

DESIGN TACTILE INTERFACES WITH VIBRATION PATTERNS IN HTML5 FOR SMARTPHONE USERS WITH VISUAL IMPAIRMENTS

George Kokkonis¹, Vasileios Moysiadis², Theofano Kollatou³ and Sotirios Kontogiannis⁴

¹Dept of Business Administration, Western Macedonia University of Applied Sciences, Greece.

²Dept of Informatics and Telecommunications Engineering, University of Western Macedonia, Greece.

³Department of Electrical Engineering, Western Macedonia University of Applied Sciences, Greece.

⁴Department of Mathematics, University of Ioannina, Greece.

ABSTRACT

This paper describes the procedure for creating tactile interfaces for Android smart phones. It uses the HTML5 Vibration API and the Javascript programming language to create vibration patterns in order to increase the interaction between visually impaired people with their smart phones. Apart from vision, audio and haptics are used to maximize the Human – Computer Interaction between the Android phones and users with visual impairments. Three methods are proposed for the creation of tactile interfaces in smart phones, one is with the use of the HTML image map tag, and the other with the Mask R-CNN Artificial Intelligence algorithm. The Mean Opinion Score procedure is used in order to evaluate the proposed vibration patterns for Human - Computer Interaction Techniques.

KEYWORDS

Tactile interfaces, Haptics, Vibration patterns, HTML5, Vibration API, Mask R-CNN.

1. INTRODUCTION

Haptics, as a form of Human Computer Interaction have gain increasing attention the last years. Haptics offer a stimulus and immersive feeling to the user. It is one of the main human senses that help people to interact with their environment and explore their surroundings.

The use of smart phone devices has also been increased rapidly in the last decade. A smart phone, apart from video and audio, has also a touch screen for interaction with the user. The touch screen, with the help of vibration motors, has evolved to a tactile interface. Visually impaired people can exploit this vibrotactile interaction in order to distinguish objects inside an image or video displayed on the touch screen. The goal of this paper is to propose distinguishable vibrotactile patterns for object identification inside images for visual impaired people. The proposed method can also be used by all users, apart from blind ones, as vibrotactile interaction offers some unique characteristic compared to audio. It can be used in very noisy environments, or it can be used discretely without disturbing others, when quietness is required.

This paper, apart from vibration patterns, it also proposes three methods for assigning different vibration patterns to distinguished objects in an image that is fingered through a touch screen in a smart phone. The three methods that are proposed is the HTML `<map>` Tag with the `` “*usemap*” attribute and the Artificial Intelligence MASK R-CNN algorithm [1].

The rest of the paper is organized as follows. Section 2 presents the related work on vibrotactile patterns and smart phone applications for blind users. Section 3 analyses the role of haptics in Human Computer Interaction. Section 4 describes the android event listener and event handling. Section 5 analyses the HTML5 Vibration APi and the Javascript methods for creating vibration patterns. Section 6 presents the performance evaluation of the proposed methods and vibration patterns described in the previous sections. Finally, section 7 concludes the paper and states the future work.

2. RELATED WORK ON VIBRATION PATTERNS AND VIBROTACTILE APPLICATIONS

Vibration technology of mobile phones is the main approach to produce tactile patterns suitable for blind or visually impaired people. Tactile Vibration can be defined by adjusting and combining parameters such as frequency, pulse, intensity, and rhythm of the vibration. This section provides state-of-the-art related work on vibration patterns suitable for visually impaired people. It also discusses and analyzes the most common techniques used in mobile phones and tablets.

In [2] the authors are presenting a vibrotactile-based application for Android OS to help visually impaired children to learn geometry. The application tries to teach common geometric shapes to children through a game on a mobile phone. Vibration is used as primary feedback in the touch exploration of the object. A text-to-speech (TTS) system is also used to inform children about the available words which they should distinguish with the shapes on the screen.

In [3] the authors develop a mobile application for Android OS for educational proposes on Braille Arabic and French alphabet. In specific, the implemented application is aiming to teach Arabic and French alphabet to visually impaired children via Braille symbols on a smart phone screen by providing vibration and sound as feedback when the users touch specific areas.

The authors in [4] designed and implemented a mobile application named SparshJa, aiming to help visually impaired people on common functionalities like calling and saving contacts. The entire screen mimics the Braille cell which the user can use for alphanumeric data entry. The primary feedback is vibration along with audio. In specific, the authors used a vibration pattern technology called V-Braille [5] to distinguish vibration feedback with numbers and letters.

In [6] the authors developed EVIAC, a dedicated experimental protocol for testing the capacity of blind people to learn, identify and recognize objects on the screen based on vibration feedback. Their implementation has been tested on a tablet with Android OS and two different experiments used to evaluate their methods, recognizing basic shapes, and recognizing simple geometric objects. Finally, various experimental metrics were used to evaluate their work.

The authors in [7] propose two methods with tactile texture interaction to recognize numbers from a smart phone for visually impaired people. The first method uses left/right slider motion gestures (LRSM), while the second utilizes up to down slider motion gestures (UToDSM). Both of them use vibration feedback. They evaluate both methods along with a vibrotactile method, and the results show that UToDSM provides better interactive efficiency along with recognition accuracy.

3. HAPTICS IN HUMAN COMPUTER INTERACTION

Haptics refer to the sense of touch. It is one of the basic human's senses that help people to understand his/her surrounding and manipulate physical objects. Haptics play a crucial role in Human Computer Interaction, especial to people with visual impairments. It complements their impairment in vision and helps them to interact and understand their environment [8].

Haptics are usually divided in two categories. One corresponds to the tactile feeling that a hand receives when it touches or manipulates physical objects. It has direct correlation to the feedback that a person receives regarding the pressure, the vibration and the heat in his/her hand when he/she touches an object. The other category of haptics refers to the kinesthetic feedback that a user receives to his/her joints and muscles when he/she tries to manipulate or move heavier or bigger objects. In order to use haptic interactions, haptic interfaces are used.

Haptic interface is the system that helps the user to interact with a computer through his haptic sensations, and especially through his hand. Most haptic interfaces have haptic update rate 1000 sampling per second. This high update rate is due to the fact that the human sensation of touch, and in particular the perception of pressure, is quite sensitive to pressure differences. It can perceive differences in pressure for periods of less than one millisecond. In order to lower this high update rate, there are techniques that enforce a haptic controller that can lower the update rate up to 20 Hz [9]. This haptic controller can enforce differential encoding, quantization and adaptive packetization [10] in order to lower the update rate to acceptable levels. This is necessary, especially in smart phone applications, where the touch sampling rate is by far lower, equal to 60 Hz, and in rare phones equal to 120 Hz, see iPhone X. This means that most smart phones try to detect if the user touches the touch screen every 16,6 ms. More complicated haptic devices, as Haptic Touch [11], has a sampling rate of 1000 Hz. The smart phones have as haptic entry, the touch of the user to the touch screen. As a tactile feedback to the user interaction, the smart phone produces vibration with time duration patterns similar to those that are described in section 2.

4. ANDROID EVENT LISTENERS AND EVENT HANDLING

Android OS is widely adopted worldwide in mobile phones as well as in tablets, offering many applications for any usage. In addition, it is open source, based on Linux and was primarily designed for touch screen mobile devices like smart phones and tablets. It is offering many tools for developers that help them to easily implement features such as media enhancements, user experience, security, privacy, connectivity, and sharing.

This section analyzes the Event Listeners and Event Handling features of Android OS which is the primary approach to intercept events from user interaction in an application. In particular, every object from the interface can interact on actions like touch event on a button, leave focus of a specific control, or even key press from the key button of the smart phone. A simple explanation for event listeners and handlers would be that event listener watches for an event to be fired, while the event handler is responsible for dealing with the event.

An event listener is a callback method from the View class, which is called from the Android framework when an interaction is occurred in the user interface. Various callback functions are responsible for different actions. The most common callback function are, `onClick()`, `onLongClick()`, `onFocusChange()`, `onKey()`, `onTouch()`, and `onCreateContextMenu()`.

The `onClick()` callback function is called when the user touches an item on the user interface while `onLongClick()` is called when the user touches and holds for a while the item. The `onFocusChange()` callback function is called when a user moves in or out from a control and

change its focus status. The `onKey()` is called when a user presses a hardware key of the device when the specified control has the focus. The `onTouch()` callback function is called when an action occurs, as a touch event on the item. Finally, the `onCreateContextMenu()` callback function is called when a context menu is being built. The above callback functions should be registered in order to be active event listeners for a specific control.

Event Handlers in Android OS, are responsible for handling the events of the component from the user interface, and they are used in combination with event listeners. The most common callback methods for event handlers are, `onKeyDown(int, KeyEvent)`, `onKeyUp(int, KeyEvent)`, `onTrackballEvent(MotionEvent)`, `onTouchEvent(MotionEvent)`, and `onFocusChanged()`.

In more detail, the `onKeyDown(int, KeyEvent)` handler is called when a key press event occurs, while the `onKeyUp(int, KeyEvent)` handler is called after a key up event. The `onTrackballEvent(MotionEvent)` callback function is responsible for handling the trackball motion events and the `onTouchEvent(MotionEvent)` is responsible for the touch screen motion events. Finally, the `onFocusChanged(boolean, int, Rect)` callback function is the handler when an item of the user interface loses or gain focus.

Regarding the HTML5 event listeners, the `TouchEvent` listeners are used. They are divided to *touchstart*, when the user touches the touch screen, *touchend* when the user removes his finger from the touch screen, and *touchmove* when the user moves his finger along the touch screen.

5. HTML5 VIBRATION API AND JAVASCRIPT

In order the authors to apply the vibration patters described at section 2 to smart phones, they used the Vibration APi [12]. This API provides access the vibration mechanism of a smart phone. It is an HTML5 feature that uses patterns with on-off pulses. The pulses and the time gap between them may vary in duration and they are described with milliseconds. The method that controls a simple variation pulse is called through the procedure

Navigator.vibrate(duration_of_pulse_in_ms)

If the developer wants to enforce a vibration patterns then the parameter of the procedure will be replaced from a duration pattern such as the following:

Navigator.vibrate([duration_of_pulse1, time_gap, duration_of_pulse2, ...,])

The vibration duration pattern can be enforced with external looping if the vibration method is called through the javascript timing methods:

setInterval(vibration_event, time_gap) and *clearInterval(vibration_event)*

SetInterval(vibration_event, time_gap) executes the *vibration_event* periodically every *time_gap*. The *clearInterval(vibration_event)* stops the periodic execution of the *vibration_event*. The

Navigator.vibrate(0)

command also stops all the vibration events that are taking place at that moment. *SetInterval()* and *Navigator.vibrate()* are HTML Document Object Model (DOM) Window objects. The javascript uses the DOM to access the HTML web page and its elements. Through the DOM method, the vibration and as a consequence the tactile attribute can be anchored to every element on the webpage.

In order the authors to help blind users to distinguish graphical objects to a webpage in their smart phone, they correlated different vibration patterns to every object contained in the webpage. They followed three methods for distinguishing objects in an image:

A. HTML <map> Tag with the “usemap” attribute

In the first method the authors manually correlated the image areas with the HTML <map> Tag. They linked the html <image> with the <map> tag through the “usemap” attribute. Every image-map area had different id in order to be easily recognize the area id through the *document.elementFromPoint()* javascript method. The authors manually gave different vibration pattern to every map area *i* base on the following pattern.

$$duration_of_pulse_in_ms_i = i * 100 \tag{1}$$

$$Timegap_i \text{ between vibrations} = \begin{cases} 100 * (10 - i) , & \text{for } i \leq 9 \\ 100 & , \text{for } i > 9 \end{cases} \tag{2}$$

From equations (1) and (2) it is understood that if the different objects in the image are fewer than 10, the period for every vibration pattern is equal to one second.

B. HTML <map> Tag with the “usemap” attribute and audio enhancement

As an extension of the previous method, the authors added and audio messages in every image area, as additional guidance to users with visual impairments. The audio message describes the object that is being touched by the user.

C. TensorFlow AI and Mask R-CNN object identification

In order to automatically identify objects in images the authors used the Artificial Intelligent algorithm TensorFlow [13]. On top of TensorFlow the authors used the open source neural network library written in Python named Keras [14]. It was used for imagery features extraction and models creation. The algorithm for object identification and instance segmentation named Mask R-CNN [15] was used, in order to cover every identified object with different color, as shown in figure 1.

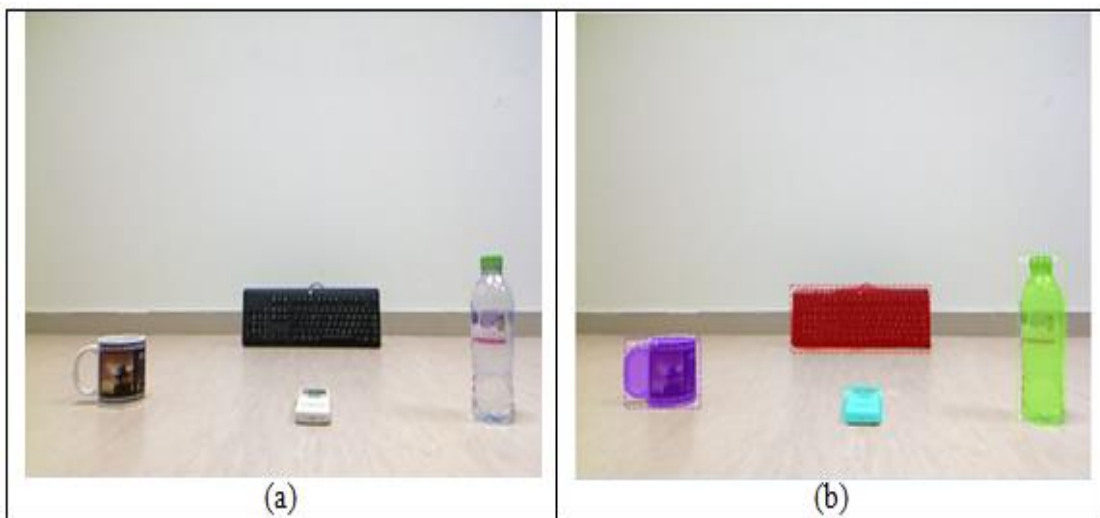


Figure 1– Object recognition with The Mask R-CNN algorithm.

The vibration pattern that was assigned to each object was depended to the color of the assigned to the object by the Mask R-CNN algorithm. The correlation between the color of the object and the vibration pattern is described in algorithm I.

Algorithm I - Pseudocode of the assignment vibration pattern to color

1. Read finger's position on touch screen
2. Read the colour of the image in the same coordinates of the figure (getImageData(x, y, 1, 1).data)
3. Transform the mean RGB colour data to vibration pattern through the equation $(\text{pixelData}[0] + \text{pixelData}[1] + \text{pixelData}[2]) / 3$
4. Duration of vibration pulse in ms = $((\text{pixelData}[0] + \text{pixelData}[1] + \text{pixelData}[2]) / 3) * 4$
5. Time gap between vibrations = 1040 - Duration of vibration pulse in ms

6. PERFORMANCE EVALUATION OF VIBRATION PATTERNS

For the performance evaluation of the three methods of section 5, the User Experience (UX) of the smart phone users was taken into consideration. The Mean Opinion Score (MOS) metric was used to evaluate the three methods. MOS measures the quality of experience of the user when he/she uses the vibration patterns described in section 5. The test users were 20 postgraduate students of Business Administration at the age between 18 and 24. Ten of them were male and the other ten female. The smart phone that was used was the Xiaomi Redmi Note 4. Before the test, they were explained of the vibration patterns, and they were given 5 minutes to explore and get familiar with the mobile application.

The test users were asked to identify the position of the objects of the touch screen while having their eyes closed. After the test, the students were asked three questions regarding the effectiveness, the efficiency and the Satisfaction of the mobile application. The metrics were: A) If they were able to complete the task (effectiveness), b) if it was easy for them to complete task (efficiency) and C) How convenient was this experience (Satisfaction). The scale for the answers was from 1 to 5. Value 1 corresponds to the lowest perceived quality, while value 5 corresponds to the highest perceived quality, as described in Table 1.

Table1–Scaling points for evaluation of effectiveness, efficiency and Satisfaction.

Rating	Label
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

The Mean Opinion Score from the User Experience evaluation are depicted in Table 2.

Table2–User Experience Score from the three proposed methods

User Experience	HTML <map> tag and “usemap” attribure	Audio messages with HTML <map> tag and “usemap” attribure	Mask R-CNN object identification and automatic vibration pattern assignment
Effectiveness	4.0	4.5	3.1
Efficiency	3.9	4.4	3.2
Satisfaction	4.0	4.6	3.5

From table 2 it is understood that the manual HTML `<map>` tag method with the `` usemap attribute usage is more effective, efficient and satisfying than the automatic Mask R-CNN object identification algorithm with automatic pattern vibration assignment. When vibrotactile feedback is accompanied with audio messages, then the User Experience is even higher. In order to make the Mask R-CNN object identification algorithm more User friendly, it should change the colors that it uses when it covers the objects. It should change the transparency to lower percentages and pick colors with greater Hue deviation when referring to the HSL color model. This means that is preferred to use the Hue value of the HSL model rather than the mean value of (r, g, b) of the RGB model. A good and simple library for parsing from one color model to the other in javascript is the tinycolor [16] library.

7. CONCLUSIONS

This paper proposes three different methods for increasing the tactile interaction between blind people and smart phones. All methods use the HTML5 and the Vibration API, in order to transform the touch of the smart phone user on the touch screen to a vibration feedback. In the first two methods, the HTML5 `` "usemap" attribute is used, in order to assign to each image section different vibration patterns. The third method uses the Artificial Intelligence Mask R-CNN algorithm to automatically recognize objects in an image. An overlay with different color is placed over the recognized images. Different vibration patterns are proposed and assigned to each color. The user experience evaluation revealed that the manual segmentation of images produces better interaction results to users, but it is more time and effort consuming. For future work, the authors intent to modify the Mask R-CNN algorithm, in order to assign selected colors with greater Hue difference to each recognized object, in order to create vibration patterns with more easily distinct interaction feedback.

REFERENCES

- [1] G. Gkioxari, P. Dollár and R. Girshick K. He, "Mask R-CNN," in IEEE International conference on Computer Vision (ICCV), 2017.
- [2] M. C., Buzzi, M., Leporini, B., & Senette, C. Buzzi, "Playing with geometry: a multimodal android app for blind children," Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter ACM., pp. 134-137, Sept. 2015.
- [3] A., & Soufi, M. Bouraoui, "Br'Eye: An Android Mobile Application to Teach Arabic and French Braille Alphabets to Blind Children in Tunisia," International Conference on Computers Helping People with Special Needs Springer, pp. 357-364, Jul. 2018.
- [4] P., Pimpalkar, N., Sawke, N., & Swain, D. Gokhale, "SparshJa: A User-Centric Mobile Application Designed for Visually Impaired," Computational Intelligence in Data Mining Springer, Singapore., pp. 405-414, 2017.
- [5] C., Acuario, C., Johnson, W., Hollier, J., & Ladner, R. E. Jayant, "V-braille: haptic braille perception using a touch-screen and vibration on mobile phones," ASSETS, no. 10, pp. 295-296, Oct. 2010.
- [6] J., Issa, Y. B., & Chbeir, R. Tekli, "Evaluating touch-screen vibration modality for blind users to access simple shapes and graphics," International Journal of Human-Computer Studies, vol. 110, pp. 115-133, 2018.
- [7] F., Chu, S., Ji, N., & Pan, R. Zhang, "Design and Evaluation of Tactile Number Reading Methods on Smartphones," IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS) IEEE, pp. 531-536, Jun. 2018.

- [8] K. Psannis, C. Asiminidis, S. Kontogiannis G. Kokkonis, "Design Tactile Interfaces with Enhanced Depth Images With Patterns and Textures for Visually Impaired People," *International Journal of Trend in Scientific Research and Development*, vol. 3, no. 1, Dec 2018.
- [9] S., Ishii, M., Koike, Y., & Sato, M. Hasegawa, "Inter-process communication for force display of dynamic virtual worl," *ASME DYN SYST CONTROL DIV PUBL DSC.* , no. 67, pp. 211-218, 1999.
- [10] G., Psannis, K. E., & Roumeliotis, M. Kokkonis, "Network adaptive flow control algorithm for haptic data over the internet–NAFCAH.," *International Conference on Genetic and Evolutionary Computing* Springer, pp. 93-102, August 2015.
- [11] 3D Systems. Touch. (2018, Dec) <https://www.3dsystems.com/haptics-devices/touch>.
- [12] S., Psannis, K. E., & Sifaleras, A. Xinogalos, "Recent advances delivered by HTML 5 in mobile cloud computing applications: a survey," In *Proceedings of the Fifth Balkan Conference in Informatics ACM*, pp. 199-204, Sept. 2012.
- [13] M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., & Kudlur, M. Abadi, "Tensorflow: A system for large-scale machine learning," 12th
- [14] Keras: The Python Deep Learning library. (2017) [Online]. <https://keras.io/>
- [15] W. Abdulla. (2017) Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. [Online]. https://github.com/matterport/Mask_RCNN
- [16] Antoine Pairet. (2017) github.com/bgrins/TinyColor. [Online]. <https://github.com/bgrins/TinyColor>

AUTHORS

Dr. George Kokkonis is a scientific staff at the Dept. of Business Administration of Western Macedonia University of Applied Sciences, Greece. He received his PhD from Department of Applied Informatics, University of Macedonia, Greece. He received a five-year Eng. Diploma from the Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki and his MSc in Information Systems from the University of Macedonia. He has been working as a Lecturer for the last 12 years at the Dept. of Computer Applications in Management and Economics, Western Macedonia University of Applied Sciences, Grevena, Greece. He has been teaching the Laboratory Lessons of Multimedia Systems, Pc Programming and Data Bases. For the last 5 years is responsible of the NOC of the branch of the TEI of West Macedonia in the city of Grevena, Greece. His research interests include haptic protocols and interfaces, teleoperation and human–computer interaction.



Vasileios Moysiadis received the Diploma degree from the Department of Informatics, Aristotle University of Thessaloniki, Greece, in 1998, and MSc degree in Information Systems, University of Macedonia, Greece, in 2008. He is now a Ph.D. student in the Department of Informatics and Telecommunications Engineering, University of Western Macedonia, Greece. His research interests include low power wide area networks, fog computing, distributed computing, image processing, and haptic protocols. Currently, he is working as a teacher in secondary education and as a research associate at the University of Western Macedonia, Greece.



Dr. Theofano Kollatou was born in Volos, Greece. She received her Dipl.-Eng. degree and the Ph.D. degree from the ECE Department of the Aristotle University of Thessaloniki, in 2005 and 2014 respectively. She is currently researcher at the Department of Electrical Engineering of the TEIWM, Kozani, Greece and her research interests include use of Metamaterials in electromagnetic compatibility applications and smart grids communications.



Dr. Sotirios Kontogiannis graduated from Democritus University of Thrace, Department of Electrical and Computer Engineering. He received an MSc in Software Engineering and PhD in the research area of algorithms and network protocols for distributed systems, from the same department. He worked as a software developer for more than ten years in the private sector .and participated into SME research and development projects. We also worked as a contract assistant professor for TEI of Western Macedonia, Dept. of Business administration for six years and as a contract lecturer at the University of Western Macedonia, Dept. of Informatics & Telecommunications Eng. His research interests focus on the areas of distributed systems, artificial intelligence, AI algorithms, sensor networks, middleware protocols and computer networks. He calls himself as the mad-eye researcher. He is currently a scientific staff member and director of the Distributed micro-computers laboratory (<http://kalipso.math.uoi.gr/microlab>), at the Applied Mathematics and Engineering research section of the Department of Mathematics, University of Ioannina. His personal web-page is at <http://spooky.math.uoi.gr/~skontog> and his e-mail is: skontog at cc.uoi.gr.

