# Navigation Aveugle des Pages Web sur Dispositifs Mobiles

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

**Résumé.** De nombreuses techniques spécialisés aux personnes déficientes visuelles ont réussi à extraire les informations affichées sur des écrans numériques et ont réussi à transformer ces informations d'une manière linéaire, soit dans un format écrit sur des dispositifs spéciaux en Braille ou en une sortie vocale. Toutefois, les lecteurs d'écran ne transforment pas la structure 2-dimensionnel de la page web navigué. Dans cet article, nous proposons une nouvelle technique vise à renforcer la capacité des personnes déficientes visuelles à naviguer le Web en se concentrant sur l'amélioration de l'accès vibrotactile non-visuel des pages web sur dispositifs mobiles, basée sur l'extraction et la réorganisation de la structure de textes et les éléments graphiques des pages web, et de conversion automatique de ces structures visuelles et des informations textuelles dans des pages vibrantes utilisant un langage vibro-tactile graphique.

## Abstract.

**Blind Navigation of Web Pages on Touch-Screen Devices**

Many techniques specialized for Visually Impaired People succeeded to extract the information displayed on digital screens and succeeded to transform this information in a linear way either into a written format on special Braille devices, or into a vocal output using text to speech synthesizers. However, screen readers failed to transform the 2-dimentional structure of the navigated web page. In this paper, we propose a new technique aims to enhance the Visually Impaired People ability to navigate the Web by focusing on improving non-visual vibrotactile access to web pages on touch-screen devices, based on extraction and re-organization the structure of texts and graphical elements for web pages, reformatting and converting automatically these visual structures and textual information into vibrating pages using a graphical vibro-tactile language.

**Mots-clés :** Dispositifs mobiles, personnes déficientes visuelles, langage vibro-tactile graphique.
**Keywords:** Touch-screen devices, visually impaired people, vibro-tactile.

## 1 Introduction

In October 2013, the world health organization estimated that the number of (Visually Impaired People) VIP in the world is 285 million, 39 million of them are blind, and 246 million of them have low vision [1]. 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 Web, either by transforming a web page into a written Braille, or into a vocal output. Some screen readers installed on touch devices transform a web page into a Vocal-Tactile output. But there are some drawbacks for these proposed solutions: on the one hand, the Braille techniques are costly, and only few number of VIP have learned Braille (in France, there are about 77 000 visually impaired people and only 15 000 of them have learned Braille) [2]. On the other hand, transforming the information of a web page into a vocal format might not be suitable in public and noisy environments. Finally most of Braille solutions are not suitable for mobile devices (Maurel et al., 2012). In addition to these drawbacks, the most important one is the failure to transform the 2-D web page structure, because as reported by many authors, perception the 2D structure greatly improves navigation efficiency and memorizing the information because it allows high level reading strategies (rapid or cursory reading, finding or locating information,...) (Maurel et al., 2003). Our work focuses on developing and evaluating a sensory substitution system based on vibro-tactile solution which may solve the mentioned drawbacks.

[1] http://www.who.int
[2] http://www.opc.asso.fr/

Where we study how to increase the VIP perception of a 2-D web page structure, and how to enhance their techniques to navigate the Web on touch-screen devices. This suggested solution is very cheap comparing with prices of Braille devices, and also it could be more efficient in noisy and public environments comparing with vocal-tactile solutions. Our contribution is three-fold:

- 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 low frequencies tactile vibrations, and running a series of experiments with blind persons in order to validate our hypotheses,
- Designing an algorithm for segmenting web pages automatically 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
- Running a series of experiments with sighted persons in order to compare the differences between automatic and manual web page segmentation.

The paper is organized as following: first, in section 2 we view the state of the art for VIP targeted technologies, and we propose our own framework. The first pre-tests achieved with blind persons will then be described in the third section, and an analysis of results will be presented in the fourth section. In section 5, the state of the art for web pages segmentation methods is reviewed. In section 6, our supervised hybrid segmentation method is presented. In section 7, 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.

# 2 VIP Targeted Technologies and Proposed Framework

Current products for VIP such as screen readers depend mainly on speech synthesis or Braille solutions, such as ChromeVox[3], Windows-Eyes[4], and Jaws (Job Access With Speech)[5]. 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. Some screen readers can support a tactile feedback when working on touch devices, such as Mobile Accessibility[6], Talkback[7] for Android, and VoiceOver[8] for IPad. Many of these products propose shortcuts for the blind user to display a menu of HTML elements existed in the web page, for example headers, links, and images. But, the main drawback of all these products is that they transfer the web page information into a linear way, and without any indication for the global web page structure (2D layouts). Many researches tried to enhance the way by which VIP interact with web pages, such as (Alaeldin et al., 2012), that proposed a tactile web navigator to enable blind people to access the Internet. This navigator extracts texts from web pages, and sends these texts 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 (Rotard et al., 2005). This browser renders texts and graphics for VIP on a tactile graphics display, and it supports 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. Main drawback of this proposed system is that it needs a pin matrix device, which is expensive and cannot be integrated with handled devices. Another interesting model called MAP-RDF ("Model of Architecture of web Pages") (BouIssa et al., 2009) proposed a method to improve the accessibility to visual information for blind persons. This model allows representing the structure of a web page, and provides the blind users with an overview of the web page layout and the document structure semantics. The main drawback of this model is that it could be applied only on well structured web pages which contain meta-data, so it could not be applied to most web pages which rarely contain meta-data. Tactos is a suggested perceptual interaction system (Lenay et al., 2003). It 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), 3- computer (Tixier et al., 2013), The graphics tablet and the stylus allow the user to explore graphical contents on the screen such as circles, rectangles, and characters. While the user explores the contents, the system transforms pixels under the stylus into tactile stimulation on the Braille cells. 30 prototypes of Tactos have been released, to be used by a lot of users in many domains. Tactos has been successfully used to recognize simple and complex shapes.

First glance could be defined as the ability -in a blink of an eye- to understand the document layout and its structural semantics (Maurel et al., 2012). We aim of our work to increase the ability of visually impaired persons to understand the web page 2-dimentional layout in order to enhance their tactics to navigate the Web. A commercial tablet connected to a vibro tactile set-up is used for that. The first phase in our model is to extract visual structures in the navigated web page, and to convert these visual blocks to zones (segments) for facilitating the navigation in later phases. We achieve this phase depending on a hybrid segmentation method. Then the system will represent on the tablet screen the extracted visual elements as symbols using a graphical language (this language is under-development). The third phase is to

[3] http://www.chromevox.com/ [Access 01/03/2015] [4] http://www.synapseadaptive.com/gw/wineyes.htm[Access 1/03/2015]
[5] http://www.freedomscientific.com/ [Access 01/03/2015]
[6] https://play.google.com/store/apps/details?id=es.codefactory.android.app.ma.vocalizerfrfdemo&hl=fr [Access 01/03 /2014]
[7] https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback&hl=fr [Access 01/03/2014]
[8] http://www.apple.com/fr/accessibility/ [Access 01/03/2015]

browse these graphical symbols depending on size of the used touched screen device, and then in the fourth phase, our system provides a vibro-tactile feedback when the blind user touches the tablet. The intensity and the frequency of the vibration depends mainly on gray level under the finger. In this paper, we focus only on the first and fourth phases which specialize in extracting visual structures in the navigated web page, and in giving the user a vibro-tactile feedback by transforming light contrasts of touch-screen devices into low-frequencies tactile vibrations. To achieve the desired system, we have designed an electronic circuit which controls two micro-vibrators placed on the hands. A Bluetooth connection with an android tablet allows controlling the vibration intensity (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 gray level of touched points is transmitted to the embedded device in order to control the vibration intensity. Only one micro-vibrator was used for pre-tests described in this paper. Figure 1 illustrates the designed electronic circuit, and the used vibrator.

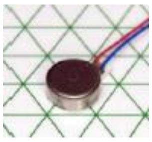

Figure 1. (a) The used micro-vibrator

Figure 1. (b) The designed electronic circuit

Figure 1. (a) The used micro-vibrator: the range of vibration frequency goes from 20 Hz up to 260 Hz. (b). The embedded system designed with the Bluetooth module.

# 3 Pre-tests Protocol

## 3.1 Objectives of Pre-tests

Our objective of the designed protocol is enhancing the ability of VIP to recognize the 2-D structure of a web page. In order to test the prototype mentioned in section 2, we designed some images contain different structures (detailed in section 3.2), and we tested the prototype firstly on 15 sighted persons (their eyes were closed) (Maurel et al., 2012), and later on 5 blind persons. Testing the protocol on sighted and blind persons gave us a more understanding of tactics and strategies achieved by sighted and blind persons to navigate the designed structures. This will be useful in designing the desired graphical vibro-tactile language (all results are detailed in next sections).

## 3.2 Designed Protocol for Vibro-Tactile Access

Each experiment (either for sighted or blind persons) consists of 4 ordered phases of training (learning task), and four ordered phases of evaluation (evaluation task). All the experiments were filmed, and the designed program stocked many parameters in log files (coordinates X, Y, pressure on the screen, and the time at each touch). Figure 2 presents the 4 images of training phases, and figure 3 presents the 4 images of evaluation phases.

Figure 2 (a). Image a of training task

Figure 2 (b). Images b (NT2), c (NT3), d (NTG)

Figure 2. Images of training task

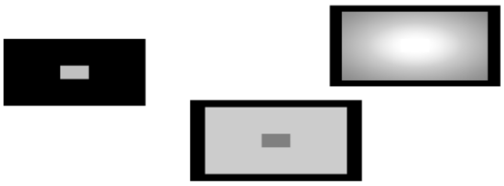

Figure 3 (a). Image a (Evaluation task)

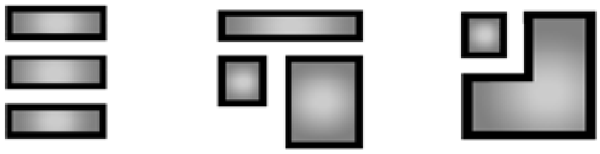

Figure 3 (b). Images b (IDP1), c (IDP2), d (IDP3)

Figure 3. Images of evaluation tasks.

In the training task, each user discovered firstly the graphical elements in each image presented in figure 2 (images a, b (NT2), c (NT3), and d (NTG)), and users were informed about names of graphical elements. The name of each image NT2, NT3, NTG, indicates how many transitions are necessary to access the square center, for example NT2 proposes 2 transitions to access the center of the square.

The evaluation task consists also of 4 phases, the first one allows to discover the image 3.a and to name each square inside it, then next phases are about images 3.b, 3.c, and 3.d, where we asked users to discover contents of each image, then to describe these contents, and to redraw discovered elements inside each image. We chose these images depending on following considerations:

- Image 3.a contains all squares on which users have trained in the training task, so it could test the ability to memorize and to distinguish the shapes.
- Image 3.b contains 3 rectangles with matched sizes and with vertical order, and the image 3.c contains 3 rectangles with different sizes and many relations of directions, so testing images 3.b, and 3.c could test the ability of distinguishing sizes, and distinguishing relations of directions.
- Image 3.d contains different shapes (a rectangle and a polygon), so it could test the ability to distinguish different shapes in the same image.
- The tested images contain examples of expected results of the segmentation process, so success of distinguishing these shapes by blind users could be an indicator of their ability to distinguish results of segmenting web pages.

The results of pretests with sighted persons were already published in (Maurel et al., 2012). Table 1 presents some results of the experiment for images NT2, NT3, NTG (the time required to distinguish graphical elements and the number of errors for the 15 sighted persons. The users have been asked to name the shapes in figure 3.a, and for each shape, we evaluated the number of correct and incorrect answers). In table 1, number or errors represents the number incorrect answers.

| Shape Name | Average Time in Seconds | Number of errors |
|---|---|---|
| NT2 | 28 | 2 |
| NT3 | 36 | 6 |
| NTG | 22 | 3 |

Table 1. Results of experiments of sighted persons for images NT2, NT3, NTG. (Table extracted from (Maurel et al., 2012)

We notice from table 1 that the lowest number of errors is assigned to image NT2, and the largest time and max number of errors is assigned to image NT3.

## 3.3 Experiment Steps with Blind Persons

The tests performed with each one of the 5 blind persons consisted of following: personal and technical questions, explanations of the test objective, a training task, and finally an evaluation task. The approximated average time for the test for each person is about 1 hour.

### 3.3.1 Personal and technical questions

Before starting the tests with the 5 blind persons, we asked them to support us with some information about their age and date of their blindness. Table 2 summarizes answers of personal questions.

| User-ID | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Age (Years) | 63 | 67 | 59 | 56 | 36 |
| Sex | Male | Female | Male | Female | Female |
| Date of the blindness | since birth | since 32 years | since 25 years | since 10 years | since 15 years |

Table 2. Personal information of the blind persons

We also asked users to provide us with some technical information about their experience in dealing with operating systems, screen readers, and what are the main problems when they navigate the Web. Table 3 shows a summary of answers for these technical questions. The first two columns indicate the number of operating systems (either Windows or Linux) used either on fixed or portable computers. The third, fourth, and fifth columns indicate the number of users who use JAWS (Job Access With Speech), NVDA (NonVisual Desktop Access), and ORCA, either on fixed or portable computers.

| | Windows | Linux | JAWS | NVDA | ORCA |
|---|---|---|---|---|---|
| Number of users with fixed computer | 4 | 0 | 4 | 2 | 0 |
| Number of users with portable computer | 2 | 1 | 2 | 1 | 1 |

Table 3. Used operating systems and screen readers

No one of the five blind persons uses a tablet, and the screen readers used with cellular phones are Talks and MobileSpeak with Nokia, and Voiceover with IPhone. Only one of the 5 persons uses a telephone to access the Web (access via IPhone). The main problems of accessing the Web via fixed or portable computers, or via IPhone telephone were: problems of access to Flash files, problems of AJAX technologies, and no ability to know the global structure of web sites. (These problems have been reported to us by the 5 users).

### 3.3.2 Explaining the objective of the test

To give the blind persons a good idea about the test, we explained in details what are the objectives and the phases of each tasks, and described contents of the embedded system; we also explained the final objectives of the project, and why we concentrate on vibro-tactile technique regardless of other techniques. This phase was important to initiate users for accepting kindly the test and for doing their best to interact with next steps as correctly as possible.

### 3.3.3 Training and learning task

In this training task, the user discovers the graphical elements in each image presented in figure 4 (images a, b (NT2), c (NT3), and d (NTG)), and the users were informed of each shape name. This task was very important for users to test the system before the evaluation task, and to know exactly how the system transforms different the grey level under the touched points on the tablet screen to a vibration mode. It is also very useful for users to control their speed of mapping the screen either to discover either the borders or the contents. During this task, the program recorded the touching information in log files (X, Y coordinates, Pressure, and Time). Table 4 indicates training times in minutes for each user, and for each image in figure2.

| User ID / Image | ID0 | ID1 | ID2 | ID3 | ID4 | Total | Average |
|---|---|---|---|---|---|---|---|
| A | 4,99 | 2,60 | 3,55 | 9,97 | 3,96 | 25,08 | 5,02 |
| b (NT2) | 4,40 | 3,11 | 0,99 | 3,30 | 1,00 | 12,80 | 2,56 |
| c (NT3) | 2,81 | 6,29 | 2,54 | 2,85 | 1,15 | 15,65 | 3,13 |
| d (NTG) | 3,11 | 3,98 | 2,02 | 3,22 | 1,32 | 13,65 | 2,73 |
| Total | 15,31 | 15,99 | 9,10 | 19,34 | 7,43 | | |

Table 4. Times of training task for each user (in minutes)

We notice from table 4 that discovering the first image takes more time, and it is normal because it is the first experiment for blind users on this prototype. We notice also that there is a significant decrease in time between discovering the first and the last image in the training task. This could be an indicator that training users could decrease the time for discovering graphical elements. We can also notice the significant difference between different tested persons, for example user with ID4 needed 7.43 minutes to scan the images (A, NT2, NT3, and NTG), but the user with ID3 needed 19.34 minutes to scan the same images.

### 3.3.4 Evaluation task

In this task, firstly we asked each user to discover the image 3.a and to find how many squares inside it and to name each founded square, then we asked them to discover images 3.b (IDP1), 3.c (IDP2), 3.d (IDP3), and to describe them to us, and to redraw discovered shapes. Table 5 illustrates an evaluation of answers for the first question to name squares in image 3.a (The blind users have been asked to name the shapes in image 3.a, and for each shape, we evaluated the number of correct and incorrect answers). (In tables 5 and 6, the symbol ✓ represents a correct answer for the touched shape, and the symbol X represents an incorrect answer or inability to select the name of the touched shape).

| User-ID/ Square Name | ID0 | ID1 | ID2 | ID3 | ID4 | Number of errors |
|---|---|---|---|---|---|---|
| NT2 | ✓ | ✓ | ✓ | X | ✓ | 1 |
| NT3 | ✓ | ✓ | X | X | ✓ | 2 |
| NTG | ✓ | X | ✓ | X | ✓ | 2 |

Table 5. Results of questions for squares in image 3.a

We notice from table 5 that the lowest number of errors is assigned to image NT2, and it is the same result which we obtained during tests with sighted persons. Results of answers for other questions related to images IDP1, IDP2, and IDP3 are summarized in table 6.

| User-ID | ID0 | | | ID1 | | | ID2 | | | ID3 | | | ID4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IDP | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Answers about number of rectangles | ✓ | ✓ | X | ✓ | X | X | ✓ | X | ✓ | X | ✓ | ✓ | ✓ | ✓ | ✓ |
| Answers about sizes of rectangles | X | X | X | X | X | X | ✓ | X | X | X | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 6. Answers of questions for images IDP1, IDP2, IDP3

We notice from data in tables 4, 5, and 6, that the best performance is for the user with ID4, and this may be because that this female user is the youngest between others, and it could be because that she was the only one that has already used touched devices (an IPhone in her case working with VoiceOver). After answering questions about each image of images (IDP1, IDP2, IDP3), we asked each user to redraw the graphical elements founded in each touched image. Figure 4 views the redrawing results of the user ID4 (ID4 is the female user who gave best answers).

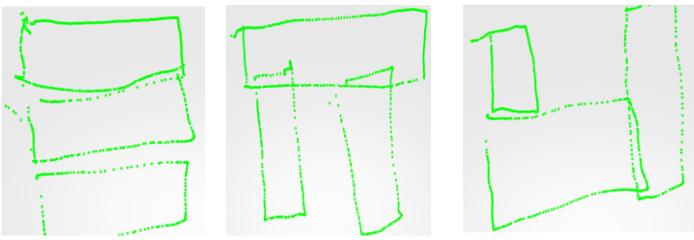

Figure 4. Results of redrawing images IDP1, IDP2, IDP3 for the user ID4

When comparing results of redrawing (Figure 4) with images IDP1, IDP2, and IDP3, we find that the results are interesting, and we can conclude the following:

1. An ability of distinguishing sizes of shapes, because the degree of scaling between redrawn shapes is nearly equal to the degree of scaling between real shapes (IDP1, IDP2, IDP3).
2. An ability of distinguishing relations of directions, because relations of directions (vertical order, left to, right to,…) between redrawn shapes is nearly equal to relations of directions between real shapes.

The average of times in minutes consumed for each evaluation question is summarized in table 7.

| User ID / Image | ID0 | ID1 | ID2 | ID3 | ID4 | Total | Average |
|---|---|---|---|---|---|---|---|
| A | 1,23 | 9,87 | 5,39 | 7,84 | 1,85 | 26,17 | 5,23 |
| IDP1 | 4,39 | 14,99 | 1,41 | 3,75 | 1,41 | 25,96 | 5,19 |
| IDP2 | 7,70 | 9,22 | 0,79 | 1,99 | 13,85 | 33,55 | 6,71 |
| IDP3 | 2,71 | 12,94 | 2,81 | 4,03 | 12,58 | 35,06 | 7,01 |
| Total | 16,03 | 47,02 | 10,39 | 17,61 | 29,68 | | |

Table 7. Times of the evaluation task for each user (in minutes)

# 4 Analysis of Results

To get an idea about the most touched and the least touched areas on the screen during learning and evaluation tasks, we divided the touched-screen into 16 areas (as in figure 5, r00...r03, r10...r13, r20...r23, r30...r33), and calculated the average of touches in each area for all users.

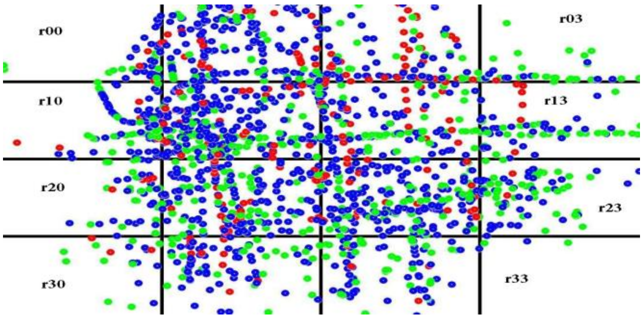


Figure 5. The 16 areas of the touched-screen of user with ID4 (Red points represent touched points with max pressure values, blue points represent touched points with pressure values less than the max and greater than the average, green points represent touched points with pressure values less than the average)

We have founded that the most touched areas are r12, r11, r22, r21, and the least touched areas are r30, r33, r32, r00 as described in figure 6. This information could be useful in next phases of our research in completing the graphical vibro-tactile language by putting the important information in the most touched areas.

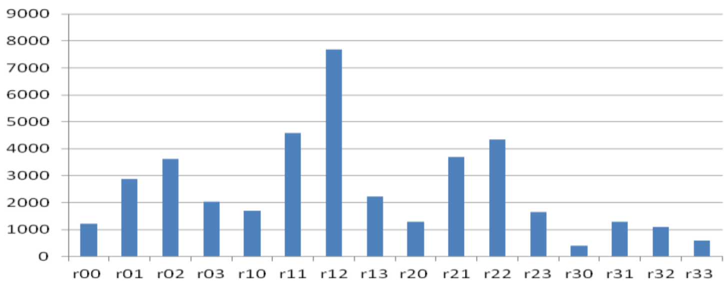


Figure 6. Most and least touched areas on the touched-screen

During analysis the results, we have noticed that there are a lot of differences between the pressure values for all users (Pressure value depends on the used tablet; in these experiments we have used Asus Model TF101 with Android operating system). To analyze pressure values, we calculated the max pressure value between all users, it was 3.19, and the average was 1.73, then we redrew the touched points for each user with considering that points with pressure values equal to the max value have been drawn in red color, points with pressure values less than the max and greater than the average have been drawn in blue color, and points with pressure values less than the average have been drawn in green color. Figure 5 represents an example of these points in different colors (for IDP3 evaluation task of the user who has id 4). After analyzing all the images drawn for all users, we have noticed that majority of red points (max pressure) are in images for which users gave right answers. This notice may be useful in designing our graphical vibro-tactile language, since we can observe when the user decreases or increases his touch pressure. The increasing of pressure may indicate that the user touches graphical elements interesting for him, and the decreasing may indicate that the user touches graphical elements non-interesting for him. During the tests we observed also that users try sometimes to scan the screen very quickly, it might be because they try to get a lot of information in a short time.

# 5 State of The Art for Segmenting Web Pages

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 have been suggested for segmenting web pages, such as:

- 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 (Sanoja et al., 2013) . In their paper, the authors determine first 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.

- 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) (Deng et al., 2003) is based on this approach.

- 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 sub-blocks (Cai et al., 2004) (Cao et al., 2010).

- 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 (Foucault et al., 2013).

- 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 ( Callan, 1994).

- Densitometric 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 ( Kohlschütter et al., 2008).

- 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 ( Chakrabarti et al., 2008) ( Liu et al., 2011).

# 6 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, vision-based segmentation, DOM-based segmentation, and graph-based segmentation.

## 6.1 Vision-Based Approach

In this phase, we render the web page using Selenium [9] 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 the 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 information about bounding boxes and DOM XPath for each HTML element. In next sections, we refer to

[9] http://www.seleniumhq.org/ [Access 01/03/2015]

bounding boxes by blocks (i.e. each bounding box represents an HTML element, and may contain other bounding boxes).

## 6.2 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) [10]. This specification divides the HTML tags into 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 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 "hidden:true"). After applying the previous filters, we apply some re-organization 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 contains 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 sub-texts 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. This rule was 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 11 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 previously). After getting the filtered DOM-tree for each page, we represented the obtained bounding boxes on the used tablet Samsung GALAXY Tab 2 (10.1inch, dimensions HeightXWidthXDepth 175.3X256.7X9.7 mm, Android version 4.1.2), after making a scaling of sizes of bounding boxes to be appropriate with the new size of used tablet.

## 6.3 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. The pseudo-code of the proposed algorithm is:

```
Blocks2Zones Clustering Algorithm
Input (Blocks, N° of desired Zones)
Output: Graph of N nodes (N Zones)
1- Transform the blocks into a graph (Un-Directed 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 node with the smallest size (node A)
     (size of node ==size of the rectangle bounds the node )
  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
```

[10] http://www.w3.org/TR/2011/WD-html5-20110525/content-models.html [Access 01/03/2015]
[11] http://jsoup.org/ [Access 01/03/2015]

The output graph is described 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 as 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={$e_j$: $e_j(v_{j1},v_{j2})$ : $v_{j1}$∈V, and $v_{j2}$∈V}. To calculate weights between nodes, we used the Euclidian distance between the centers of two nodes (center of a node is the center of the rectangle which bounds the node). 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 9 zones. The first iteration of the algorithm will divide the page into 13 zones (because there are 13 main nodes in the DOM structure – <Div> nodes) as illustrated in figure 7.a, and will construct a graph of 13 nodes as illustrated in figure 7.b.

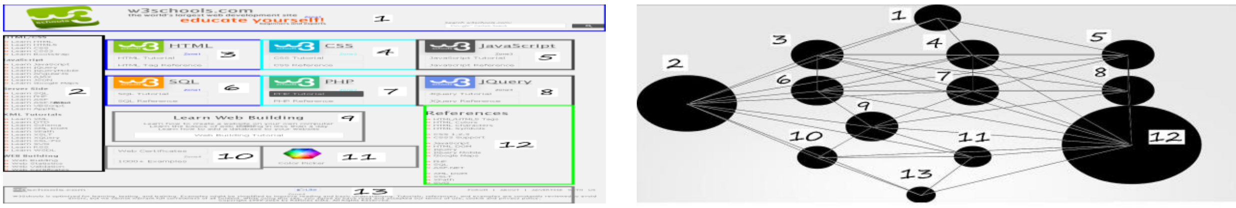


Figure 7.a Segmentation for 13 zones

Figure 7.b Constructed graph

Figure 7. First segmentation of www.w3schools.com

Figures 8, and 9, illustrate next iterations of the algorithm and how the 13 zones are converted to 9 zones.

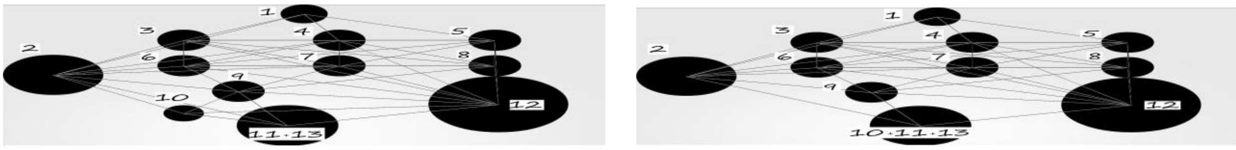


Figure 8.a 12 zones segmentation

Figure 8.b 11 zones segmentation

Figure 8. 12, 11 zones segmentation of w3schools.com

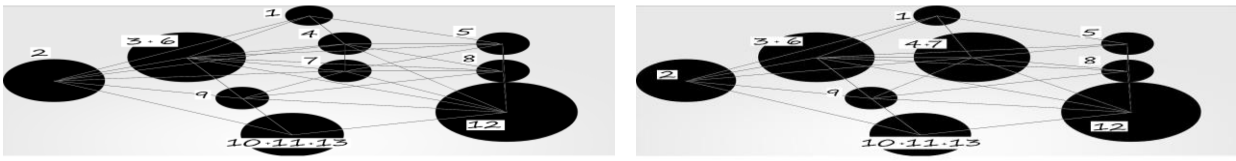


Figure 9.a 10 zones segmentation

Figure 9.b 9 zones segmentation

Figure 9. 10, 9 zones segmentation of w3schools.com

Applying this hybrid segmentation algorithm on a filtered DOM-tree (obtained from applying Vision-based approach and 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.

## 7 Manual and Automatic Web Page Segmentation Differences

In order to evaluate our algorithm results, and to know how users understand web layout structures based on their visual perception, we made an experiment as following: we asked to 15 volunteers a manual segmentation of different kinds of web pages. The volunteers were of different ages (between 25 and 50 years old), and most of them were 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, 1 page from www.w3schools.com, and 1 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),
- users can start segmenting from any part or direction of the page (left, right, top, down, or center),
- we asked users to write the ordering 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 end it.

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

- 70% of users do not start the segmentation process for certain number of zones by the same way they start segmenting for other numbers 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 pages,
- 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,
- 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 have 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 tables 8 and 9.

| | Page1 | Page2 | Page3 | Page4 | Page5 | Page6 | Page7 | Page8 | Total |
|---|---|---|---|---|---|---|---|---|---|
| 3 Zones | 1 | 0 | 0 | 6 | 3 | 0 | 9 | 1 | 20 |
| 4 Zones | 1 | 5 | 4 | 3 | 3 | 0 | 3 | 2 | 21 |
| 5 Zones | 4 | 4 | 1 | 4 | 3 | 0 | 0 | 2 | 18 |
| 6 Zones | 4 | 1 | 2 | 2 | 4 | 1 | 0 | 1 | 15 |
| Total | 10 | 10 | 7 | 15 | 13 | 1 | 12 | 6 | 74 |

Table 8. Strong-criterion based matching results of manual and automatic segmentation

Table 8 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 table 1 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 15.42% (74 identical matched results of 480 segmented copies).

Table 9 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. We can conclude from table 2 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 weak identical matched results of 480 segmented copies).

| | Page1 | Page2 | Page3 | Page4 | Page5 | Page6 | Page7 | Page8 | Total |
|---|---|---|---|---|---|---|---|---|---|
| 3 Zones | 8 | 10 | 8 | 9 | 9 | 2 | 12 | 6 | 64 |
| 4 Zones | 7 | 5 | 7 | 8 | 10 | 4 | 13 | 7 | 61 |
| 5 Zones | 8 | 6 | 6 | 5 | 7 | 6 | 8 | 9 | 55 |
| 6 Zones | 6 | 4 | 7 | 5 | 8 | 5 | 10 | 3 | 48 |
| Total | 29 | 25 | 28 | 27 | 34 | 17 | 43 | 25 | 228 |

Table 9. Weak-criteria based matching results of manual and automatic segmentation

# 8 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. The designed vibro-tactile protocol transforms the information viewed on the screen and touched by users to vibration by transforming light contrasts of touched pixels into low-frequencies tactile vibrations. The obtained results are interesting, since we used in these experiments only one vibration motor of low level quality (phone vibrator), and the learning period was very short, so there are many enhancements to be achieved in next versions either on the hardware/software level or on the level of learning phase (increasing the number and quality of micro-vibrators, making more control on frequencies and amplitudes sent to micro-vibrators, adding vocal abilities to the current approach, …). 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 with blind VIP to study effects of suggested segmentation algorithm on their web navigation models.

## References

Maurel F., Dias G., Routoure J-M., Vautier M., Beust P., Molina M., Sann C. (2012). Haptic Perception of Document Structure for Visually Impaired People on Handled Devices. *Procedia Computer Science*, Volume 14, Pages 319-329.

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

Alaeldin A., Mustafa Y., Sharief B. (2012). Tactile Web Navigator Device for Blind and Visually Impaired People. *In Proceedings of the 2011 Jordan Conference on Applied Electrical Engineering and Computing Technologies, Jordan,* Pages 1-5.

Rotard M., Knödler S., Ertl T. (2005). A Tactile Web Browser for the Visually Disabled. *In Proceedings of the sixteenth ACM Conference on Hypertext and Hypermedia. ACM, New York, NY, USA*. pages 15-22.

BouIssa Y., Mojahid M., Oriola B., Vigouroux N. (2009). 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*, Pages: 591 – 594.

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

Tixier M., Lenay C., Le-Bihan G., Gapenne O., Aubert D. (2013). Designing Interactive Content with Blind Users for a Perceptual Supplementation System. *In Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, Barcelona, Spain*, Pages 229-236.

Sanoja A., Gançarski S. (2013). Block-o-Matic: a Web Page Segmentation Tool. *La 29 édition des journées "Bases de Données Avancées".*

Deng C., Shipeng Y., Ji-Rong W., Wei-Ying M. (2003). VIPS: a Vision-based Page Segmentation Algorithm. *Technical Report MSR-TR-2003-79, Microsoft Research.*

Cai D., He X., Ma W-Y., Wen J-R., Zhang H. (2004). Organizing WWW Images Based on the Analysis Of Page Layout And Web Link Structure. *IEEE International Conference on Multimedia and Expo*. Volume 1, Pages 113-116.

Cao J., Mao B., Luo J. (2010). 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, Pages: 93-104.

Foucault N., Rosset S., Adda G. (2013). Pré-segmentation de pages web et sélection de documents pertinent en Questions-Réponses. *TALN-RÉCITAL-2013*. Pages 479-492.

Callan J.P (1994). Passage-level Evidence in Document Retrieval. *The Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin*. Pages: 302-310.

Kohlschütter C., Nejdl W. (2008). A Densitometric Approach to Web Page Segmentation. *Proceedings of the 17th ACM conference on Information and knowledge management*, Pages: 1173-1182.

Chakrabarti D., Kumar R., Punera K. (2008). A graph-theoretic approach to webpage segmentation. *Proceedings of the 17th international conference on World Wide Web, USA,* Pages: 377-386.

Liu X., Lin H., Tian Y. (2011). Segmenting Webpage with Gomory-Hu Tree Based Clustering, *International Journal of Software Engineering and Its Applications*, Volume 6, Pages: 2421-2425.