VISUAL-BASED WEB PAGE ANALYSIS

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Kuang-Yao Lee
Spring 2014
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Kuang-Yao Lee:

Visual-Based Web Page Analysis

Alan Harrison, Chair
Department of Computer Science

Joseph Lewis
Department of Computer Science

Stephen Reed
Department of Psychology

March 25, 2014
Approval Date
ABSTRACT OF THE THESIS

Visual-Based Web Page Analysis
by
Kuong-Yao Lee
Master of Science in Computer Science
San Diego State University, 2014

This research investigates efforts to identify different content areas appearing on a webpage by comparing the visual features and the relative characteristics of each content area, called visual block in this study. The process is to use the Image Segmentation technique to extract and parse a webpage’s visual features, as well as analyze it to identify the functionality of each content area based on its layout and position.

To accomplish this, this study reviews several techniques that have been used in related fields and discusses the strengths and the weaknesses of these techniques. The main weakness for the past techniques is they rely heavily on HTML; in other words, they are language-dependent. This paper proposes a visual-based technique that focuses on using visual features rather than HTML; hence it is more language-independent. To determine the functionality of each visual block, the technique uses an algorithm to parse webpages into a tree structure and apply a rule of how humans determine the relationship between two objects on a 2D monitor.

The goal of this research is to design an automated visual-based algorithm to exam each visual block showing on the webpage and apply human cognitive processes to decide the role of each block. For example, one might wish to identify the main content, the sub content, the navigation menu, and the advertisement.

Chapter 1 describes the motivation, the issue, and possible solution to the problem. Chapter 2 reviews several different technologies that can be used to solve the problem and elucidates possible future research. Chapter 3 focuses on explaining how to prepare the test environment and techniques that have been used. Chapter 4 describes the result, what was accomplished, what was missing, and necessary further research. Chapter 5 concludes with the possibilities of this research and how future research might help accomplish the final goal of this research.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td>PAGE</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Purposes of the Research</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Limitation of the Study</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Issues</td>
<td>2</td>
</tr>
<tr>
<td>2 REVIEW OF LITERATURE</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Semantic Web</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Web Scraping</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Screen Scraping</td>
<td>7</td>
</tr>
<tr>
<td>2.4 Image Processing</td>
<td>7</td>
</tr>
<tr>
<td>2.4.1 Computer Vision</td>
<td>8</td>
</tr>
<tr>
<td>2.4.2 Optical Character Recognition</td>
<td>8</td>
</tr>
<tr>
<td>2.4.3 Edge Detection</td>
<td>9</td>
</tr>
<tr>
<td>2.4.4 Image Segmentation</td>
<td>10</td>
</tr>
<tr>
<td>2.5 Human Computer Interaction-HCI</td>
<td>11</td>
</tr>
<tr>
<td>2.6 Webpage Data Extraction</td>
<td>11</td>
</tr>
<tr>
<td>3 VISUAL BLOCK TREE AND VISUAL FEATURES</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Vision-Based Page Segmentation</td>
<td>15</td>
</tr>
<tr>
<td>3.2 Structure of the Visual Block</td>
<td>17</td>
</tr>
<tr>
<td>3.3 Structure of the Visual Block Tree</td>
<td>18</td>
</tr>
<tr>
<td>3.4 Visual Block Type Analysis Algorithm</td>
<td>20</td>
</tr>
<tr>
<td>3.5 Webpage Analysis Algorithm</td>
<td>22</td>
</tr>
<tr>
<td>3.5.1 Main Content</td>
<td>23</td>
</tr>
<tr>
<td>3.5.2 Advertisement</td>
<td>23</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 2.1. Summary of the Qualitative Analysis .................................................................13
Table 4.1. Visual Block Type Analysis Algorithm Variable Setup Values .......................26
Table 4.2. Webpage Analysis Algorithm Detail Test Result .............................................26
LIST OF FIGURES

PAGE

Figure 2.1. The Layers of the semantic web ................................................................. 5
Figure 2.2. Graphical perspective of the qualitative analysis ................................. 14
Figure 3.1. The vision–based page segmentation algorithm .................................... 16
Figure 3.2. The coordinate system of a webpage ..................................................... 17
Figure 3.3. Sample segmented webpage with the visual block and the separator .......... 19
Figure 3.4. The corresponding specification of Figure 3.3 ........................................ 19
Figure 3.5. The layout structure and vision-based content structure of Figure 3.3 .......... 19
CHAPTER 1

INTRODUCTION

1.1 Purposes of the Research

It has been decades since the World Wide Web was released, and finding information via websites has become a major daily activity for many people. People exchange information, ideas, and buy products through the web; however, as the web’s functionality grows, so does the complexity.

Today’s websites are so complicated that even those familiar with the web can get lost in a webpage. For example, people can have a difficult time finding the information they want on some webpages because there are too many links, ads, and pictures. It might be difficult, in other words, to find a particular link. Too, navigation becomes harder for websites with many advertisements. The result is that many people spend hours on the web to find particular information, but the same information could be retrieved in seconds if the computer is equipped with an algorithm to filter the webpage or to extract relevant data.

If we were able to request that a computer find particular data, the computer would then be able to filter out unnecessary information and target the data we need. This should help users save time as they seek information on the web. Also, it can help computers gather data on the Internet automatically.

Two things must be accomplished to achieve automated information extraction or filtration. First, render the webpage and parse it into a structure with enough information for the algorithm to use, which will require a web scraping technique and an image processing technique. These techniques will be reviewed in Chapter 2. Second, determine and apply rules based on the webpage design and human cognition, which will be discussed in Chapter 3.
1.2 LIMITATION OF THE STUDY

This research focuses on using visual features, hyperlink analysis, and webpage design principles to analyze visually the web content to identify different contents. It can also use HTML source code to help the process when needed.

This research does not analyze the graphical content in a webpage. For example, assuming there is an image file A showing on the left side of a webpage that is advertising for a drink: the algorithm will analyze the webpage layout visually and decide A’s functionality, but it will not decipher the kind of drink advertised or the relationship between A and the webpage.

Indeed, picture and video analysis may be difficult for this particular research, as it requires the program to understand not only the image but too the relationship between the image and the webpage. Despite these aspects transcending the scope of this research, they can be useful for further enhancement.

1.3 ISSUES

This research aspires to identify the content in a webpage. In order to accomplish this, a few issues need to be solved. For example, the following questions pose some of these issues: How can a computer understand the structure of a website? How can a computer tell the differences between an advertising picture and a picture of merchandise? Finally, how can a computer identify the important content in a webpage?

After some research, I found that web scraping can answer some of these questions and thus solve some of the problems these questions entail. Web scraping is a technique that typically uses regular expression to match HTML code and extract the target data in a webpage. Web scraping is one way to extract and identify information in a webpage, but with current interactive web technology, most content can change immediately, and hence decrease the efficacy of web scraping. Another difficulty involves the fact that the languages programmer uses to write the webpages, such as HTM, CSS, and JavaScript, change from time to time. There are many different versions already in use on the World Wide Web, and I believe the W3C (World Wide Web Consortium), the main international standards organization for the World Wide Web, will continue to introduce new versions in the future, so the number of different versions will only increase. In order to make these scraping
algorithms work for the majority of websites, these text-based algorithms would have to update constantly or even re-write much of the code every time a new version was released. One way to overcome this problem is via visual analysis, because the appearance of a webpage stays the same no matter what version of HTML or CSS a webpage uses. Hence the visual-based algorithm can ignore the different versions of the languages and analyze the webpage by its appearance. The webpages are designed to be used by humans, so it must follow basic design rules that humans can understand. If we can apply these rules to help, perhaps we can analyze the webpage independently from the HTML, CSS, and others.
CHAPTER 2

REVIEW OF LITERATURE

This chapter reviews technologies that should be used in this visual-based analysis. Some of these technologies may not be employed in this research because of the present limitations of my expertise, but will be required in future research to increase the usability of the visual-based analysis. I introduce these technologies to help readers understand the tools being developed that can enable computers to extract important information automatically. The first half of this section reviews some basic concepts and the second reviews the research paper that uses these techniques to analyze webpages.

2.1 SEMANTIC WEB

The main focus of this research is to be able to identify the content in a webpage; one way is through utilization of the Semantic Web concept. The Semantic Web is a network of data that enables machines to understand the meaning of information on the Internet. It uses metadata to describe the relation and meaning of the data. It then can be used by automated programs to use, share, and create more data. The term was coined by Tim Berners-Lee, the inventor of the World Wide Web and director of the W3C (World Wide Web Consortium).

According to Berners-Lee’s article, “The Semantic Web” [1], the main purpose of the Semantic Web is to create a understandable language for the Web so that a computer can use the Web’s already large amount of data to help humans perform daily tasks. Since the amount of the data is huge, many different languages are used to build several layers of description atop the current web languages. See Figure 2.1 [2].

These languages are commonly used today. Through these layers, computers can interchange data and collect related information performing closer to human cognition. See Appendix A for a detailed description on these languages.

The purpose of the Semantic Web is to render webpages machine-readable so that computers can use the information on the Internet automatically. This research is to use a visual-based algorithm to identify web content so computers can extract necessary data. The

The goal of these two approaches is similar, but the Semantic Web’s approach to re-design the web is not likely to be adapted by the world because re-writing all extant webpages would be a daunting task.

Moreover, the Semantic Web still has many problems, including the following:

- **Vastness:** The web contains billions of pages and is still growing fast. The programming languages used for these myriad pages differ greatly. The terminology, i.e., the different words, people have used on the webpages also numbers in the millions. Thus, automated system will need to process huge amounts of information.

- **Vagueness:** Human languages are difficult to understand because the same word can have different meanings under different circumstances. For the Semantic Web to work correctly, the automated system will not only need to understand the meaning of every word, concept, and graph, but also to understand the surrounding concepts, relationship between concepts, or even the current social environment. Fuzzy logic must be used here in order to deal with this problem.

- **Uncertainty:** Precise concepts may have uncertain values.

- **Inconsistency:** Logical contradictions can arise when developing large ontologies and when combining ontologies from separate sources. Defeasible reasoning and paraconsistent reasoning are two techniques that can be used here.

- **Deceit:** Deceit occurs when the programmer intentionally misleads the user. Cryptograph techniques are currently used to deal with this problem.
2.2 Web Scraping

This research uses web scraping to acquire data from the webpage, and this section will describe this technique in some detail. Web scraping, also called web harvesting or web data extraction, is a software program or technique used to extract information from a website. Web scraping usually uses a software program to simulate human surfing on the web. It is closely related to web indexing, which is used by most web search engines to index a website by using a bot. The difference is web scraping is used mainly for transforming unstructured websites into structured ones that can be stored and analyzed by an automated program. It is very useful for website change detection, web search, online price comparison, special data monitoring, and web data integration. This research uses automated programs to extract website appearance and to analyze it, one kind of web scraping technique. It is an actively developing field and shares a common goal with Semantic Web.

Currently there are many different levels of automation that consider as web scraping:

- **Human copy and paste**: This is the tradition human manual examination, which is manually finding the information on the web and copy it. It is the best solution for some situation where automatic web scraping software is being blocked by webpages.

- **Text searching and regular expression matching**: It uses a special computer command or regular expression to match the target text and extract certain the information. It is very basic technique but very useful and widely used.

- **HTML parsers**: Some query languages such as XQuery and the HTQL can be used to parse and retrieve plain HTML pages. It has very limited functionality when used on many dynamic webpages.

- **Vertical aggregation platforms**: Some companies have developed vertical harvesting systems that can create and monitor a multitude of web bots for specific websites. The process involves establishing a knowledge base for the entire vertical website links, and the system automatically creates the bots. This technique is typically used for the long tail of sites that common aggregators find complicated or too labor intensive to harvest the contents.

- **Http programming**: This technique uses HTTP requests to the web server to retrieve static or dynamic webpages and store them; this kind of programming can then analyze the stored webpages via other information extraction technique to harvest information.

- **DOM parsing**: It can retrieve the dynamic contents generated by the client side script by embedding the program into a web browser. This research uses this technique to get the DOM object from webpages, detailed in Chapter 3.
Most of these techniques are based on HTML, CSS, and text analysis, and the problem is that these approaches can be easily misled intentionally or unintentionally by programmers using a vague HTML tag or description in the HTML tag. That is why analysis of the final image is a better way to avoid these problems.

2.3 Screen Scraping

This research attempts to use a visual-based webpage analysis algorithm and it uses the Vision-based Page Segmentation (VIPS) algorithm to retrieve visual feature data. This is one kind of Screen scraping. The following paragraphs describe the concept of Screen scraping.

Screen scraping normally indicates the use of the program to collect visual data from a source. Unlike web scraping, screen scraping focuses on obtaining the screen visual data, while web scraping focuses on getting the HTML and parsing it. Generally, this research employs a method closer to screen scraping.

Screen scraping can also mean bidirectional exchanging of data. For example, to best illustrate this idea, imagine the following simulation: a program use screen scraping technique to read in a form on a webpage and fills out the asked information at the same time. More advanced techniques capture the bitmap data from the screen and run it through an Optical Character Recognition (OCR) engine, or programmatically obtain the visual object in a Graphic User Interface (GUI) and control it. This research does not use OCR, but OCR will make the vision-based page analysis more accurate.

These scraping techniques can be very useful, but they also introduce some problems, such as copyright. It is easy to scrape someone’s website and represent it in a different format in another website. Someone can also scrape all the useful information without going through the original website’s ad, and these ads serve as major revenue streams for most websites. Thus, many websites do not allow these scraping techniques.

2.4 Image Processing

This research uses the VIPS algorithm, as it is one kind of image processing. So it is important to understand some of the image processing concepts, to be able to use it in this paper. However, detailed techniques about image processing are not the main focus for this
research. The following sections will describe a few techniques in this area that I am now using and that might be useful for future research.

Image processing is any form of signal processing. It takes an image, such as a picture or video frame as an input, and outputs a set of characteristics related to the image. The term usually refers to digital image processing, but optical and analog image progressing are also possible, such as OCR.

There are many types of image processing, ranging from image enhancement, image restoration, image comparison, image recognition, to image analysis. Essentially, everything related to getting information from an image is in this field. The few that are important to this paper are image recognition that includes optical character recognition, edge detection, computer vision, and image segmentation. The following section will describe these important technologies.

2.4.1 Computer Vision

Computer vision includes capture, processing, analyzing, and understand images and three-dimensional data from the real world. It can be used on controlling manufactory processes, navigation, detecting events, organizing information, interaction between humans and computers, automatic inspection and more. It is being used by many industries right now. Edge detection is one of its subcategories, but computer vision involves many more fields, such as robotics, artificial intelligence, machine learning, physics, mathematics, neurobiology, and imaging.

The tasks of computer vision typically are object recognition, detection, identification, content-based image retrieval, optical character recognition, facial recognition, motion analysis, image restoration, etc. The most relevant fields for this research are object recognition, detection, and identification. Machine learning can also help the automated webpage analysis program.

2.4.2 Optical Character Recognition

This research does not use the OCR technique to visually extract text from a webpage image but it is very important and useful add-on to this paper, as it can increase the accuracy of the webpage analysis algorithm. Although I did not include this technique due to current limitations of expertise, I intend to employ it in future research.
OCR is an electronic conversion of a scanned image of handwritten or printed text into machine-encoded text. It is frequently utilized to extract data from paper-based sources such as documents, sales receipts, letters, or printed records. OCR is a field of research in computer vision, pattern recognition, and artificial intelligence. OCR is commonly employed in everyday life; for example, OCR is used in check verification, receipt scanning and management, automatic number plate recognition, and more. OCR was originally designed to help blind and visually impaired people but now helps everyone with everyday tasks.

OCR software usually consists of three stages: pre-processes, character recognition, and post-processing. In the pre-processes stage, OCR software can align the document, smooth edges, remove lines, identify columns and paragraphs, and normalize aspect ratio and scale. These steps reduce problems that might occur in the recognition stage.

The character recognition stage uses two basic types of algorithms, matrix matching and feature extraction.

- Matrix matching involves comparing a text image to a stored glyph on a pixel-by-pixel basis. It is known as pattern matching, relies on the input glyph being correctly isolated, and has the similar font and scale as the scanned text image. This method works best with typewritten and basic sets of font text.

- Feature extraction decomposes glyphs into features such as lines, loops, line direction and intersection; these are used to compare with an abstract vector type representation of a character. Techniques such as feature detection in computer vision and k-nearest neighbors algorithm are commonly used in modern OCR software. This research also benefit from such techniques.

Post-processing stages can increase accuracy by constraining the output via a lexicon, a list of words that are allowed to appear in the document, or by applying the knowledge of the grammar of the language to eliminate errors.

OCR software can also be customized to specific the kind of text or documents, such as license plates, screenshots, ID cards, business cards, and sales receipts. The accuracy of many applications is very high, while some may still need improvement.

2.4.3 Edge Detection

Edge detection is a fundamental tool in image processing, image pattern recognition, and computer vision, particularly in the feature detection and feature extraction area. It is a mathematical method that aims to identify points in the image which the image brightness
changes sharply, and these points are usually organized into a set of line segments called edges.

The ideal result of edge detection will be a set of connected lines that indicate the boundaries of objects, which will significantly reduce the information needed to analyze via the image processing program. Hence, good edge detection is one major requirement for image processing. However, it is not always possible to obtain such ideal results in real life situations. For example, there may be a discontinuing line where the change of color and brightness are too small, such that the edge detection program fails to pick up, and this would complicate subsequent tasks. The algorithm employed within this paper’s research utilizes the edge detection, technique but experienced a similar problem.

Many algorithms and research are being developed from time to time to help eliminate such problems, beginning with the Canny Edge Detector, developed by John F. Canny in 1986, which uses a multi-stage algorithm to detect a wide range of edges in images. It uses noise reduction based on Gaussian, and the result is a slightly blurred version of the original image, not affected by a single noisy pixel to any significant degree.

### 2.4.4 Image Segmentation

Image Segmentation in computer image processing refers to the process of partitioning an image into many segments. The purpose is to assign labels to each pixel so that those pixels with similar characteristics can be grouped together and analyzed more easily. This technique can be used to identify lines, edges, objects, boundaries, and images.

Segmentation is employed and proves very useful in this research, as it can separate different visual objects from the webpage appearance so that it can be easily analyzed by the algorithm.

The final result should be a set of segments that collectively cover the entire website appearance image, or a set of content. Typical segmentation is each pixel inside a region is similar with respect to some characteristic such as color, shape, or texture. Adjacent regions are significantly different respective to these same characteristics. This research performed segmentation a bit differently, discussed in detail in Chapter 3.
2.5 **Human Computer Interaction - HCI**

Human Computer Interaction, or Human Computer Interface (HCI), is the study of the interaction between the user and computer. It includes the study of computer science, behavioral sciences, cognition science, design principle and more. This is one key component in this research because image processing will need to understand the interface design in order to do segmentation or analysis. Data clustering process will also need to understand the human cognition process in order to cluster the visual data accurately.

HCI studies humans and computers, so it uses knowledge from both sides. On the humans’ side, it studies design disciplines, linguistics, communication theory, social sciences, cognitive psychology, and other human factors. Within computers, HCI studies computer graphics, development environments, operating systems, and programming languages.

This research needs to explore some of the design principles in order to simplify the data clustering process. Appendix B is a list of the display design principles from the book *An Introduction to Human Factors Engineering* [3] and *The Design of Everyday Things* [4].

### 2.6 Webpage Data Extraction

The webpage analysis algorithm in this research uses the VIPS algorithm to segment the webpage and that why it is related to webpage data extraction. This section will discuss some techniques that have been used in this field.

There is much research about extracting information from webpages, and [5-7] reviews have compared the technology they use and the functionality they embody.

*A Survey of Web Information Extraction System* [5] compares different Information Extraction (IE) and Wrapper Induction (WI) methods by three criteria: the task difficulty, technique used, and automation degree. It then classifies them into a manually constructed IE system, a supervised WI system, a semi-supervised IE system, and a unsupervised system. Afterwards, it gives three reviews, each based on one of the three criteria, and ends the review section with an overall comparison. This paper gives a complete review for the current technology we have in the IE and WI.

*A Brief Survey of Web Data Extraction Tools* [6] first categorizes these methods via different technologies they use, such as languages for wrapper development, HTML-aware
tools, natural language processing (NLP) based tools, WI tools, modeling-based tools, and ontology-based tools. Then it compares these methods by its automation level, technology category, adeptness level, and support of complex objects. Below is the Table 2.1 [6] and graph view (Figure 2.2), which illustrates a summarized view of each algorithm and its respective capability. The support for complex objects column reveals whether it can work within a complex web structure. The XML output column reveals whether the algorithm provides a XML output. The semi-structure data (SD) and semi-structure text (ST) in the type of page contents column reveals which type of webpage is a best fit for the method. Section 4 offers a detailed explanation of the Table 2.1 [6].

After reviewing these algorithms, we can categorize them into three types based on the automation level.

- **Manual approach:** The oldest way to extract information from the web. It designs languages to help a programmer build a wrapper for specific data items and fields. TSIMMIS [8], Minerva [9], and Web-OQL [10] are tools that use this approach. This approach lacks efficiency and scalability.

- **Semi-automatic approach:** Based on the extraction process, there are two techniques. First is sequence-based, such as WIEN [11], Soft-Mealy [12], and Stalker [13]. It parses webpages into sequences of tokens and applies extraction rules based on a few training samples. Second is the tree-based technique, such as W4F [14] and Xwrap [15]. It parses webpages into a DOM tree and extracts data from it. Both approaches require a certain degree of manual operation, such as identification and labeling of sample webpage contents.

- **Automatic approach:** In recent years, more researchers have focused on fully automated approaches, such as Omini [16], RoadRunner [17], IEPAD [18], MDR [19], and DEPTA [20]. These techniques have been discussed in [5-7]. One of the weaknesses of the previous algorithm is that they all analyze the HTML instead of the visual features. Some recent algorithms, such as ViPER [21], VENTex [22], ViNTS [23], ViDE [24], and HCRF [25], use a bit of the visual features. ViPER and ViNTS use the visual feature to help data region extraction, but they still heavily rely on tag structure. HCRF and ViDE use VIPS algorithm as I do, but HCRF still relies on tag structure.

Vision-based Data Extraction (ViDE) is the only algorithm that mainly uses visual features to extract deep web data. It uses position, layout, appearance, content features, and some special supplementary information, such as text similarity, frequently repeated symbols, and data types, to extract the data record. Each of these features has several rules, and based on the ViDE paper, these rules apply to general webpages.
<table>
<thead>
<tr>
<th>Type</th>
<th>Tools</th>
<th>Degree of Automation</th>
<th>Support for Complex Objects</th>
<th>Ease of Use</th>
<th>XML Output</th>
<th>Support for Non-HTML Sources</th>
<th>Type of Page Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Languages</td>
<td>Minerva</td>
<td>Manual</td>
<td>Coding</td>
<td>+</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>TSIMMIS</td>
<td>Manual</td>
<td>Coding</td>
<td>+</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>Web-OQL</td>
<td>Manual</td>
<td>Coding</td>
<td>+</td>
<td>No</td>
<td>None</td>
<td>SD</td>
</tr>
<tr>
<td>HTML-aware</td>
<td>W4F</td>
<td>Semi-Automatic</td>
<td>Coding</td>
<td>++</td>
<td>Yes</td>
<td>None</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>XWRAP</td>
<td>Automatic</td>
<td>Yes</td>
<td>+++</td>
<td>Yes</td>
<td>None</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>RoadRunner</td>
<td>Automatic</td>
<td>Yes</td>
<td>+++</td>
<td>No</td>
<td>None</td>
<td>SD</td>
</tr>
<tr>
<td>NLP-based</td>
<td>WHISK</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>++</td>
<td>No</td>
<td>Full</td>
<td>ST</td>
</tr>
<tr>
<td></td>
<td>RAPIER</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>++</td>
<td>No</td>
<td>Full</td>
<td>ST</td>
</tr>
<tr>
<td></td>
<td>SRV</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>++</td>
<td>No</td>
<td>Full</td>
<td>ST</td>
</tr>
<tr>
<td>Induction</td>
<td>WIEN</td>
<td>Semi-Automatic</td>
<td>No</td>
<td>++</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>SoftMealy</td>
<td>Semi-Automatic</td>
<td>Partial</td>
<td>++</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>STALKER</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>++</td>
<td>No</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Modeling-based</td>
<td>NoDoSE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>+++</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td></td>
<td>DEByE</td>
<td>Semi-Automatic</td>
<td>Yes</td>
<td>+++</td>
<td>Yes</td>
<td>Partial</td>
<td>SD</td>
</tr>
<tr>
<td>Ontology-based</td>
<td>BYU</td>
<td>Manual</td>
<td>Coding</td>
<td>++</td>
<td>No</td>
<td>Full</td>
<td>ST/SD</td>
</tr>
</tbody>
</table>

Notice that these approaches are only efficacious for semi-structured data or text, and only few can work on both and most of them require some degree of manual operation. Furthermore, all of them are HTML dependent, except the ViDE.

The ViDE process begins with noise blocks filtering, which remove top and bottom visual block if they meet certain condition. Then it use blocks clustering to form clusters. The difference between ViDE and this paper is the former focuses on extraction of deep webpage data region and data items, while the latter focuses on identification of the functionality of each visual block. I compare the relative position, the color, font, font size, the composition of each visual block, and I also apply more visual perception rules and block-level link analysis, described in detail in Chapter 3.
CHAPTER 3

VISUAL BLOCK TREE AND VISUAL FEATURES

This chapter explains how to segment a webpage via its visual features and use the result to identify the webpage content. Section 3.1 briefly explains VIPS, which is used to segment a webpage mainly via its visual features. Section 3.2 and 3.3 explain the structure of a visual block and the structure of the visual block tree, which is used to represent the segmented webpage. Section 3.4 explains the Visual Block Type Analysis algorithm (VBTA), used to analyze a visual block and identify its basic functionality. Section 3.5 explains the Webpage Analysis Algorithm (WPA), used to identify the main content, navigation area, and advertisements. The goal of this chapter is to explain the vision-based webpage analysis process.

3.1 Vision-Based Page Segmentation

This research uses VIPS [26] to segment webpages. The output of the VIPS algorithm is an XML file that has detailed visual information about the webpage. The VIPS algorithm source code is from the original research team’s website, and the website also has a sample program to demonstrate how to use the algorithm. The reason to use an open source project lies in the fact that the main focus for this research is to analyze webpages, so the image segmentation algorithm, such as VIPS, is merely a bridge to what this research is trying to accomplish. This research will describe the VIPS algorithm basic process, but will not explain everything in detail, because this research seeks only to use the output from the algorithm. In fact, this research can use any image segmentation algorithms if they provide a better result, so please keep in mind that this VIPS algorithm can, and likely will be updated or even replaced in the future for better results. The remainder of this section will explore the basic concept and flaws of the VIPS.

VIPS algorithm takes a webpage as input, and it extracts the webpage’s DOM object and the visual feature from the web browser, then uses this information to form a segmented webpage content structure. Each segmented content area is called the visual block and it is
the basic element of the tree structure, used to describe the content structure of the webpage. The detailed structure of the visual block tree is explained in Section 3.3.

The VIPS takes the information from the web browser and constructs a basic content structure, and the algorithm starts to detect the separator between the visual blocks and performs a segmentation process to further divide or merge each visual block. The separators, usually lines, are used to help the algorithm decide whether or not a block should be divided or merged. The specific rule of how the algorithm determines the separator is in Section 4.2 [26]. Based on the visual block and the separators, the algorithm creates another further divided content structure, and then it checks whether each visual block meets certain conditions. If there is a visual block that does not meet the condition, if, for example, it is too big or there is a separator in the middle, the algorithm further partitions or merges the visual block. The algorithm keeps running recursively until all segment areas (visual blocks) meet all the granularity requirements. Once all the requirements are met, the algorithm outputs a XML file containing a vision-based analyzed webpage structure.

This process is fully automated and the user only needs to enter the URL and hit the generate XML file button. The VIPS algorithm will output a XML file with the position, height, width, font size, font color, background color, content, Degree of Coherence (DoC) value, and the HTML source code for each segment area. The DoC value here represents how coherent the child node is with its parent node. In other words, the DoC value is higher when the content of the child node visually or structurally resembles the parent node.

Figure 3.1 [27] is the flow chart for the VIPS algorithm that summarizes the process. It is from the Extracting Content Structure for Webpages based on Visual Representation [27].

The ideal situation is that all data will be extracted from the XML file. Each segmented data region in the XML file is parsed to a tree node and has all the visual features as property. Visual blocks that cannot be further segmented are the leaf nodes, the basic objects such as text and picture. The problem is the VIPS algorithm does not fully satisfy these requirements. For example, sometimes the background color or the font color is missing from the XML file, or the leaf node is not a basic object such as text or picture. In conclusion, the VIPS algorithm did not generate an output that meets this research’s requirement perfectly. This is sometimes problematic because some visual blocks are harder to analyze. The solution will be discussed in the following sections.

3.2 Structure of the Visual Block

After the VIPS outputs the XML file, another PHP class extracts the data, and then I put them together to form a tree structure. The parent-child relation and the segmented area’s visual features are directly copied from the XML file. Each visual block has properties such as ID, position, background color, height and width, font size and color extracted from the XML file. The position is the (X, Y) which X is the horizontal coordinate starting at the left and Y is the vertical coordinate starting at the top of the webpage. Figure 3.2 shows the webpage coordinate system.

![Figure 3.2. The coordinate system of a webpage.](image)

I also add another property called type. This property is implemented to describe the basic functionality for each block, whether it be background, picture, text, text hyperlink, etc. It is analyzed by another algorithm, and the detail is in Section 3.4. Adding this extra property allows the analysis of the functionality of each block better. The VIPS algorithm only provides very basic information, such as position, so it is hard to analyze the content. The type property tells me the visual block is mainly text or picture, or whether it has
hyperlinks or not. In other words, this type of property offers basic analysis for the visual block.

### 3.3 Structure of the Visual Block Tree

This research uses a tree structure to describe a webpage.

To make it easy to understand, I number each level of the tree. It is numbered from the top to the bottom. The level zero is the root and level one is the child nodes of the root, and so on. This organizational method allows me to categorize each level. For example, level zero is the summary data for the webpage, level one is the basic layout of the webpage, and level two is the visual feature for each data record.

The root of the tree represents the entire webpage with position \((X, Y)\) at \((0, 0)\), which is the upper-left corner’s coordinate. The background color is the major color of the webpage, type is background, and height and width is the webpage’s height and width.

The level one child nodes are the partition of the root, and the union of all the level one child nodes should be smaller or equal to the root. In other words, the root represents the webpage and the level one child nodes represent different visual objects in the webpage; the level \(n\) child nodes are the partition of the \(n-1\) node. This structure resembles the one the research team used in the section 3 [27]. Figure 3.3 [27] shows the segmented web page image and Figure 3.4 [27] illustrate the detail of this structure.

The layout of the tree is like Figure 3.5 and the level zero is the root, which is the whole webpage that has the basic information, such as the basic color of the website and the size of the webpage.

The level one child nodes are the general layout of the website. These nodes should distinguish the major division, and level one should be able to decipher the layout of the webpage. For example, in Figure 3.5, the level one child nodes are \((VB1, VB2, VB3, VB4)\), and the Figure 3.5 illustrates that the webpage has four major area, which are divided horizontally. The level two \((VB1_1, VB1_2, VB2_1, VB2_2, VB2_3)\) child nodes divide the level one object into smaller visual block with more detail. It offers a more detailed look of the level one visual block. If the level one is not enough for the analysis algorithm to identify
Figure 3.3. Sample segmented webpage with the visual block and the separator. Source: D. Cai, S. Yu, J. Wen, and W.-Y. Ma. Extracting content structure for webpages based on visual representation. Proceedings Asia Pacific Web Conf. (APweb), 2003.

\[
O = (VB1, VB2, VB3, VB4) \\
\Phi = \{\varphi^i, \varphi^j, \varphi^k\} \\
\delta = \begin{pmatrix} \varphi^i \\ \varphi^j \\ \varphi^k \\ \text{NULL} \end{pmatrix} \\
\Phi^i = \{\varphi_s^i, \varphi_s^j\} \\
\delta^i = \begin{pmatrix} \varphi_s^i \\ \text{NULL} \end{pmatrix}
\]

Figure 3.4. The corresponding specification of Figure 3.3. Source: D. Cai, S. Yu, J. Wen, and W.-Y. Ma. Extracting content structure for webpages based on visual representation. Proceedings Asia Pacific Web Conf. (APweb), 2003.

Figure 3.5. The layout structure and vision-based content structure of Figure 3.3.
the functionality of the area, then going to the level two should help this research decide the functionality of the area, and so on.

For a simple webpage, level two should have enough information to decide the functionality of each section. But it is possible to include more levels to offer even more detailed information to help the algorithm. In fact, in my test, the analysis algorithm has to analyze much more levels than I planned because of the way the VIPS algorithm output the segmented structure. This is one problem that can be studied in the future.

3.4 Visual Block Type Analysis Algorithm

This section explains how the VBTA algorithm works. It is used as a first-step analysis to identify basic functionalities of a visual block.

Below are the descriptions for the types I have in this paper.

- Text: the area contain mainly text
- Pictures: single or multiple pictures with clear separation objects to separate them
- Text hyperlinks: hyperlinked text that links to the same domain
- Text and text hyperlinks: an area mixed with standard text and hyperlinked text
- Picture hyperlinks: hyperlinked picture that links to the same domain
- Foreign: hyperlink that links to a different domain
- Noise block: small block that has no important information
- Multiple items: Composition of the types above.

With a list of types, the analysis algorithm can better decide functionality of each visual block.

With definition for types, the algorithm then use web scraping techniques to capture the hyperlink in the HTML source code and analyze it along with other data. The HTML source code is provided by the VIPS output file as Section 3.1. This hyperlink information will be used to determine the type of each visual block and help the analysis algorithm to decide the functionality of the block. Fetching the hyperlink is a common technique in the industry and the usability has been proven. See Webbots, Spiders, and Screen Scrapers [28] for more detail about this technique.

As mentioned in the previous chapter, the VIPS did not segment the webpage as it supposed to. Some leaf nodes still contain too many pictures, text, and hyperlinks. In order to
analyze the visual block more accurately, the algorithm used some equations to help identify, and they are listed below:

- **Degree of Text Hyperlink (DoTH):** a percentage of hyperlink text in the block and the higher the number, the more hyperlinks in the block.
  \[
  \text{DoTH} = \frac{\text{number of text hyperlinks in the block}}{\text{number of text tags in the block}}
  \]  
  (1)

- **Degree of Picture Hyperlink (DoPH):** a percentage of hyperlink images in the block.
  \[
  \text{DoPH} = \frac{\text{number of pictures hyperlink in the block}}{\text{number of image tags in the block}}
  \]  
  (2)

- **Text Ratio (TR):** It is for checking the relative text percentage of a visual block compared to the whole page. The block is more important if the ratio is higher.
  \[
  \text{Text Ratio} = \frac{\text{number of characters in the block}}{\text{number of characters in the webpage}}
  \]  
  (3)

- **Linked Text Ratio (LTR):** the percentage of linked text inside a block.
  \[
  \text{Linked Text Ratio} = \frac{\text{number of linked characters in the block}}{\text{number of characters in the block}}
  \]  
  (4)

  The difference between LTR and DoTH is LTR measures the hyperlink character count and DoTH measures the hyperlink number. If LTR is high, we can determine that this block has a higher percent of hyperlink text, which means this block’s function is mainly a hyperlink to another webpage. If this visual block has higher DoTH, it means the number of hyperlinks is high, and that means this block’s function is closer to navigation because it has many hyperlinks to different places.

- **Picture Text Ratio (PTR):** It measures the picture verse text ratio, which means this visual block has a higher percentage of images if the PTR is high.
  \[
  \text{Picture Text Ratio} = \frac{\text{number of image tags in the block}}{\text{number of text tags in the block}}
  \]  
  (5)

- **Hyperlink Text Ratio (HTR):** It measures the average character in each hyperlink.
  \[
  \text{Hyperlink Text Ratio} = \frac{\text{number of linked characters in the block}}{\text{number of text hyperlinks in the block}}
  \]  
  (6)

- **Degree of Local Text Hyperlink (DoLTH):** It measures the number of hyperlinks that are linked to the same domain as the analysis webpage, and higher the number means higher number of local hyperlinks.
  \[
  \text{DoLTH} = \frac{\text{number of local text hyperlinks in the block}}{\text{number of text hyperlinks in the block}}
  \]  
  (7)

- **Degree of Local Picture Hyperlink (DoLPH):** same as DoLPH but it measures the picture hyperlink.
The VBTA algorithm for assigning type works as follows: It first checks the PTR, and if it passes a threshold $e_1$, the block is classified as a picture block; otherwise it is a text block. Next, the algorithm checks DoTH or DoPH depending on the previous classification. The algorithm checks the DoTH if the block is a text block, and if the DoTH passes a threshold $e_2$, then this block is classified as hyperlinks block. It is the same with the picture block, except the algorithm checks the DoPH and a threshold $e_3$. Finally, the algorithm checks the DoLTH or DoLPH. The same rule applies here. The algorithm checks DoLTH for text blocks and DoLPH for picture blocks. The block is classified as foreign block if DoLTH or DoLPH pass a threshold $e_4$ or $e_5$.

The purpose of these equations is to help the VBTA algorithm better decide the visual block functionality. For example, we know the block is designed for the user to click on if the visual block is a hyperlink text or picture; otherwise, it is designed to work as a label or divider between sections, or just some information to read. Further, if the visual object is linked to another domain, then it is likely to be advertisement; otherwise it is more like a navigation link. This research only analyzes whether the link is linked to the same domain or not, but future research can apply more link analysis. Please read *Block-Level Link Analysis* [29] for more detail about the block-level link analysis.

This type property is very useful for the content analysis and is discussed in the following section.

### 3.5 Webpage Analysis Algorithm

Previous sections explored how to extract information from a webpage and segment it to a structure that is easier to analyze. This section will explore how to use the segmented structure to identify the different functionalities of each segmented block.

Again, the purpose of this research is to identify webpage content by its appearance and layout. In order to do this, the algorithm needs a set of cognition rules that humans will use when looking at a webpage.

Below is the feature list for the main content, advertisement, and navigation, and each has four set of features: position, layout, appearance, and content. I obtain these features via
observation of webpages on the World Wide Web. Future research can test these visual features and determine they apply to most webpages.

Before we start, I like to explain some acronyms used here. The data record is a smaller visual block with only small amount of data and usually represents one item or one news article. For example, it can be one picture with one paragraph. A data region is a larger visual block with multiple data records that represent a group of merchandise or one section of news such as today's financial news.

### 3.5.1 Main Content

Position features:
- PF1: Data regions are always centered in the webpage.

Layout features:
- LF1: Usually has more data record than other data regions.
- LF2: The size of the data region is usually larger than other data regions.

Appearance feature:
- AF1: Data records are very similar in their appearances, and the similarity includes the sizes of the image contained and the fonts used.

Content features:
- CF1: The presentation of data item in data records follows a fixed order.
- CF2: The text ratio is usually higher.

### 3.5.2 Advertisement

Position features:
- PF1: data region usually locates at the edge of a webpage.

Layout features:
- LF1: The data regions are usually around the main content region.
- LF2: The size of the data region is usually smaller than the main content data region.

Appearance feature:
- AF1: Data records are different in their appearance compared to the main content, and the difference includes the sizes of the image they contain and the fonts they use.
Content features:

- CF1: Most likely is hyperlinked to a different webpage under different domain.

### 3.5.3 Navigation

Position features:

- PF1: Data region is always placed horizontally at the top of a webpage or placed vertically at the left or right side of a webpage. Occasionally it is placed horizontally at the middle of a webpage with contents on top.

Layout features:

- LF1: The data records are usually aligned vertically in the data region when the data region is vertical, and horizontally when the data region is horizontal.
- LF2: All data records are adjoining or have same space between them.

Appearance feature:

- AF1: Data records are very similar in their appearance.

Content features:

- CF1: The presentation of data item in data records follows a fixed order.

The purpose of the Webpage Analysis algorithm (WPA) is to select main content, advertisement, and navigation menu.

For selecting the main content area, I select the relatively larger block with position relatively closer to the center or the top of the page with high TR.

For the navigation block, I used a method to select a list of text hyperlinks or picture hyperlinks without the foreign keyword, with the size relatively smaller than the main content. If there are multiple objects that have these characteristics, then I select the one with the largest font closer to the main content block or closer to the top left corner of the webpage to be the main navigation. WPA output the rest as sub navigation block.

For advertisements, I select foreign text or picture hyperlinks from the type with the position at the edge of the webpage, and relatively smaller size compared to the main content block.

After the area has been selected, I output the image of the webpage and use different colors to circle the corresponding visual block and label to describe the category.
CHAPTER 4

TEST RESULT

The previous chapter explained how the algorithm works. This chapter will explain how I set up the test environment and reveal the test results in the web browser.

First I have to obtain sample webpages with different designs. I chose 14 sample pages from the World Wide Web. They include webpages from school, news, shopping, banks, video games, airlines, and online media. Below is the list.

- http://blackboard.sdsu.edu
- http://www.sdsu.edu
- http://www.yahoo.com/
- http://www.ign.com/
- http://www.amazon.com/
- http://www.buy.com/
- http://www.united.com/
- https://www.bankofamerica.com/
- https://www.chase.com/
- http://www.cnyes.com
- http://www.youtube.com
- http://www.cnn.com

The process of the test is as follows: I run VIPS first and input the URL of the webpage to generate the XML file. I then use the algorithms I explained in Chapter 3 to process the XML file and output a webpage with a color rectangle surrounding different visual blocks. The color is based on the functionality of the visual block. I use black for main content, red for main navigation menu, blue for sub navigation menu, and green for advertisements. I use screen capture software to capture the webpage image and use it as background image instead of showing the color rectangle on the real webpage. It is easier to do it this way but sometimes the outline is not aligned correctly.
The final result is easy to check manually by determining whether the circled visual block's color correctly represents the correct functionality. I run the algorithm on 14 webpages and 11 of them can be correctly parsed and circled out the main content, navigation, and advertisement area. The accuracy is about 78.6%. Three of the webpages are impossible to analyze, as the XML files only contain one visual block. It seems VIPS has some problems segmenting the webpages.

Again, the detail of the algorithm is explained in section three. The threshold values are listed in the Table 4.1. These test values were obtained by my observation through the World Wide Web.

**Table 4.1. Visual Block Type Analysis Algorithm Variable Setup Values**

<table>
<thead>
<tr>
<th>Name</th>
<th>PTR(e1)</th>
<th>DoTH(e2)</th>
<th>DoPH(e3)</th>
<th>DoLTH(e4)</th>
<th>DoLPH(e5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
<td>0.2</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

The test shows that this algorithm can identify possible areas such as main content, navigation menus, and advertisements based on the visual features and the hyperlinks, but the accuracy depends on the VIPS algorithm. If the visual block tree giving to the WPA algorithm is accurate, and the VIPS divides each visual block as its paper stated, then the accuracy is good, otherwise it is not. For example, VIPS cannot segment the chase.com webpage and it output the xml file with only one visual block. Clearly the WPA algorithm is not able to identify the functionality. Two other webpages also produce the same result. The Table 4.2 shows the test results.

**Table 4.2. Webpage Analysis Algorithm Detail Test Result**

<table>
<thead>
<tr>
<th>Total VIPS segmented</th>
<th>Correctly classified</th>
<th>Partially correctly classified</th>
<th>Wrong classified</th>
<th>Failed to segment webpages</th>
</tr>
</thead>
<tbody>
<tr>
<td>58 Visual blocks</td>
<td>53 Visual blocks</td>
<td>3 Visual blocks</td>
<td>2 Visual blocks</td>
<td>3 pages</td>
</tr>
</tbody>
</table>

Yahoo, Amazon, and other webpages, with clear visual separators between visual blocks, can be analyzed accurately by the VIPS. The WPA algorithm is able to point out the main content, main navigation, sub navigation, and advertisements, but there are still some problems, as when the VIPS did not divide the visual blocks correctly. Based on the VIPS
paper, all leaf nodes should be the basic visual elements, such as a picture, a sentence, or a paragraph. However, some leaf visual blocks in the test result are still too big and contain too many visual elements; the WPA algorithm thus analyzes a visual block with multiple purposes. For example, a visual block with both advertisement and navigation purposes is segmented to one visual block. It should be segmented to two visual blocks, but the VIPS fails to do that, and the WPA algorithm has a problem identifying the functionality of such visual block. The DoLTH, DoLPH, and other numbers I used in this research can help increase the accuracy of the WPA algorithm when VIPS fail to deliver an accurate visual block tree. But the end result is still trying to decide the functionality between two purposes, such as navigation menu and advertisements. The WPA algorithm is designed to pick the one with higher possibility, so the output result is most likely to ignore the secondary functionality. Conclusively,, the WPA algorithm is more likely to identify the block with the functionality that has relatively a larger portion if a visual block has more than one functionality.

Also, if the website is designed for some special purpose; for example, for a complex advertisement, this will decrease the WPA algorithm’s accuracy, because the main purpose of that webpage is to have the advertisement central as the main content.

Also, some advertisements cannot be identified and some can. For example, common advertising techniques are:

- pop-up ads
- plain text
- ad banners
- flash animations
- keyword hyperlinks
- browser plugins/extensions
- external applications

Some of these advertisements are easy to identify via the WPA algorithm, such as keyword hyperlinks and ad banners. Others may be hard to identify because VIPS will not capture those into the xml file.

Besides that, the test results also show that some websites use cookies and session to store browser history and use it to change webpage content dynamically. The VIPS algorithm
is not able to extract the content if it does not have those cookies or session information, which is stored on the client side; therefore, the accuracy of the WPA for a webpage that utilizes client side information will be lower.

In the end, the biggest problem of this research is that the accuracy is affected by the VIPS algorithm because each step in this research must be accurate to produce an accurate result. If the visual block tree is well parsed and the visual block’s type is well defined, then it is easier for the WPA algorithm to decide the functionality.
CHAPTER 5

CONCLUSION

5.1 TEST CONCLUSION

The goal of this research was to find a better way to analyze webpages and to design an algorithm to identify the functionality of each object visible on the screen. After reviewing some of the technologies and other research about deep web data extraction, I find the visual-based page analysis is the better solution to this problem, as the test result reveal that 11 out of 14 webpages can be processed correctly.

There are a few advantages to this approach. First, it will not have to deal with dynamic contents such as AJAX, because it only analyzes the final view a user can see. It can re-analyze the screen if it changes. Second, it can avoid intentionally misleading tags, I.D., and other text-based information. Third, it can use the same rules with different version of HTML and CSS. Fourth, it can avoid some websites that prohibit web scraping.

However, the current technology still has some constraints that keep it from fully analyzing webpages with only visual feature, so I decided to use the vision-based page segmentation algorithm with some hyperlink analysis. The test reveals satisfactory results.

This research proposes a new way to analyze webpages by its visual features, design principles, and hyperlink features. It uses VIPS algorithms to extract visual features and web scraping to gather other data, and then analyze these based on the set of features most commonly used by webpages, and finally to output the test results to a web browser.

Visual-based approaches reveal that it is possible to analyze websites using visual features, since the website is very limited in design rules, and the web designer is trying to keep it that way so their webpages can be widely read by ordinary users. With the help of web scraping and hyperlink analysis, this approach can do fairly well at judging the functionality of the content.

The major problem of this approach is that the VIPS algorithm has problems segmenting some webpages, and the page segmentation process is not fully based on image
analysis, so it is somehow still not completely code-independent. However, this problem can be fixed by using another fully visual-based image segmentation algorithm.

Three major topics can be studied for further research to help improve the accuracy of visual-based webpage analysis. First is to better identify and separate the text and the picture in each webpage, which can solve the problem of leaf visual blocks containing too many elements. Second is to be able to segment webpages fully based on the visual feature, and this should help VIPS segment some webpages better. Third is to apply more visual feature rules that webpages use to help identify the functionality of each visual block. The rules this research use are still very basic compared to human cognition’s capabilities. The accuracy of the WPA algorithm should increase with more rules.

Finally, I would like to point out that there are other essays that explore ways to make the web more accessible via machines. For example, the Semantic Web is trying to design a new web programming standard so the machine can easily understand it. I feel that we need both approaches, because it is impossible to ask people to recreate all the webpages. We will still need an automated system to gather all the information currently on the web, even after we have a new standard to create new webpages.

5.2 Difference Between Web Scraping and WPA

Normal web scraping is mainly for extracting data from the webpage, so the main functionality is to identify and extract the important data. It does not identify other types of content within a webpage, for example, advertisement and navigation. However, WPA is for analyzing the webpage content so it not only can identify and extract important data but can identify other contents. This gives WPA further functionality and can be used in many different situations. Companies can use WPA for data extracting, webpage structure analysis, webpage reforming and reorganizing, automated webpage browsing, webpage usability analysis, and more. WPA is designed so that computers can recognize the content of a webpage and therefor WPA can be applied to more scenarios than normal web scraping.

5.3 What Can WPA Do

This Visual-Based Webpage Analysis algorithm can be used for commercial or academic purposes. For example, search engines can use this technique to aim for the main content inside a webpage. Daily users can use it to block some advertisements they do not
want to see. Internet security companies can use it to block dangerous links or fake websites. Academics can use it to analyze webpage structures and see how a webpage design can affect usability.
REFERENCES


APPENDIX A

DETAILS OF SEMANTIC WEB LAYERS
XML is the syntax for the content structure within the document. It is a markup language and is both human-readable and machine-readable. It does not provide the meaning of the data, but it uses tags to describe the type of the data, and provides a way for program to give some properties to the data. XML is fairly common in today’s web design. Many document formats have been developed based on the XML syntax, such as RSS, Atom, SOAP, and XHTML, and it is default format for many office productivity tools, such as Microsoft Office, OpenOffice.org, and Apple’s iWork. It is also the base language for communication protocols such as XMPP.

Extensible Messaging and Presence Protocol (XMPP) is a communications protocol for message-oriented middleware based on XML.

RDF, which is the fundamental of Semantic Web, is a simple language which describes the relationship between data or data model. It is a classic conceptual model such as entity-relationship. It uses subject-predicate-object expression, called as triples, to present the relationship. For example, “the sea has color blue”, where the “sea” is the subject, “has color” is predicate, and “blue” is object. We can also use object orientated concept to represent this idea. That way the “sea” is the object, “color” is the attribute and “blue” is the value. It is a major component in the automated software throughout the Web. Its ability to model disparate, abstract concepts has also led to its increasing use in knowledge management.

RDF Schema, is a set of classes with certain properties using the RDF extensible knowledge representation language, providing basic elements for the description of ontology, otherwise called RDF vocabularies.

OWL, Web Ontology Language, is a family of knowledge representation languages for authoring ontology. It adds more vocabulary for describing properties and classes of RDF-based resources.

SPARQL, is a query language used for RDF format and can store, retrieve, and manipulate data. It allows users to write unambiguous queries.

RIF, Rule Interchange Format, is used to exchange rules between many rule languages. It is not just a format about syntax but an extensible framework for rule-based languages, which includes precise and formal specification of the syntax, semantics, and XML serialization.
APPENDIX B

USER INTERFACE DESIGN PRINCIPLE
**Perceptual principles**

1. *Make displays legible (or audible)*. A display’s legibility is critical and necessary for designing a usable display. If the characters or objects being displayed cannot be discernible, then the operator cannot effectively make use of them.

2. *Avoid absolute judgment limits*. Do not ask the user to determine the level of a variable on the basis of a single sensory variable (e.g. color, size, loudness). These sensory variables can contain many possible levels.

3. *Top-down processing*. Signals are likely perceived and interpreted in accordance with what is expected based on a user’s past experience. If a signal is presented contrary to the user’s expectation, more physical evidence of that signal may need to be presented to assure that it is understood correctly.

4. *Redundancy gain*. If a signal is presented more than once, it is more likely that it will be understood correctly. This can be done by presenting the signal in alternative physical forms (e.g. color and shape, voice and print, etc.), as redundancy does not imply repetition. A traffic light is a good example of redundancy, as color and position are redundant.

5. *Similarity causes confusion: Use discriminable elements*. Signals that appear to be similar will likely be confused. The ratio of similar features to different features causes signals to be similar. For example, A423B9 is more similar to A423B8 than 92 is to 93. Unnecessary similar features should be removed and dissimilar features should be highlighted.

**Mental model principles**

6. *Principle of pictorial realism*. A display should look like the variable that it represents (e.g. high temperature on a thermometer shown as a higher vertical level). If there are multiple elements, they can be configured in a manner that looks like it would in the represented environment.

7. *Principle of the moving part*. Moving elements should move in a pattern and direction compatible with the user’s mental model of how it actually moves in the system. For example, the moving element on an altimeter should move upward with increasing altitude.

**Principles based on attention**

8. *Minimizing information access cost*. When the user’s attention is diverted from one location to another to access necessary information, there is an associated cost in time or effort. A display design should minimize this cost by allowing for frequently accessed sources to be located at the nearest possible position. However, adequate legibility should not be sacrificed to reduce this cost.

9. *Proximity compatibility principle*. Divided attention between two information sources may be necessary for the completion of one task. These sources must be mentally integrated and are defined to have close mental proximity. Information access costs should be low, which can be achieved in many ways (e.g. proximity, linkage by
common colors, patterns, shapes, etc.). However, close display proximity can be harmful by causing too much clutter.

10. **Principle of multiple resources.** A user can more easily process information across different resources. For example, visual and auditory information can be presented simultaneously rather than presenting all visual or all auditory information.

**Memory principles**

11. **Replace memory with visual information: knowledge in the world.** A user should not need to retain important information solely in working memory or to retrieve it from long-term memory. A menu, checklist, or another display can aid the user by easing the use of their memory. However, the use of memory may sometimes benefit the user by eliminating the need to reference some type of knowledge in the world (e.g. an expert computer operator would rather use direct commands from memory than refer to a manual). The use of knowledge in a user’s head and knowledge in the world must be balanced for an effective design.

12. **Principle of predictive aiding.** Proactive actions are usually more effective than reactive actions. A display should attempt to eliminate resource-demanding cognitive tasks and replace them with simpler perceptual tasks to reduce the use of the user’s mental resources. This will allow the user to not only focus on current conditions, but also think about possible future conditions. An example of a predictive aid is a road sign displaying the distance from a certain destination.

13. **Principle of consistency.** Old habits from other displays will easily transfer to support processing of new displays if they are designed in a consistent manner. A user’s long-term memory will trigger actions that are expected to be appropriate. A design must accept this fact and utilize consistency among different displays.