algorithms.Another interesting area of research is improving results of visual segmentation by transforming DOM tree into a Semantic tree.

The HTML code and corresponding DOM tree is often very different from semantic structure of the page as user perceives it. The purpose of the transformation algorithm would be to design a new tree which would correspond with visual perception of the page. Of course the page would need to be rendered, then a semantic tree would need to be constructed and finally a minimal mapping between the DOM tree and the semantic tree would have to be found. Classification can be then performed on the semantic tree and by means of inverse mapping (from the semantic tree to the DOM tree) it can be applied to the page itself. The idea is that classification of nodes of semantic tree has a potential to be much more precise than the classification performed on the DOM tree.

7. CONCLUSION:

In This paper is an overview of distinct methods which can be used for finding a relevant content on the web page.

Each method has its advantages and disadvantages and their usage should be considered according to a particular task which needs to be solved. Many of presented algorithms were originally targeted at a detection of a content on news servers. But if we consider how modern web pages are designed, the same approach can be applied to blogs, CMS-based sites and also most of company web sites. New approach for extracting web content structure based on visual representation was proposed. The produced web content structure is very helpful for applications such as web adaptation, information retrieval and information extraction. By identifying the logic relationship of web content based on visual layout information, web content structure can effectively represent the semantic structure of the web page.

An automatic top-down, tag-tree independent and scalable algorithm to detect web content structure was presented. It simulates how a user understands the layout structure of a web page based on its visual representation. Compared with traditional DOM based segmentation method, our scheme utilizes useful visual cues to obtain a better partition of a page at the semantic level. It is also independent of physical realization and works well even when the physical structure is far different from visual presentation. The algorithm is evaluated manually on a large data set, and also used for selecting good expansion terms in a pseudo-relevance feedback process in web information retrieval, both of which achieve very satisfactory performance.

Acknowledgment:

This paper is Heartily Dedicated to my parents Sri.S.Dandasi & Smt.Janaki, My Wife Mrs.Suneetha bose and My life Inspirer Eminent Scientist Sri. Dr. A.P.J.Adbulkalam.

Authors' Profiles:

Bosubabu Sambana working with as an Assistant Professor in Simhadhri Engineering College, Visakhapatnam. He is completed Master of Computer Applications and Master Degree in Computer Science & Engineering from Jawaharlal Nehru Technological University – Kakinada, Pursing Master of Science in Mathematics, Andhra University, Andhra Pradesh, India. He has 4 years good teaching experience and having a good Knowledge on Space Research, Future Internet Architecture, Cloud Computing, Internet of Things/Services/Data, Computer Network and Hacking along with Computer Science Subjects. He is Published 4 Research Papers in various reputed International Journals and Magazines. He is the member of NASA, INTERNET SCOCIETY, W3C, MECS-PRESS, IAENG, IAAE and IJECSE.

References:

[1] A segmentation method for web page analysis using shrinking and dividing: http://www.tandfonline.com/action/journal Information?journal Code=gpaa20

[2] A Web Page Segmentation Method based on Page Layouts and Title Blocks: IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.10, October 2011

[3] VIPS:a Vision-based Page Segmentation Algorithm

[4] Images and Information– http://www.google.co.in/