Supervised Segmentation of Web Pages for Vibro-Tactile Access on Touch-Screen Devices

Waseem Safi, Fabrice Maurel, Jean-Marc Routoure, Pierre Beust, Gaël Dias

Supervised Segmentation of Web Pages for Vibro-Tactile Access on Touch-Screen Devices

Waseem Safi Fabrice Maurel Jean-Marc Routoure Pierre Beust Gaël Dias
University of Caen Basse-Normandie
GREYC CNRS UMR 6072
Bd Maréchal Juin
14032 Caen Cedex - France
firstName.lastName@unicaen.fr

Abstract

Perception the visual layout of a web page and analyzing its DOM structure is a fundamental step in automatic adaptation process of web pages. In this paper, we present a new hybrid web page segmentation algorithm dedicated to vibro-tactile access on touch-screen devices. This hybrid algorithm combines three approaches for adapting web page contents to be presented on touch-screen devices, and for testing new navigation paradigm. The proposed algorithm is essential in our framework which aims to enhance the ability of Visually Impaired Persons (VIP) to navigate the Web by converting web pages into vibrating pages using a graphical vibro-tactile language. We present a comparison between automatic segmented pages (obtained by our proposed algorithm) and manual segmented pages. The objectives of this comparison are, on the one hand, to know how users understand web layout structures based on their visual perception, and on the other hand, to explore the main differences between automatic and manual segmentation.

I. Introduction

VIP depend on screen readers in order to deal with computer operating systems and computational programs. One of most important and desired targets by VIP is navigating the Web, considering the increased importance and expansion of web-based computational programs. Screen readers present some solutions to navigate the textual and graphical contents of web pages, either by transforming a web page into a written Braille, or by transforming the web page into a vocal output. In addition to these solutions, some screen readers installed on touch devices transform a tactile input into a vocal output.

There are some drawbacks for these proposed solutions. On the one hand, Braille techniques are costly and only few number of VIP has learned Braille [1]. On the other hand, transforming the information of a web page into a vocal format might not be suitable in public and noisy environments, and most of Braille solutions are not suitable for mobile devices [1]. In addition to these drawbacks, an important drawback is the failure to transform the 2-dimension web page structure. Indeed, as reported by many authors, perceiving the 2D structure of web documents greatly improves navigation efficiency and memorization as it allows high level text reading strategies such as: rapid or cursory reading (skimming process), finding or locating information (scanning process), to name but a few [2]. Because of these difficulties to learn and retain web site organization, most of blind people are not able to really have a "surfing experience"; Their web usage is limited to few sites they are familiar with.

Our work focuses on developing and evaluating a sensory substitution system based on a vibro-tactile solution, which may improve the mentioned drawbacks. In particular, we study how to increase the VIP perception of a 2D web page structure and how to enhance their techniques to navigate the contents of web pages on touch-screen devices. The suggested solution is cheaper than Braille devices and may be efficient in noisy and public environments compared to vocal-tactile solutions.

Our contribution is three-fold: (1) designing a Tactile Vision Sensory System (TVSS) represented by an electronic circuit and an Android program in order to transform light contrasts of touch- screen devices into haptic vibrations; (2) designing an algorithm for segmenting web pages in order to support the VIP by a way which may enhance their ability to navigate the textual and graphical contents of web pages and (3) exploring how users understand web layout structures based on their visual perception. The paper is organized as follows. First, in section 2, we review rapidly most advanced VIP targeted technologies, and propose our own framework. In section 3, the state of the art for webpages segmentation methods is reviewed. In the fourth section, our hybrid segmentation method is presented. In section 5, we present the results obtained in an experiment where the differences between automatic and manual web page segmentation are compared. Finally, perspectives and conclusions are presented.

II. VIP TARGETED TECHNOLOGIES AND PROPOSED FRAMEWORK

Current products for VIP such as screen readers mainly depend on speech synthesis or Braille solutions, e.g. ChromeVox [1], Windows-Eyes [2], or JAWS (Job Access With Speech) [3]. Braille displays are complex and expensive electromechanical devices that connect to a computer and display Braille characters. Speech synthesis engines convert texts into artificial speech, where the text is analyzed and transformed into phonemes. These phonemes are then processed using signal processing techniques.

Some screen readers can also support tactile feedback when working on touch-screen devices, such as Mobile Accessibility and Talkback [4] for Android, or VoiceOver [5] for iOS. Many of these products propose shortcuts for blind users to display a menu of HTML elements existing in the web page, for example headers, links and images. But, the main drawback of all these products is the fact that they transfer the information of web pages into a linear way i.e. without any indication of the 2-dimension global structure.

Many researches tried to enhance the way VIP interacts with web pages, such as [3], who proposed a tactile web navigator. This navigator extracts texts from web pages and sends them to a microcontroller responsible of displaying the text in Braille language using an array of solenoids. A tactile web browser for hypertext documents has been proposed by [4]. This browser renders texts and graphics for VIP on a tactile graphical display and supports also a voice output to read textual paragraphs and to provide a vocal feedback. The authors implemented two exploration modes, one for bitmap graphics and another one for scalable vector graphics. A pin matrix device is used to produce the output signal for blind users. The main drawback of these two proposed systems is that they need specific devices (solenoids and pin matrix), which are expensive and cannot be integrated in handled devices such as PDAs or Tablet PCs. Another interesting model called MAP-RDF (Model of Architecture of Web Pages) has been proposed by [5]. This model allows representing the structure of a web page and provides blind users an overview of the web page layout and the document structure semantics. Finally, Tactos is a perceptual interaction system, which has been suggested by [6] and consists of three elements: (1) tactile simulators (two Braille cells with 8 pins) represent a tactile feedback system, (2) a graphics tablet with a stylus represents an input device and (3) the computer. More than 30 prototypes of Tactos have been released to serve a lot of users in many domains. Tactos has been successfully used to recognize simple and complex shapes. The device has been also used in geometry teaching domain in an institution for visually impaired and blind children. Tactos also allowed psychology researchers to propose and develop new paradigms for studying perceptions and mediated communication of blind persons [7]. However, it shows the same drawbacks (expensive and need specific devices) as the previous systems. Moreover, the blind user can only explore the web page with a stylus and both hands are occupied by the system. Moreover, it is unemployable for a large set of environments, for example in public.

The "first glance" can be defined as the ability to understand the document layout and its structural semantics in a blink of an eye [3]. In this work, we aim to increase the ability of VIP to understand the 2-dimension web page layout in order to enhance their tactics to navigate the Web with a vibro-tactile feedback. The first phase in our model is to extract visual structures in the navigated web page and convert these "visual" blocks into zones (or segments) to facilitate the navigation in later phases. We achieve this phase using a hybrid segmentation method. Then the system represents the extracted visual elements as symbols using a graphical language. The third phase is to browse these graphical symbols depending on the size of the used touched-screen device; and in the fourth phase, our system provides a vibro-tactile feedback when the blind user touches the tablet by giving the user a vibro-tactile feedback by transforming light contrasts of touch screen devices into haptic vibrations.

To achieve the desired system, we have designed an electronic circuit, which controls four micro-vibrators placed on two fingers. A bluetooth connection with an Android tablet allows controlling the vibration intensity (i.e. amplitude) of vibrators. An Android dedicated program on the tablet displays an image on the screen and detects where the user touches the tablet screen (the viewed image represents the result of web page segmentation). The light intensity at touched points is transmitted to the embedded device in order to control the vibration intensity. In this paper, we focus only on the first phase (extracting visual structures in the navigated web page, and converting them into zones). A detailed description of hardware components of the system and results of pre-tests are described in [2] [8], and [18].

III. RELATED WORKS

Segmenting a web page is a fundamental phase for understanding its global structure. Extracting the global structure of web pages is useful in many domains such as information retrieval, data extraction, and similarity of web pages. Many approaches

[1] http://www.chromevox.com/ [Access 10/12/2014]

[2] http://www.synapseadaptive.com/gw/wineyes.htm [Access 10/12/2014]

[3] http://www.freedomscientific.com/ [Access 10/12/2014]

[4] https://play.google.com [Access 10/12/2014]

[5] http://www.apple.com/fr/accessibility/ [Access 10/12/2014]

have been suggested for segmenting web pages, such as:

a) DOM-based segmentation: it depends on analyzing the DOM tree (Document Object Model), and extracting the main structure of web pages depending on HTML tags as described in [9]. In this paper, the authors determine firstly the layout template of a web page, and then it divides the page into minimum blocks, and finally collects these minimum blocks into content blocks.

b) Vision-based segmentation: this method divides the web page depending on the visual view of web page contents on a web browser. The well known tool VIPS (VIsion based Page Segmentation) [10] is based on this approach.

c) Image processing based segmentation: this approach captures an image for the visual view of a web page, and then depends on image processing techniques to divide the captured image into subblocks [11] [12].

d) Text-based Segmentation: this approach focuses on extracting only information about texts in a web page. After dividing the web page into blocks of texts, it could be possible to find the semantic relations between these textual blocks. This method is useful in many information retrieval domains such as question answering applications [13].

e) Fixed-length segmentation: this approach divides the web pages into fixed length blocks (passages), after removing all HTML tags, where each passage contains a fixed number of words [14].

f) Densitométrie analysis based segmentation: this approach depends on methods applied in quantitative linguistics, where text-density refers to a measure for identifying important textual segments of a web page [15].

g) Graph-based segmentation: This approach depends on transforming the visual segments of a web page into graph nodes, then applying many common graph methods on these nodes for combining them into blocks, or for making a clustering for these nodes. Some common works which depend on this approach are [16] [17].

h) and Hybrid-based segmentation: This approach combines many approaches indicated previously.

IV. SUGGESTED HYBRID SEGMENTATION ALGORITHM

Most of segmentation algorithms render first the web page using a web browser, and then segments the HTML elements into many blocks depending on the visual layout. Our constructed hybrid segmentation algorithm has been tested on 154 pages collected manually from many newspapers and e-commerce sites (www.leparisien.fr, www.lefigaro.fr, www.liberation.fr, www.amazon.fr, www.materiel.net, wwww.photobox.fr). The results have been integrated with our under-development Android program. The obtained results are promising because the segmentation algorithm can efficiently extract the web page blocks depending on the visual structure, and the algorithm can also convert correctly these blocks into zones (clustering the blocks). Our algorithm mixes three segmentation approaches, DOM-based segmentation, vision-based segmentation, and graph-based segmentation.

A. Vision-Based Approach:

In this phase, we render the web page using Selenium [6] web driver and Mozilla FireFox browser, and we get its visual structure by Java-script code injection inside the HTML source code of the rendered web page. The obtained visual structure indicates a global hierarchy of the rendered web page. This phase assigns additional information for each DOM HTML element such as XPath and bounding box (location [X0, Y0], and size [height and width]). The input of this phase is a web page HTML source code, and its output is an augmented HTML web page with injected infor- mation about bounding boxes and DOM XPath for each HTML elements. In next sections, we refer to bounding boxes by blocks (i.e. each bounding box represents an HTML element, and may contain other bounding boxes).

The main steps to obtain these blocks for a web page W are:

```
1. Running Selenium web driver (rendering
   the web page W using FireFox browser),
2. Getting the DOM-structure elements
   DOM-Tree:=DOM(W),
3. For each element e (Block) in DOM-Tree
      elements:
    3.1. Get information about element e (XPath,
         and bounding box),
    3.2. Inject the obtained information in HTML
         source,
4. Return the augmented web page (DOM-Tree with
   injected information).
```

[6] http://www.seleniumhq.org/

B. DOM-Based Approach:

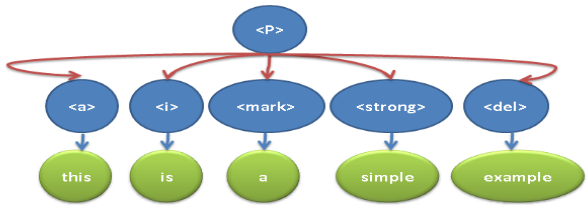
After obtaining the visual structure of a web page, we analyze its DOM structure by applying filters and re-organization rules for enhancing results of next phases. We divide the DOM elements depending on the specification of HTML5 content models proposed by the World Wide Web Consortium (W3C) [7]. This specification divides the HTML tags to 7 categories (Metadata content, Flow content, Sectioning content, Heading Content, Phrasing content, Embedded content, and Interactive content). The first applied filter is Metadata-Content-Filter, which deletes all the elements considered as metadata content elements except "title" tag. We delete the other tags because they do not contain useful visual information in next steps. The deleted tags are "base", "command", "link", "meta", "noscript", "script", and "style". We do not delete the "title" node because the textual information existing in this node might be useful for the user (the page title). We add also to this filtered group the following tags "comment", "br", and "doctype".

The second applied filter is Dead-Nodes-Filter, where it deletes all HTML nodes that do not affect on the appearance, for example nodes with height or width equals to "0px" (zero pixel); or nodes with style properties ("display:none" or "visibility:hidden" or "hid- den:true"). After applying the previous filters, we apply some reorganization rules in order to enhance visualizing the information in next phases. One example of these rules is Paragraph-Reorganization-Rule, where this rule re-constructs all paragraph child-nodes in one node which should contain the extracted sub-texts. We made this rule after analyzing many DOM structures, and observing that the text in some paragraph nodes is distributed between many child-nodes such as ⟨i ⟩ (italic), ⟨span⟩, ⟨b⟩ (bold), ⟨strong⟩, ⟨em⟩ (emphasized), ⟨small⟩, ⟨mark⟩(marked), ⟨del⟩ (deleted), ⟨ins⟩ (inserted), ⟨cite⟩ (defining a title of work), ⟨u⟩ (underline), and ⟨sub⟩ (subscript). So extracting these subtexts and collecting them in one text is useful and more efficient for visualizing them as one block in next phases rather than visualizing them as many separated blocks.

To illustrate the result of applying this rule, let us consider the following HTML source code:

```
<p>
    <a> this </a>
    <i> is </i>
    <mark> a </mark>
    <strong> simple </mark>
    <del> example. </del>
</p>
```

Its DOM structure is:



And after applying the rule, the DOM representation will be:

This rule is applied on all tags of type ⟨P⟩ (paragraph), and on all tags existed in the heading content category (specification of HTML5 content models), this group contains the following nodes (h1, h2, h3, h4, h5, h6, and hgroup). We also apply this rule on many other HTML tags which might contain textual child nodes such as ⟨a⟩ (hyperlink tag), ⟨abbr⟩ (abbreviation), ⟨acronym⟩ (this tag is not supported in HTML5), ⟨address⟩ (contact information for the author/owner), ⟨bdi⟩ (Bi- directional Isolation), ⟨button⟩, ⟨label⟩, ⟨li⟩ (list element), and ⟨q⟩ (quotation). We used Jsoup tool [8] in order to access to a web page DOM structure, and getting its HTML hierarchy. The result of this phase is a filtered DOM-tree; each of its nodes is visible and contains XPath and bounding box information. The designed filters and re-organization rules were integrated with our framework, and then we applied these rules and filters on the vision-based segmented web pages (154 pages mentioned pre-viously). After getting the filtered DOM-tree for each page, we represented the obtained bounding boxes on the used tablet Sam sung GALAXY Tab 2 (10.1inch, dimensions HeightXWidthXDepth 175.3X256.7X9.7 mm, Android version 4.1.2), after making a scaling for sizes of bounding boxes to be appropriate with the new size of used tablet.

C. Graph-Based Approach:

After segmenting the web page depending on its visual structures and analyzing its DOM-structure, we apply a new graph-based segmentation algorithm called "Blocks2Zones Clustering" in order to group many similar blocks together in one zone. Clustering many blocks together is necessary in order to decrease the number of viewed blocks in some interfaces (instead of viewing many blocks, we view one zone which represents these blocks and then the user can navigate intra-elements inside the zone by double clicking on the graphical element of the chosen zone), and to group closed blocks in one zone (here, closeness depends on distances between blocks, this will be described next sections in details). The pseudo-code of the proposed algorithm is:

[7] http://www.w3.org/TR/2011/WD-html5-20110525/content-models.html

[8] http://jsoup.org/

```
Blocks2Zones Clustering Algorithm
Input (Blocks, Number of desired Zones)
Output: Graph of N nodes (N Zones)
1- Transform the blocks into a graph
 (Undirected graph)
     1.1. Blocks --> Nodes,
     1.2. Make relations between the nodes,
     and assign weights for these relations.
2- If number of zones <= number of blocks
            end the algorithm,
 Else
 3- Find the smallest size node (node A)
 4- For node A, find the connection which has
    the largest weight (node B).
 5- Group the nodes A, and B (A+B).
 6- Repeat steps 3,4,5 till number of blocks
     ==  number of zones.
```

We can describe the output graph as following G=(V,E), where G is undirected graph, V is a set of vertices (nodes or zones), and E is a set of edges (connections between zones), and |V|=n (number of desired zones). We define the set of vertices V={v_i: 1≤i≤n where n is the number of desired zones, and v_i is set of sub-zones}. We define the set of connections as following E={e_j: $e_j(v_{j1},v_{j2})$: v_{j1} and v_{j2} are nodes in V}. To calculate the weights between nodes, we tested 2 relations of distances: the first one is Manhattan distance (d(p,q) = d(q,p)= $\parallel p-q \parallel = \sum_{i=1}^{n}$ |p_i-q_i |, and the second is Euclidian distance (d(p,q) = d(q,p)= $\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+.....+(p_n-q_n)^2}$)). Where p and q are centers of two nodes between them we want to calculate the distance. To ensure which distance should be used, we applied an internal quality criterion for the two used distance relations; the applied criterion is Sum of Squared Error (SSE) (SSE = $\sum_{K=1}^{n}\sum_{\forall X_i \in C_k}^{n} \parallel X_i - \alpha_i \parallel^2$ Where C_k is the set of instances in cluster K, and $\alpha_{k,j} = \frac{1}{N_K}\sum_{\forall x_i \in C_k}^{n} x_{i,j}$). Results of applying SSE measure on the two distances (Manhattan, and Euclidian) were identical; which means that using either Manhattan distance or Euclidian distance is equal in our algorithm to calculate weights between nodes.

In following, an example of applying this algorithm on the main page of web site w3schools.com is given. In this example we want that the algorithm segments the main page into 7 zones. The first iteration of the algorithm will divide the page into 13 zones (depending on the DOM structure) as illustrated in figure 1.a, and will construct a graph of 13 nodes as illustrated in figure 1.b.

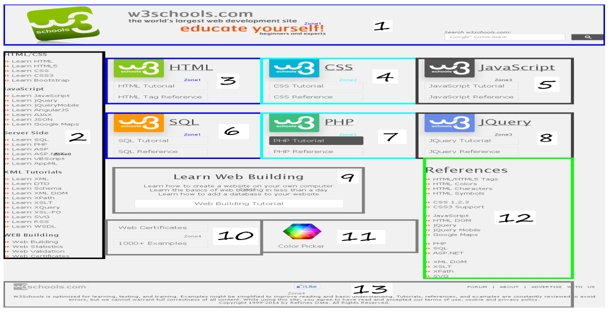


Figure 1.a Segmentation for 13 zones

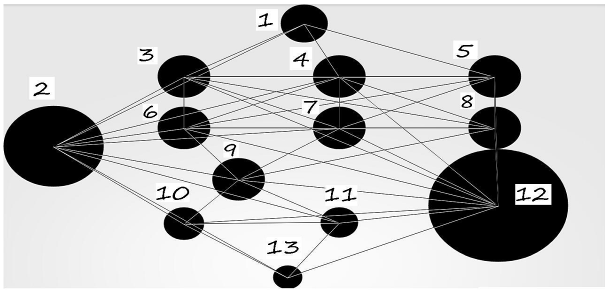
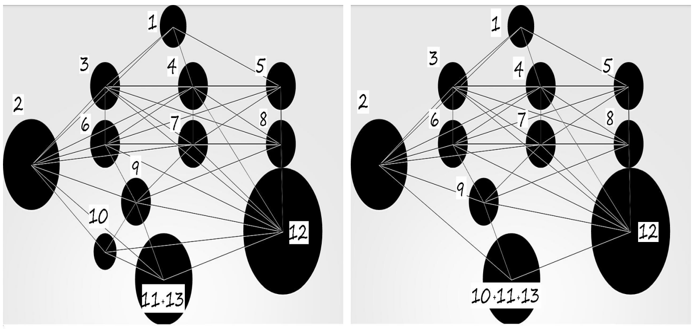


Figure 1.b Constructed graph for 13 zones

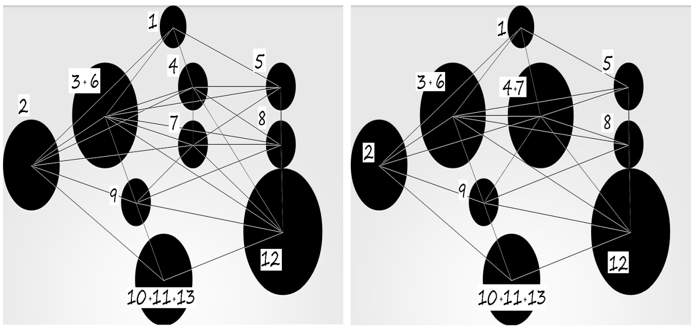
Figure1: First segmentation of www.w3schools.com

Figures 2, 3, and 4, illustrate next iterations of the algorithm and how the 13 zones are converted to 7 zones.



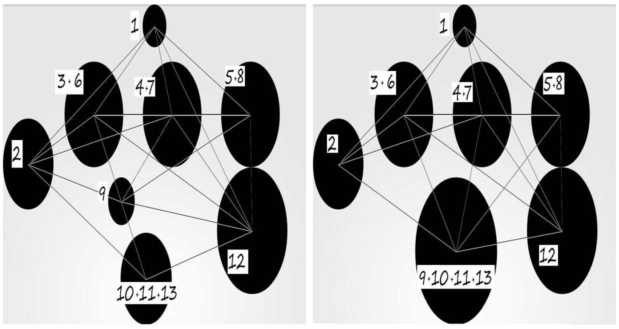
2.a 12 zones segmentation 2.b 11 zones

Figure2. 12, 11 zones segmentation of site www.w3schools.com



3.a 10 zones segmentation 3.b 9 zones

Figure3. 10, 9 zones segmentation of site www.w3schools.com



4.a 8 zones segmentation 4.b 7 zones
Figure4. 8, 7 zones segmentation of site www.w3schools.com

Applying this hybrid segmentation algorithm on a filtered DOM-tree (obtained from applying Vision-based approach then DOM-based approach) converts a web page to a set of zones, each zone contains many other zones or blocks, since each block represents a visual structure of HTML element and may contain many other blocks. The purpose of the proposed vibro-tactile access protocol is then to transform semantics of symbols in these zones, or blocks, or HTML elements into vibrations with different frequencies and amplitudes.

V. MANUAL AND AUTOMATIC WEB PAGE SEGMENTATION DIFFERENCES

In order to evaluate our algorithm results, and to know how users understand web layout structure based on their visual perception, we made an experiment: we asked 15 volunteers to make a manual segmentation of different kinds of web pages. The volunteers are different ages (between 25 and 50 years old), and most of them are informatics specialists. We presented for each volunteer 4 printed copies (A4 size papers) of 8 web pages (2 pages from www.cdiscount.com, 2 pages from www.photobox.com, 2 pages from www.rueducommerce.fr, one page from www.w3schools.com, and one page from www.leparisien.fr). We asked each volunteer to segment the 4 copies of each web page into 3, 4, 5 and 6 zones with following considerations:

- all the pages are printed in gray scale in order to avoid affecting the colors on the segmentation process. We chose this option because the current version of our algorithm does not depend on color differences between blocks,

- users can segment the page using polygons with minimum of 4 points (triangles are not allowed),

- Intersecting between segmented polygons are not allowed,

- users can start segmenting from any part or direction of the page (left, right, top, down, or center),

- we asked users to write the order number of zones (inside or beside the zone) while they make the segmentation process; this is very useful for us to know how the users start segmenting the pages, and how they finish it.

After collecting all manual segmented copies (480 papers : 15 users X 8 websites X 4 copies), we noticed the following:

- A lot of users do not start the segmentation process for certain number of zones by the same way they start segmenting for other number of zones,

- 16% of papers have been segmented starting from the center of the page, 3.5% have been segmented starting from the bottom of the page, and the majority of papers 80.5% have been segmented starting from top of the page,

- 20% of papers have been segmented vertically, and 80 have been segmented horizontally,

- 92.6% of segments are rectangles, and 7.4% of segments are polygons with more than 4 points,

- The most similar results between users were obtained for pages segmented in 3 zones, and the most different results between users were obtained for pages segmented in 6 zones.

- Finally, we noticed that it is very difficult to detect a segmentation method common between all users; since each user segments the pages depending on his understanding of the web page layout structure, on his visual perception of the visible elements, and on his interests and visual experience.

We also run our algorithm with the 8 mentioned web pages. The algorithm segmented each web page into 3, 4, 5, and 6 zones. The comparison between manual and automatic segmentation is illustrated in figures 5 and 6. Figure 5 illustrates matching results based on what we call "Strong criterion". Depending on this criterion, we can consider that 2 segmentation results are matched if the results are 100% identical without any difference. We can conclude from figure 5 that pages segmented automatically and manually into 4 zones are more matched than pages segmented into other numbers of zones. The percentage of identical matching depending on the strong criterion is 15.41% (74 identical matched results of 480 segmented copies).

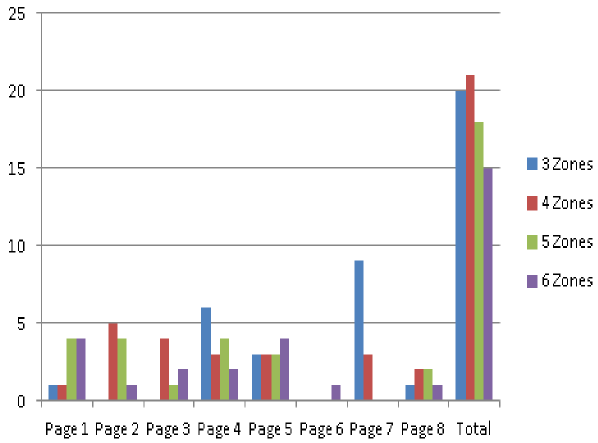


Figure5. Strong-criterion based matching results of manually and automatic segmentation

Figure 6 illustrates matching results based on what we call "Weak criterion". Depending on this criterion, we can consider that 2 segmentation results are matched if at least 50% of the results are identical.

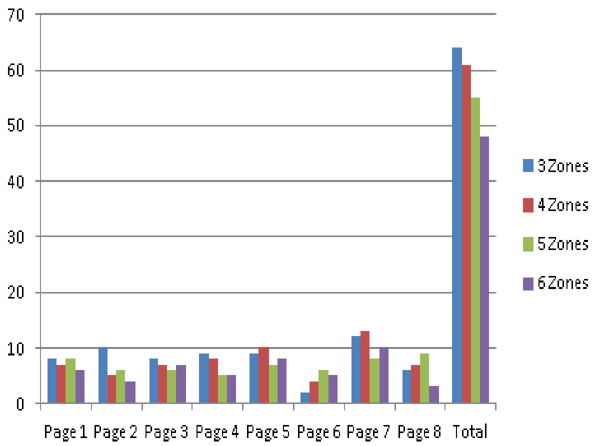


Figure6. Weak-criteria based matching results of manually and automatic segmentation

We can conclude from figure 6 that pages segmented automatically and manually into 3 zones are more matched than pages segmented into other numbers of zones. The matching percentage depending on the weak criterion is 47.5% (228 identical matched results of 480 segmented copies).

These results are promised considering that this is the first version of the algorithm, and especially that this version does not take into account the human visual perception aspects, and does not use image processing techniques which could be very useful in enhancing the achieved results.

VI. Conclusion and Perspectives

In this paper, we summarized our current work which aims to design an approach for non-visual access to web pages on touch-screen devices, and we focused on the suggested hybrid segmentation algorithm. We expect that integrating this method of segmentation with the designed vibro-tactile protocol can give VIP an impression of the first glance layout of web pages.

Next steps in this research will be 1) adding some image processing techniques in order to enhance the proposed segmentation algorithm 2) adding advanced techniques in text summarization to facilitate navigating textual information, 3) making real experiments to study effects of suggested segmentation algorithm on VIP web navigation models.

References

[1] F. Maurel, G. Dias, J-M. Routoure, M. Vautier, P. Beust, M. Molina, and C. Sann, *Haptic Perception of Document Structure for Visually Impaired People on Handled Devices* , Procedia Computer Science, Volume 14, Pages 319-329, ISSN : 1877-0509, 2012. DOI=http://dx.doi.org/10.1016/j.procs.2012.10.036

[2] F. Maurel, N. Vigouroux, M. Raynal, and B. Oriola, *Contribution of the Transmodality Concept to Improve Web Accessibility*, In Assistive Technology Research Series, Volume 12, 2003, Pages 186-193. International conference; 1st, Smart homes and health telematics; Independent living for persons with disabilities and elderly people. ISSN : 1383-813X. 2003.

[3] Alaeldin, Y. Mustafa, and B. Sharief, *Tactile WebNavigator Device for Blind and Visually Impaired People*, In Proceedings of the 2011 Jordan Conference on Applied Electrical Engineering and Computing Technologies, Jordan, 2011. DOI=http://dx.doi.org/10.1109/AEECT.2011.6132519

[4] M. Rotard, S. Knödler, and T. Ertl, *A Tactile Web Browser for the Visually Disabled*, In Proceedings of the sixteenth ACM Conference on Hypertext and Hypermedia. ACM, New York, NY, USA, 2005, pages 15-22, 2005. DOI= http://dx.doi.org/10.1145/1083356.1083361

[5] MY. BouIssa, M. Mojahid, B. Oriola, and N. Vigouroux, *Accessibility for the Blind, an Automated Audio/Tactile Description of Pictures in Digital Documents*, IEEE International Conference on Advances in Computational Tools for Engineering Applications, 2009, Pages: 591 – 594, 2009. DOI=http://dx.doi.org/10.1109/ACTEA.2009.5227855

[6] C. Lenay, O. Gapenne, S. Hanneton, C. Marque, and C. Genouëlle, *Sensory Substitution, Limits and Perspectives*, In Touch for Knowing Cognitive psychology of haptic manual perception, Amsterdam, Pages: 275-292, 2003,

[7] M. Tixier, C. Lenay, G. Le-Bihan, O. Gapenne, and D. Aubert, *Designing Interactive Content*

with Blind Users for a Perceptual Supplementation System,TEI 2013, 2013, in Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, Barcelona, Spain, Pages 229-236, 2013. DOI= http://dx.doi.org/10.1145/2460625.2460663,

[8] F. Maurel, W. Safi, P. Beust, and J.M. Routoure, *Navigation aveugle sur dispositifs mobiles : toucher le Web... pour mieux l'entendre*,16ème Colloque International sur le Document Électronique, CIDE16, Lille, France, Europia productions, 2013.

[9] A. Sanoja, and S. Gançarski, *Block-o-Matic: a Web Page Segmentation Tool*,BDA. Nantes, France. 2013. http://hal.archives-ouvertes.fr/hal-00881693/.

[10] C. Deng, Y. Shipeng, W. Ji-Rong, and M. Wei-Ying, *VIPS: a Vision-based Page Segmentation Algorithm*,Nov. 1, 2003, Technical Report MSR-TR-2003-79, Microsoft Research. 2003. http://research.microsoft.com/pubs/70027/tr-2003-79.pdf

[11] D. Cai, X. He, W-Y. Ma, J-R. Wen, and H. Zhang, *Organizing WWW Images Based on the Analysis Of Page Layout And Web Link Structure*,Microsoft Research Asia, Beijing, China, 2004. http://research.microsoft.com/pubs/69080/25.pdf

[12] J. Cao, B. Mao, and J. Luo, *A segmentation method for web page analysis using shrinking and dividing*,International Journal of Parallel, Emergent and Distributed Systems - Network and parallel computing, Volume 25 Issue 2, April 2010. Pages:93-104, 2010. DOI=http://dx.doi.org/10.1080/17445760802429585

[13] N. Foucault, S. Rosset, and G. Adda, *Pré-segmentation de pages web et sélection de documents pertinent en Questions-Réponses*,ITALN-RÉCITAL 2013.

[14] J.P. Callan, *Passage- level Evidence in Document Retrievals*, the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Pages: 302-310. Publisher: Springer-Verlag New York, Inc, 1994. http://dl.acm.org/citation.cfm?id=188589

[15] C. Kohlschütter, and W. Nejdl, *A Densitometric Approach to Web Page Segmentation*, USA, CIKM'08, Proceedings of the 17th ACM conference on Information and knowledge management, 2008. Pages: 1173-1182. DOI: http://dx.doi.org/10.1145/1458082.1458237

[16] D. Chakrabarti, R. Kumar, and K. Punera, *A graph theoretic approach to webpage segmentation*,Proceedings of the 17th international conference on World Wide Web, WWW'08, ACM, USA, 2008. Pages: 377-386. Publisher: ACM New York, NY, USA, 2008. DOI: http://dx.doi.org/10.1145/1367497.1367549

[17] X. Liu, H. Lin, and Y. Tian, *Segmenting Webpage with Gomory-Hu Tree Based Clustering*, Journal of Software, Vol 6, No 12, Pages: 2421-2425. 2011.

[18] W. Safi, F. Maurel, J-M. Routoure, P. Beust, and G. Dia, *Blind Browsing on Hand-Held Devices: Touching the Web to Understand it Better*, Data Visualization Workshop (DataWiz2014) associated to 25th ACM Conference on Hypertext and Social Media (HYPERTEXT 2014).