AudioFunctions.web: Multimodal Exploration of Mathematical Function Graphs

Dragan Ahmetovic
Università degli Studi di Torino
Dipartimento di Matematica
dragan.ahmetovic@unito.it

Cristian Bernareggi
Università degli Studi di Milano
Dipartimento di Informatica
cristian.bernareggi@unimi.it

João Guerreiro
Carnegie Mellon University
Robotics Institute
jpvguerreiro@cmu.edu

Sergio Mascetti
Università degli Studi di Milano
Dipartimento di Informatica
sergio.mascetti@unimi.it

Anna Capietto
Università degli Studi di Torino
Dipartimento di Matematica
anna.capietto@unito.it

ABSTRACT

We present *AudioFunctions.web*, a web app that uses sonification, earcons and speech synthesis to enable blind people to explore mathematical function graphs. The system is designed for personalized access through different interfaces (touchscreen, keyboard, touchpad and mouse) on both mobile and traditional devices, in order to better adapt to different user abilities and preferences. It is also publicly available as a web service and can be directly accessed from the teaching material through a hypertext link.

An experimental evaluation with 13 visually impaired participants highlights that, while the usability of all the presented interaction modalities is high, users with different abilities prefer different interfaces to interact with the system. It is also shown that users with higher level of mathematical education are capable of better adapting to interaction modalities considered more difficult by others.

CCS Concepts

•Human-centered computing → Usability testing; *Auditory feedback; Keyboards; Pointing devices; Touch screens; Accessibility systems and tools;* **•Social and professional topics → Assistive technologies; People with disabilities;**

Keywords

Visual Impairments and Blindness, Mathematics, Function graphs

1. INTRODUCTION

Mathematics accessibility is of paramount importance for people with visual impairments or blindness (VIB) who study or work within STEM (Science, Technology, Engineering, Mathematics). At the same time, maths accessibility is particularly challenging because many mathematical concepts are better conveyed through bi-dimensional representations, such as visual sets drawings in early education, or function graphs, typically used in higher education.

W4A '19, May 13–15, 2019, San Francisco, CA, USA

ISBN 978-1-4503-6716-5/19/05…$15.00

DOI: https://doi.org/10.1145/3315002.3317560

Such representations are convenient for sighted people because they provide an overview of the represented function and its global properties (*e.g.*, raising trend). At the same time they allow detailed exploration and convey quantitative information, such as the value of the function for $x = 0$. Instead, for VIB people, bi-dimensional representations are difficult to convey, because they cannot be accessed through linear scanning (*e.g.*, with a refreshable Braille display).

Traditional solutions to this problem are physical tactile supports, such as embossed drawings, which however cannot be easily changed once created. Haptic interfaces also exist, but they are expensive, need to be programmed, and require supervision. Desktop or mobile software for sound-driven exploration have been investigated with promising results. However, no existing solution was designed to adapt to different user abilities and preferences through personalized and diversified interaction. Furthermore, graph exploration is most needed while studying maths, but none of the existing solutions is designed to be accessed directly from maths documents.

In this paper we present *AudioFunctions.web*, a web app that enables VIB users to explore function graphs. It adopts a combination of sonification, earcons [7] and speech, similar to *AudioFunctions*, an earlier iPad prototype [33]. However, the interaction is further improved to provide quick understanding of the graph global characteristics, as well as detailed exploration of its quantitative information on different interfaces and platforms. *AudioFunctions.web* main characteristics that differentiate it from prior work are:

- It is platform-independent, accessible on traditional and mobile devices through different interfaces (touchscreen, keyboard, touchpad, mouse), which is relevant for personalizing the system to different user preferences and abilities.
- It is publicly available under a Creative Commons BY-NC-SA license[1]. It can be linked directly from the teaching material (*e.g.*, PDF or web pages), which makes the approach practical and accessible anywhere, without additional software.

This paper presents two main contributions: 1) we describe the design principles behind *AudioFunctions.web* and its technical implementation; 2) we conduct an experimental evaluation with 13 VIB users, to assess the system usability with different interfaces. The participants were capable of using the system with all interaction modalities and they evaluated them positively. We also highlight that user abilities and maths education influenced their appreciation and capability with different interaction modalities. We discuss why this is a particularly relevant result to drive future system development.

---

[1] https://creativecommons.org/licenses/by-nc-sa/4.0/

2. RELATED WORK

Being able to access mathematical content is one of the major challenges for VIB people who want to study or conduct research in STEM subjects [21, 3]. Indeed, it is difficult to render multi-dimensional mathematical content, such as graphs, visual representations or formulae using traditional assistive technologies (*i.e.*, screen readers and refreshable Braille displays). Previous research has explored assistive technologies for math learning across different education levels, from elementary school [15, 1] to university [11], also considering the problem of adaptation to different devices [9]. In particular, among the investigated problems, prior works have addressed the issue of accessing mathematical formulae on web pages [10, 31] and within digital documents such as PDF files generated from LaTeX [2, 4]. Other works have focused on the exploration of arbitrary drawings using proprioceptive exploration and sonification on tablet devices [16].

For what concerns the accessibility of mathematical function graphs, prior work have investigated how to extract information from graph images [23] and text, and how to convey graph data using textual descriptions [27, 17], tactile graphs on paper, haptic instruments and sonification approaches. Regarding tactile graph exploration, effective techniques to emboss function on paper have been extensively investigated [12, 24, 25]. Other research has focused on methods to enrich tactile graphics with additional information (*e.g.*, labels, captions) [32, 5, 28, 18].

Even though tactile representations of function graphs can be straightforwardly explored by sliding the finger over the embossed paper, some issues remain. First, a tactile image is static. Hence, once embossed, it cannot be edited or changed (*e.g.*, enlarged/reduced) for more comfortable exploration. Second, on a standard paper sheet, only a limited amount of tactile graphical elements can be embossed (*e.g.*, one or two lines in one coordinate system). Third, it is difficult to integrate tactile resources with digital documents (*e.g.*, within exercise books or in the notes) since tactile images need to be explored on paper while the digital document is read through a screen reader or on a refreshable Braille display.

Concerning haptic solutions, research has primarily focused on designing and developing multimodal systems for enabling exploration of function graphs [29], statistical diagrams [22], directed and undirected graphs [6]. These systems present three main advantages. First, the graph can be manipulated while being touched in a virtually infinite workspace. For example the graph can be rotated/translated, enlarged/reduced and the window can be scrolled in any direction according to the exploration needs. Second, labels can be added to the haptic exploration through speech or audio cues. Therefore, unlike paper-based solutions, labels do not overlap with the exploration. Third, to the purpose of giving a global view of the graph, the hand of the user can be guided along the curve by the force generated by the arm of the haptic device. Nonetheless, in order to obtain these advantages, high resolution haptic devices are necessary. However, such devices are not portable, and since they are not widespread they are also expensive and therefore not accessible to the majority of visually impaired users.

Finally, approaches using sonification, that is the auditory representation of information by modulating sound properties, has been investigated to provide non-visual access to geometric shapes [26], maps [19] and especially to graphs of mathematical functions. Gardner et al. [14] propose *Audio Graphing Calculator*, a desktop application that sonifies a function graph, reproducing it as a sequence of sounds. This approach maps function coordinates to sound frequencies, and evaluations with blind people give evidence of the effectiveness of this sonification approach especially for understanding the trend of a function graph.

Web-based tools have been proposed to provide similar sonification capabilities. *Audio Graphing Calculator*[2] enables exploration of the trend of function diagrams through sound and *SAS Graphics Accelerator*[3] enables non-visual exploration of bar charts, heat maps, line charts, scatter plots and histograms through audio feedback. However these solutions sonify information as a predefined sequence, without real-time proprioceptive exploration. *Desmos*[4] is another web-based approach, which instead leverages touch based interaction to sonify function diagrams on touchscreen devices. However it requires to trace the function graph on the screen in order to sonify its value, which is difficult for blind users.

Taibbi et al. [33] propose *Audio Functions*, an iPad application that leverages proprioception and sonification to explore a function graph on a tablet. *Audio Functions* enables the blind person to access a function diagram through three exploration strategies: by listening to the sonified diagram, by sliding the finger over the touchscreen following the sonified curve and sliding the finger along a horizontal bar (i.e. the x-axis) and listen to the sonified function value. Further information about the function diagram (*e.g.*, concavity and point coordinates) are provided on demand through speech. The evaluation shows that the combination of sonification, proprioception and speech messages significantly improves the construction of the mental image of a diagram compared to approaches like Audio Graphing Calculator which provide sequential information access.

Analogously, Goncu et al. [17] introduce graCALC, a graphing calculator that sonifies function graphs and statistical diagrams and enables exploration on a touch device. In addition to *Audio Functions*, *graCALC* adds an overview containing a sonification of the graph and an automatically generated verbal description in order to help the blind person in the initial navigation stage.

Existing solutions provide a one-size-fits-all interface, and cannot adapt to different users' preferences and capabilities. Furthermore, they are not designed to enable access to graphs directly from digital documents (*e.g.*, within a PDF file of an exercise book), which limits their practical use. Instead, *AudioFunctions.web* is implemented as a web app and can be invoked by URL from any digital document. It also promotes personalization since it provides platform-independent exploration through multiple interaction modalities.

3. AUDIOFUNCTIONS.WEB DESIGN

We designed *AudioFunctions.web* with the following objectives:

1. Allow global overview as well as precise analytical exploration of a given function graph.
2. Provide platform-independent access to graphs, on mobile and traditional devices with different hardware characteristics.
3. Enable exploration with different interfaces (touchpad, mouse, keyboard, touchscreen) based on user needs and preferences.
4. Access to function graphs directly from digital documents such as textbooks and scientific papers, and from web pages.

To satisfy these objectives, *AudioFunctions.web* is designed as a platform independent web app that can be accessed through touchscreen, touchpad/mouse and keyboard interfaces. It enables the exploration of function graphs by sensory substitution of the function values through sonification, supported by additional verbal or sound earcons. It further allows precise point exploration, conveying function values and derived quantities through verbal messages.

---

[2] `http://www.viewplus.com`

[3] `http://support.sas.com/software/products/graphics-accelerator/`

[4] `http://www.desmos.com`

3.1 Graph exploration

AudioFunctions.web defines a *sensor point* $s = (x_s, y_s)$[5] positioned at the coordinates currently explored by the user inside the viewport area which covers the whole browser window. The viewport width spans between (x_{min}, x_{max}) and its height spans between (y_{min}, y_{max}). The scale of the graph adapts to devices with different screen resolution in order to keep the same proportions and range regardless of the device size. The sensor point corresponds to the point touched on the touch screen, and to the pointer position when using mouse or touchpad. For keyboard interaction, exploring the whole bi-dimensional area of the function graph with keys would be unpractical. Therefore, we limit the exploration to the x coordinates only using left-right keys. In this case the y_s coordinate of the sensor point s is set to the $f(x_s)$, that is the value of the function at the explored coordinate x_s. This means that the sensor point s is effectively snapped onto the function graph.

The system computes the projection of the point s on the graph of the function $f(x)$ as a point p having the same x_s coordinate as s and y set to the value of $f(x_s)$ corresponding to that x_s coordinate. Therefore $p = (x_s, f(x_s))$ (see Figure 1). This means that, in case of keyboard exploration, $s = p$. The coordinates of the projection point p, as well as the distance between s and its projection p on the function graph are sonified according to the criteria described in the next section as a way to describe the structure of the displayed function graph to a blind person.

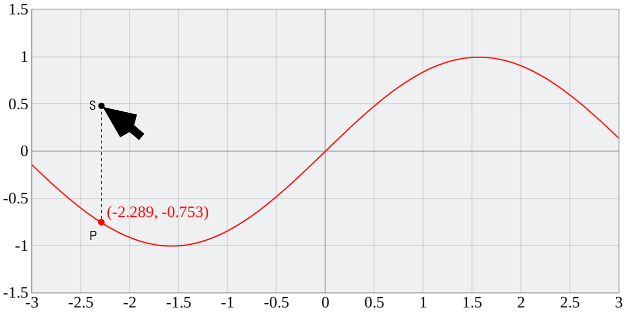


Figure 1: *AudioFunctions.web* showing a $\sin x$ function.

3.2 Sonification Principles

Differently from prior work [33], which presented 2 sonification paradigms accessed from different areas of the screen, *AudioFunctions.web* proposes a unique sonification approach on the whole exploration area which better adapts to the small form factor of smartphone mobile devices and resizeable browser windows on PCs. In the proposed approach, when the user moves the sensor point s *AudioFunctions.web* modulates a sound to convey the coordinates of the corresponding projection point p and the distance between s and p using respectively the sound frequency and intensity.

The frequency ν of the modulated sound is designed to convey the function value $f(x_s)$: for higher values of $f(x_s)$, a sound with higher frequency is generated. The frequency varies between a minimum ν_{min} and maximum ν_{max} and is proportional to value of $f(x_s))$ in the range $[y_{min}, y_{max}]$. Formally, the frequency ν is defined as

$$\nu = \nu_{min} + y_n \cdot (\nu_{max} - \nu_{min})$$

where $y_n = (f(x_s) - y_{min})/(y_{max} - y_{min})$ is the value of $f(x_s)$ normalised with respect to the range $[y_{min}, y_{max}]$. As in [33], we use $\nu_{min} = 200hz$ and $\nu_{max} = 2000hz$, to provide an ample frequency range for sonification.

The intensity of the produced sound is designed to be stronger the closer the sensor point s is to the projection point p, and therefore to the graph of the function $f(x)$. The intensity I is set to a minimum value of $I_{min} = -30dB$ when the euclidean distance $d = |y_s - f(x_s)|$ of the sensor point from the projection point is $d > d_{max}$. The parameter $d_{max} = 0.1 * (y_{max} - y_{min})$ in order to make the intensity tracking feasible also on small form factor devices.

When $d < d_{max}$, the sound intensity is defined as

$$I = I_{min} + (1 - d_n) \cdot (I_{max} - I_{min})$$

where $d_n = d/d_{max}$. That is, the distance d is normalised with respect to the range d_{max}. The maximum intensity range is set to $I_{max} = -10dB$, but can be modified through standard volume controls of the device according to user preferences. By following the maximum intensity of the sound, the user can track the function shape using touchscreen, touchpad or mouse. In the case of keyboard interaction, we recall that $s = p$, and therefore the intensity is always maximum: $I = I_{max}$.

While proprioception provides the information on the explored position on the device screen, the sound is also spatialised through stereo channels in order to reinforce the perception of the explored position with respect to the viewport. Specifically, when the x_s coordinate of the sensor point s is to the left or right of the viewport, the sound is designed to be played more intensely on the left or right speaker respectively. The sound intensities of the left and the right audio channel, normalised with respect to the range (x_{min}, x_{max}), are defined as $I_r = I \cdot (x_s - x_{min})/(x_{max} - x_{min})$ and $I_l = I - I_r$ respectively. Thus, at $x_s = x_{min}$, the sound is reproduced only on the left side, while at $x_s = x_{max}$ it is completely on the right.

3.3 Earcons Design

To improve the understanding of the function, the user is notified when the sensor point reaches features of interest of the function, such as local minima / maxima, intersections with the axes or the passage through the origin of the graph. This information is designed to reinforce the understanding of the shape of the graph, and also as a cue to request additional information at those points. For this purpose we use earcons [7], short auditory cues designed to raise the user's attention and be easy to recognise.

AudioFunctions.web can provide verbal or musical earcons when a point of interest is explored. Verbal earcons are intuitive and therefore suitable for novice usage while musical earcons are designed to be less intrusive in presence of other verbal messages and more appropriate for prolonged use by experienced users. Since our experiments were not designed to assess prolonged system usage, we used only verbal earcons. Musical earcons will be evaluated as future work.

Verbal earcons are designed to read the label of points of interest when the sensor point s is on them. The possible earcons read "min" and "max" when the sensor point is on a local minimum or maximum respectively. "x" and "y" are read when crossing x and y axes respectively, and "origin" is read when passing through the origin of the plane.

Conversely, musical earcons are designed as different musical instrument notes. They are more concise and do not overlap with other verbal information, but they not are as immediate as verbal earcons, which makes them more suitable for prolonged usage after some learning. For intersections with axes we use acoustic guitar notes. On intersection with x and y axes, $A2$ and $D4$ notes are played, while passing through origin triggers $B3$ note. When minima and maxima are explored, respectively earcons using $C5$ and $C6$ piano notes are played. All notes used are at least a musical octave distant one from another in order to be easily recognizable.

---

[5] Henceforth, coordinates refer to the cartesian coordinate system.

Table 1: User actions available in *AudioFunctions.web* and corresponding interactions on different interfaces.

Action	Touchscreen	Touchpad/mouse	Keyboard
Exploration	At touch coordinates	At mouse pointer coordinates	Left/right **arrow** keys
Fast Exploration	–	–	Up/down **arrow** keys
Return to Center	–	–	**Escape** key
Request Information	Touch with second finger	Left click	**Space bar** key
Cancel Information	Release second finger	Right click	**C** key
Complete Sonification	Double tap with one finger	Double left click	**M** key
Toggle Edit Mode	Double tap with second finger	Double right click	**E** key

3.4 Additional Information Requests

The sonification conveys the overall structure of the graph, which is useful to form a high level mental model of the function. However, this is not sufficient for an analytical understanding of its values. Also, some characteristics of a function, such as its trend or its concavity, which can be grasped with a glance, are not as easily conveyed through sound only. Therefore, we design verbal messages which provide additional information on the function on demand.

While exploring, the user can request additional information on the function at the explored point. Such information include the function coordinates $(x_s, f(x_s))$ corresponding to the sensor point s. Additionally, to convey more involved characteristics of the function, we provide the first and the second derivatives of the function ($f'(x)$ and $f''(x)$ respectively), and their values.

3.5 Interaction through Different Interfaces

We designed an integrated interaction paradigm, accessible from different interfaces and capable of providing consistent exploration experience on diverse devices, including small form factors. We defined a set of actions needed to use the system, and corresponding interactions on different interfaces: touchscreen, touchpad, mouse and keyboard (see Table 1). These interfaces are also designed to be used concurrently; for example, touchscreen **Exploration** interaction can be coupled with keyboard **Request Information** interaction for quicker access to additional information at the explored point. The following user actions available in *AudioFunctions.web*:

Exploration.

The exploration action involves moving the sensor point s within the viewport. While the user explores, the coordinates of the function are sonified as described previously. On a touch screen, touchpad or mouse, the sensor point moves with touch or pointer respectively. Instead, when using the keyboard, the sensor point is initially set to last pointer coordinates. Then, using left and right keys, the sensor point moves by a fixed value j to the left or to the right respectively. The value of j is defined through a parameter in proportion of the viewport range. In our usage, j was set to $(x_{max} - x_{min})/50$.

Fast Exploration.

On keyboard the exploration tends to be precise but also slow. Thus, we defined a fast exploration mode: using up and down keys will move the sensor point by $5 \cdot j$ to the right and left respectively.

Return to Center.

Differently from other modalities, the keyboard interaction lacks a consistent reference frame. Thus, the users may loose their perception of what part of the function they are exploring. To address this issue, *AudioFunctions.web* presents a “Return to Center” action.

Request Information.

The user can analytically explore the function by requesting its value at the sensor point s, as well as first and second derivatives of the function and their values.

Cancel Information.

The user may be interested only in some of the provided additional information, or may need to cancel the request for additional information to proceed in the exploration. Therefore, an action to stop the reading of additional information prematurely is also available on every interface.

Complete Sonification.

Another functionality, present also in other graphing assistive technologies [13, 36, 33] reproduces the displayed portion of the function as one sequence of sounds. The x range of the viewport is subdivided in a number r of equally-sized frames. For each frame, the sensor point is placed at its coordinates and the corresponding function value is sonified as previously described for a duration t.

Toggle Edit Mode.

The user may also be interested in exploring other parts of the graph. Thus, we provide an option to switch to and from edit mode, in which moving the sensor point instead moves the viewport by the same quantity and the new viewport range is read verbally.

3.6 Access from Digital Documents

In order to access *AudioFunctions.web* directly from the teaching material, which can be provided in a digital document format such as PDF, ebook or as a web page, we designed the system to be accessed directly from a hypertext link. Thus, all the parameters used can be passed as GET variables in the link URL.

Specifically the URL format exposes the following parameters:

f - this parameter accepts a string representation in *interval-arithmetic* javascript library notation format[6] of the function to render (*default*: $\sin(x)$)

center - this parameter specifies the coordinates of the viewport center as an array (*default*: $[0,0]$)

scale - since the system can be used on devices with unknown form factors, there is no absolute mapping of the graph values to the screen size. This parameter specifies the scale of the viewport, defined as the number of cartesian coordinate units displayed in the horizontal range of the browser window. For example, setting center to $[0,0]$ and scale to 10 renders coordinates between $x = -5$ and $x = 5$. The y range is computed to be proportional to the defined x range. (*default*: 10)

m_earcons: this boolean value enables musical earcons if set to 1. Otherwise, verbal earcons are used. (*default*: 0)

The following example of an *AudioFunctions.web* link will show an $\arctan(x)$ function in the range $[-5, 15]$, with musical earcons:

`https://ewserver.di.unimi.it/audiofunctions/view?f=atan(x)&center=[5,0]&scale=20&m_earcons=1`

---

[6] `https://mauriciopoppe.github.io/interval-arithmetic/`

Table 2: Participants' demographic information.

PID	Age	Impairment		Self-Assessed Expertise Level with							Years of Usage		Functions Studied at
		Type	Onset	PC	Keyboard	Touchpad	Mouse	Mobile	Touchscreen	Graphs	PC	Mobile	
P1	44	Blind	Birth	5	6	6	2	6	7	2	31	11	High school
P2	50	Blind	20	7	7	1	1	5	6	6	25	4	University
P3	25	Blind	Birth	7	6	4	4	6	6	6	12	3	University
P4	31	Blind	13	6	6	5	6	4	4	6	22	1	University
P5	38	Blind	29	7	7	7	7	7	7	4	30	9	High school
P6	40	Blind	Birth	7	7	5	4	7	7	4	25	7	High school
P7	21	Blind	Birth	6	7	3	3	6	6	5	10	4	High school
P8	24	Light	5	6	7	5	5	6	7	4	10	4	High school
P9	43	Blind	15	6	7	6	6	6	7	3	20	3	High school
P10	29	Shapes	15	5	7	6	6	5	6	2	10	5	High school
P11	35	Blind	3	7	7	6	4	7	7	6	20	6	University
P12	37	Blind	7	6	7	5	5	6	7	3	21	6	High school
P13	41	$< 2/10$	31	5	6	2	2	5	6	7	22	5	University

3.7 System Implementation

AudioFunctions.web is implemented using JavaScript, on top of novel web technologies and standards. It is therefore available on all modern browsers, on both mobile devices and PCs. Additionally, since *AudioFunctions.web* and its dependencies are all standalone JavaScript code, it is also possible to create a single self-contained html file which includes all the required code and can be embedded within a digital document such as PDF to be displayed offline[7].

3.7.1 Function Graph Drawing and Exploration

AudioFunctions.web draws graphs of mathematical functions using *Function-Plot*[8], a javascript library which employs interval arithmetic [20] in order to create pixel-perfect visualization of function graphs that compensate for rounding errors. *Function-Plot* is built on top of *D3.js*[9] javascript library, which uses SVG, HTML5 and CSS web standards[10] to create, manipulate, style and interact with dynamic data visualizations.

3.7.2 Sonification

To generate responsive and accurate spatialized sonification effects, *AudioFunctions.web* employs *Web Audio API*[11], a **W3C Candidate Recommendation** since 18 September 2018. Recently, the feasibility of accessible sound-based representations of visual information using *Web Audio API* has been investigated [37, 34]. At the time of writing, *Web Audio API* is implemented by all recent PC and mobile browsers.

In *AudioFunctions.web*, the *Web Audio API* is accessed using *Tone.js* javascript library[12]. *Tone.js* is a framework for creating interactive music and sound effects, with advanced scheduling capabilities and musical abstractions. *Tone.js Oscillator()* class is used for modelling the frequency and the intensity of the generated sound, which we recall are used in order to convey the function value $f(x)$ and the distance of the sensor point s from the function respectively. Instead, *Tone.js Panner()* class is used for generating stereophonic spatialized sound to convey the x coordinate of the sensor point s.

3.7.3 Voice Generation

For speech generation, *AudioFunctions.web* uses *Web Speech API*[13], which defines speech synthesis and speech recognition capabilities. As of 1 October 2018, *Web Speech API* is a **W3C Community Draft**. However, the support for its *SpeechSynthesis* specification is already included in major PC and mobile browsers (Chrome, Edge, Firefox and Safari).

4. USER STUDY

To evaluate the proposed interaction modalities and assess how they are perceived by the users during the interaction with *AudioFunctions.web*, we conducted user studies with 13 blind and visually impaired participants. The participants were asked to explore 3 different mathematical functions with all 3 interaction modalities, focusing on the usability of the system and the feasibility to explore the function graph with the proposed modalities. A final questionnaire assessed the participants' preferred interaction modality, perceived pros and cons for each modality and collected suggestions on improving the system.

4.1 Participants

The study was conducted with 12 blind participants. While the system was not designed specifically for users with low vision, we also included one such participant who has a high degree of expertise in mathematics, in order to assess the feasibility of the usage of the system also with this user group.

Table 2 lists participants' demographic information and self-assessed expertise, on a scale from 1 to 7 with different platforms (PC and mobile devices), interfaces (keyboard, touchpad, mouse and touchscreen) and function analysis. Participants $P1-P12$ were totally blind or had residual vision unusable for visual function graph exploration. Specifically $P10$ perceived only the presence of large shapes and $P8$ could only detect the presence of light. These participants are referred as *blind* in the paper.

$P13$ had a residual visual acuity $< 2/10$. While this level of vision was not sufficient to see the graph, it could be used to track the movement of the finger or the pointer on the screen. Due to diverse sight conditions, this participant is not grouped with others during data analysis. Instead, the results for this participant are reported separately.

---

[7] Embedded *AudioFunctions.web* (requires compatible reader): WebAudioFunctions.html

[8] `https://mauriciopoppe.github.io/function-plot/`

[9] `https://d3js.org/`

[10] `https://www.w3.org/standards/webdesign/`

[11] `https://www.w3.org/TR/webaudio/`

[12] `https://tonejs.github.io/`

[13] `https://w3c.github.io/speech-api/`

Q1	I think that I would like to use this system frequently
Q2	I found the system unnecessarily complex
Q3	I thought the system was easy to use
Q4	I think I would need support of a technical person to use the system
Q5	I found the various functions in this system were well integrated
Q6	I thought there was too much inconsistency in this system
Q7	I imagine that most people would learn to use this system very quickly
Q8	I found the system very cumbersome to use
Q9	I felt very confident using the system
Q10	I needed to learn a lot of things before I could get going with the system

(a) System Usability Scale Questions

S1	Exploring the graph with this interface was intuitive
S2	Interacting with this interface was intuitive
S3	Supporting cues at points of interest were useful
S4	Request information functionality was useful
S5	Complete sonification functionality was useful
S6a	Tracking the graph shape by volume was intuitive
S6b	Return to center functionality was useful

(b) Additional Questions

Figure 2: Questionnaires compiled after each test

Participants' age ranged between 21 and 50 (34.75 ± 8.92[14]). Among blind participants, 4 had visual impairment at birth, and 3 had visual impairment onset under 8 years of age. Prior literature categorizes these participants as "early-onset" blind [35], while others, with visual impairment onset over 13 years of age are labeled as "late-onset" blind. On average, participants have been using a PC for 19.67 ± 7.62 years, and a mobile device (smartphone or tablet) for 5.08 ± 2.43 years.

Participants felt confident in using both PC (6.25 ± 0.75) and mobile devices (5.92 ± 0.90). They also felt confident with keyboard (6.75 ± 0.45) and touch screen (6.41 ± 0.90) interaction. Conversely, touchpad and mouse interfaces had lower scores (4.92 ± 1.62 and 4.41 ± 1.78 respectively) since these interfaces are rarely used by blind people. All participants have studied mathematical functions at least at high school level and 4 have further studied them at university level. For those participants, their self-assessed expertise level with function graphs was consistently higher (6.0 ± 0.0) compared to others (3.37 ± 1.06).

4.2 Apparatus

The user studies were conducted remotely, with participants accessing the system through their own devices and an experimenter providing instructions telephonically. Since iOs devices disable web audio during phone calls, for most participants another device needed to be used for telephonic instructions. During tests, headphones were used to convey spatialized stereophonic sonification feedback to the participants. The training was performed without headphones in order for the participants to receive instructions from experimenters.

7 participants used a Windows PC during the experiments while others used a Mac. On PC, the participants used either Chrome or Firefox browsers. All participants used iPhone mobile devices with Safari browser, besides $P3$ who used Chrome browser on an Android smartphones. Since these configurations produced functionally identical results no further analysis considered them as variables. $P2$ and $P13$ had a desktop PC and therefore used a mouse, while others used a touchpad on a notebook PC.

Since many of the technologies used by *AudioFunctions.web* require recent browser versions, the participants were asked to update their devices before the study. One candidate participant had an older browser version which did not support the required technologies. Another candidate participant had technical issues with his PC and could not complete the study. These participants were therefore excluded and are not considered in this study nor analysis.

4.3 Procedure

The study protocol[15] initiated with an introductory briefing to explain the motivation and the scope of the study to the participants. Afterwards, the participants' demographic data was collected and expertise self-assessment questionnaires were administered. Then, we proceeded with the training step, experimental tasks and final questionnaires for each interaction modality. In total, the experiment lasted about 1 hour and the collected data was transcribed anonymously.

During the training step, the participants were first explained how the system works and how to perform the available actions in the considered interaction modality. Then, they were presented the graph of a linear function ($f(x) = x$)[16] and they were asked to explore it for a couple of minutes. The training step for each interaction modality was about 5 minutes long.

For each experiment the participants were given one function to explore, focusing on the usability of the system with different interaction modalities. Specifically, the following functions were assigned randomly to different interaction modalities and were presented to the participants to explore for a couple of minutes: a vertical cusp ($f(x) = log(x^2)$)[17], a sine function ($f(x) = sin(x)$)[18], and a bell curve ($f(x) = 2e^{-x^2/2}$)[19]. The interaction modalities were ordered in a counter-balanced way during tests to offset possible learning effects.

After each experiment, the participants were asked to respond to a questionnaire which included SUS questions (see Figure 2a) and additional 6 questions specific to the interaction modality used (see Figure 2b). Additionally, the participants were asked to provide a brief description of the explored function. This served the purpose of understanding what functionalities the participants used and how they explored the function graph.

Following the experiments, we assessed the users' appreciation of the proposed interaction modalities through a 5-point Likert-scale questionnaire, which included the System Usability Scale [8] (SUS) questions, and additional questions specific for the proposed interaction modalities. The questionnaires were presented in English or in Italian according to participants' preferred language.

---

[14] As a convention *Mean* ± *Standard Deviation* will be used

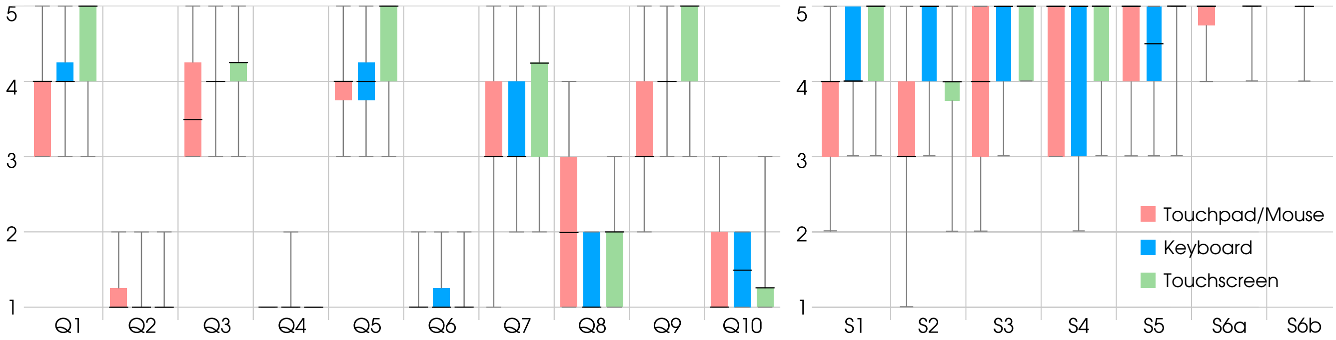
[15] Transcript available at: `https://ewserver.di.unimi.it/audiofunctions/viewpr.html`

[16] `https://ewserver.di.unimi.it/audiofunctions/view?f=x`

[17] `https://ewserver.di.unimi.it/audiofunctions/view?f=log(x^2)&scale=40`

[18] `https://ewserver.di.unimi.it/audiofunctions/view?f=sin(x)`

[19] `https://ewserver.di.unimi.it/audiofunctions/view?f=2*exp(-x^2/2)&scale=10`



(a) System Usability Scale scores (b) Additional Questions scores

Figure 3: Boxplots of System Usability Scale and additional questions. (Whiskers: min/max, Box: Q1/Q3, Line: Median)

5. RESULTS

All participants managed to proficiently use the system and explore the presented function graphs successfully with all three interaction modalities. However, the characteristics of the descriptions of the function graphs, the quality of the descriptions and the vocabulary used varied based on participants' knowledge of the domain. In particular, participants who reported a lower knowledge of function graphs ($P9$, $P10$. $P12$) frequently tried to describe the overall shape of the graphs.

> $P1$ (cusp on touchpad): *"It was shaped like a V."*

Other less experienced partipants described functions referring to their general increasing and decreasing trends.

> $P9$ (bell curve on mobile): *"It goes up, then down."*

Some of them also reported the relative position of the function with respect to the axes.

> $P12$ (sine on keyboard): *"It goes under the x axis, then over it, then under."*

Sometimes such information was provided referring to key points of interest.

> $P6$ (sine on keyboard): *"It decreases, reaches a minimum under the x axis to the left of the y axis, it increases and crosses the y axis, it reaches a maximum, and then goes down crossing the x axis to reach a minimum."*

Conversely, more expert participants ($P2$, $P3$, $P4$, $P11$) frequently reported quantitative values, in particular associated to the points of interest of the function graph such as local minima or intersection with axes.

> $P11$ (bell curve on touchpad): *"It grows from 0 at $-\infty$, it intersects the y axis at about 1.9, and then it decreases asymptotically to the x axis."*

Participants with the highest expertise level with function graphs and analysis (including $P13$) would often identify the exact function represented in the examined graph.

> $P3$ (bell curve on keyboard): *"It is a gaussian curve."*
>
> $P13$ (cusp on mobile): *"It is a vertical cusp asymptotic to the y axis."*

5.1 System Usability Scale

SUS scores (see Figure 3a) show that all three interaction modalities were evaluated positively by the participants. In particular, touchscreen interaction registered the highest score (86.7 ± 9.90), ranking A^+ according to SUS metrics [30]. Keyboard interaction ranked A (82.7 ± 8.36) and touchpad/mouse ranked B^+ (77.7 ± 10.5). Pairwise T-tests show that the average SUS score for touchpad/mouse interaction was significantly lower than both touchscreen ($t_{(11)} = -2.21$, $p < 0.05$) and keyboard ($t_{(11)} = -4.23$ $p < 0.01$).

Considering the specific questions, we can see that the participants deemed touchpad/mouse interaction to be more cumbersome than keyboard (2 ± 1.04 vs 1.33 ± 0.49, $t_{(11)} = 2.35$, $p < 0.05$). In particular, they felt that they would use touchpad/mouse less often than both keyboard (3.75 ± 0.62 vs 4.08 ± 0.67, $t_{(11)} = 2.35$, $p < 0.05$) and touchscreen (4.25 ± 0.75, $t_{(11)} = 2.57$, $p < 0.05$). This consideration was also reported by $P13$, who replied to $Q1$ with a score of 5 for keyboard and touchscreen, and 2 for touchpad/mouse.

Touchscreen interaction was also considered faster to learn than touchpad/mouse (3.83 ± 0.94 vs 3.17 ± 1.03, $t_{(11)} = 3.55$, $p < 0.005$), and the participants felt more confident with touchscreen than with touchpad/mouse interaction (4.25 ± 0.75 vs 3.5 ± 0.80, $t_{(11)} = 3.46$, $p < 0.01$). Furthermore, the scores highlight that the various functions of the touchscreen modality were perceived by the participants' to be better integrated than the touchpad/mouse ones (4.25 ± 0.75 vs 3.83 ± 0.58, $t_{(11)} = 2.80$, $p < 0.05$).

We have also evaluated the impact of participants' characteristics on the assigned SUS scores. An indicator which was found to significantly impact the appreciation of the system was the level of education at which the participants studied mathematical functions, and the corresponding self-assessed expertise with function graphs. The participants with university level education in mathematics assigned generally higher scores than others, in particular for touchpad/mouse (86.88 ± 7.74 vs 73.13 ± 8.63, $t_{(10)} = 2.68$, $p < 0.05$). This holds true also for $P13$, who assigned a score of 97.5 to touchscreen and keyboard and 90 to touchpad/mouse.

Regarding the specific questions, participants with university level maths education found the touchpad/mouse interaction to be easier to use (4.5 ± 0.58 vs 3.38 ± 0.74, $t_{(10)} = 2.63$, $p < 0.05$) compared to other participants. The effort required to learn to use the system with touchpad/mouse was also perceived to be lower for participants with university level maths education (1 ± 0 vs 1.85 ± 0.64, $t_{(10)} = 2.66$, $p < 0.05$), and they similarly assumed it would be easier to learn to use the system for other users as well (4 ± 0.82 vs 2.75 ± 0.89, $t_{(10)} = 2.35$, $p < 0.05$), compared to participants with high school maths education.

5.2 Specific Interaction Questions

All participants evaluated the specific functionalities of different interaction modalities positively (see Figure 3b). Curve tracking was found to be less intuitive than exploration for both touchpad/mouse (3.17 ± 1.03 vs 3.67 ± 0.89, $t_{(11)} = 2.57$, $p < 0.05$) and touchscreen (3.67 ± 0.89 vs 4.5 ± 0.67, $t_{(11)} = 4.02$, $p < 0.05$) interaction modalities. However, on touchscreen, both exploration ($t_{(11)} = 3.46$, $p < 0.01$) and curve tracking functionalities ($t_{(11)} = 2.57$, $p < 0.05$) were found to be more intuitive than on touchpad/mouse.

The usefulness of additional information requests was generally perceived lower for keyboard interaction (4.42 ± 0.69) than for other modalities (4.58 ± 0.67 for touchpad/mouse and 4.83 ± 0.58 for touchscreen), perhaps due to the highly predictable nature of this interaction modality. In particular, it was significantly lower with respect to touchscreen interaction ($t_{(11)} = 2.80$, $p < 0.05$).

5.3 User Preferences

Touchscreen was preferred by 6 blind participants ($P1$, $P6$, $P7$, $P8$, $P11$, $P12$) as well as $P13$. Among the others, 5 preferred keyboard ($P2$, $P4$, $P5$, $P9$, $P10$), while one participant ($P3$) equally appreciated touchpad/mouse and touchscreen interaction. Participants who preferred touchscreen interaction considered it more intuitive, responsive, and immediate to learn.

> $P6$: *"I find it intuitive to explore with a finger. I can move left and right, or jump around as needed"*

They also compared this type of interaction to the exploration of a graph on embossed paper.

> $P1$: *"The whole surface can be used. It's like feeling the graph on paper"*

Participants also enjoyed the capability to explore a function on their mobile device while using their PC for taking notes or studying.

> $P7$: *"I can easily follow the graph and input commands. I can also take notes on my computer if I'm studying."*

However some participants also found it difficult to find specific coordinates during touchscreen interaction ($P1$, $P6$, $P9$, $P11$, $P12$) or track the graph ($P10$) since mobile device screens are small ($P3$, $P4$). $P2$ also reported that he would get confused between interaction gestures and standard Voiceover gestures.

Participants who preferred keyboard interface appreciated that the interaction was simple and precise.

> $P9$: *"I have full control over the movements, I don't need to roam around the page to understand the graph."*

Furthermore, as the keyboard is the default interface for blind users accessing a PC, they also felt accustomed to this type of interaction.

> $P5$: *"I have been using mostly keyboard so I am confident with it. It is simple and since I will have course books on my PC it is useful to have this tool on it too."*

On the other hand, keyboard interaction was found to be slower ($P1$, $P6$) since the exploration is sequential. It also provides less awareness of the general shape of the graph since the exploration is constrained to one dimension ($P1$, $P2$, $P4$). $P8$ also reported that it was difficult to remember all keys used for the interaction.

Touchpad/mouse interaction combines the proprioceptive qualities of the touchscreen and the ability to use *AudioFunctions.web* on PC, which was appreciated by $P1$, $P3$ and $P13$.

> $P3$: *"The exploration is similar enough to the touchscreen, and I could do it from my PC."*

However, the participants sometimes reported to get lost ($P1$, $P2$, $P5$). Also, they were not used at all to this interface and therefore could not easily adapt to its sensitivity and precision ($P6$, $P7$, $P8$). However participants also reported that with practice it could be enough for knowledgeable users.

> $P13$: *"It is also quite intuitive, for users with good mathematical knowledge it should be enough."*

6. DISCUSSION

AudioFunctions.web was found to be usable with all 3 proposed interaction modalities, and all participants managed to explore and describe function graphs with all 3 interfaces. However, user capabilities and personal preferences play a major role in the choice of the preferred modality and the overall acceptance of the system.

6.1 Impact of User Knowledge

The key implication of our findings is that user knowledge of the problem domain impacts the usability of the system more than the expertise with the interface. Indeed, higher expertise with the platform or the interface used did not impact the exploration usability. Instead, the reported SUS scores were influenced by the participants' knowledge of mathematical functions. Participants with university level education in maths also had higher appreciation and confidence in the capabilities of the system. They perceived less effort in learning to use the system, and believed that the system would be easy for others to use as it was for them.

This could be motivated by the fact that users with high knowledge of the problem domain know what to expect as the result of the interaction and therefore can assess if the interaction is proceeding correctly. Instead, even with high expertise with the interface, users unfamiliar with the problem domain will be uncertain during the interaction, will require more verbose feedback and will need some form of validation that the interaction is proceeding correctly. Clearly, this influences the design requirements of interaction paradigms for systems that present highly specialized knowledge. In such systems, it is crucial to focus on the need to personalize the interaction in order to support users with diverse expertise levels. Indeed, attention should be drawn to the design of systems such as *AudioFunctions.web* to accommodate users that are still developing their knowledge of the problem domain (e.g., new students in the field of mathematical function analysis).

6.2 Interaction Modalities

Our study also exposed a strong division between the participants who favoured the proprioceptive exploration of the touchscreen modality and those who preferred more constrained keyboard-driven exploration. Touchscreen interaction was considered very easy to learn since it is similar to the physical exploration of a function graph on embossed paper. The presence of a clear physical reference frame also helped to easily track the explored position with respect to the graph, and therefore understand the shape of the graph function.

The keyboard interface, being familiar to most participants, was considered simple to operate and enabled investigating the function without exploring the area of the graph. Indeed, the participants felt there were less possibilities for errors or getting lost as this interaction modality is one-dimensional and they could, at any time, return to the origin with a single key.

Instead, touchpad/mouse interaction was less appreciated since it lacks the physical reference frame of the touchscreen, or the constrained exploration with "Return to Center" functionality of the keyboard. Participants were also less familiar with these interfaces, and they sometimes lost their orientation within the graph area, which made the exploration more difficult.

6.3 Comparison with prior work

While our prior work [33] evaluated the feasibility of sonification-driven proprioceptive function graph exploration, in this paper we focus on multiplatform, interface independent exploration. The modifications to the design of the exploration technique are therefore motivated by the need to adapt to diverse target interfaces and not as a direct improvement over the interaction paradigm. Thus, a direct performance comparison with the prior solution is outside the scope of the work. However, during the experiments we discovered two key differences in the interaction technique with our previous work, which impacted the experimental results, which we describe here.

6.3.1 Dual Sonification

The introduction of frequency and volume dual sonification in *AudioFunctions.web*, in place of the mono-dimensional and bi-dimensional exploration, present in *AudioFunctions*, is favorably perceived by the participants. Indeed, such interaction enables the exploration of the function graph without dividing the viewport in separated areas. This is confirmed by the positive replies to questions $S1$ and $S6a$ which investigate exploration and tracking of the function using sound frequency and intensity respectively.

6.3.2 Verbal Earcons

The addition of diversified verbal earcons in *AudioFunctions.web*, instead of a simple notification sound on points of interest in *AudioFunctions* is also evaluated positively with all 3 exploration modalities (Question $S3$). The importance of these cues is also reflected in the fact that 9 participants reference key points in their description of the functions, and 3 report actual function equations (which also confirms a perfect understanding of referenced key points).

7. CONCLUSIONS AND FUTURE WORK

In this paper we propose *AudioFunctions.web*, a web app that supports blind people during the exploration of mathematical function graphs. *AudioFunctions.web* is publicly available for mobile devices as well as on PCs, and can be accessed using touchscreen, keyboard, touchpad or mouse. We evaluated *AudioFunctions.web* with 12 blind and 1 low-vision participants, focusing on the usability of the system when accessed through different interaction paradigms.

Results show that all proposed interaction modalities are highly usable, but touchpad interaction is more difficult due to the absence of a consistent reference frame. In particular, mathematical knowledge plays a crucial role in participants' evaluation scores and capability to interact with *AudioFunctions.web*.

As future work we will investigate how to better personalize the interaction with our system considering user expertise level, abilities and preferences. For example, we will allow to customize which additional information should be read and which earcons should be played. We will also create personalized system tutorials specific for the user's knowledge (*e.g.*, high school vs. university level).

Furthermore, we intend to design new interaction modalities to personalize the system for people with different visual impairments, such as reduced visual acuity or limited field of view. We will also investigate multi-modal, concurrent interaction combining different interfaces such as tactile graphs coupled with auditory feedback. Similarly, we will address touchpad interaction limitations, for example by constraining the exploration to the graph viewport only, and by providing a "Return to Center" functionality similar to keyboard interaction. This will be achieved using the novel *Pointer Lock 2.0 API*[20]. Finally, we will augment the system with the capability to sonify arbitrary graphs and shapes other than functions.

8. REFERENCES

[1] Dragan Ahmetovic, Valeria Alampi, Cristian Bernareggi, Andrea Gerino, and Sergio Mascetti. 2017. Math Melodies: Supporting Visually Impaired Primary School Students in Learning Math. In *Proceedings of the 14th Web for All Conference on The Future of Accessible Work*. ACM, 26.

[2] Dragan Ahmetovic, Tiziana Armano, Cristian Bernareggi, Michele Berra, Anna Capietto, Sandro Coriasco, Nadir Murru, Alice Ruighi, and Eugenia Taranto. 2018. Axessibility: a LaTeX Package for Mathematical Formulae Accessibility in PDF Documents. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 352–354.

[3] Nancy Alajarmeh. 2012. Doing Math: Mathematics Accessibility Issues. In *Proceedings of the International Cross-Disciplinary Conference on Web Accessibility (W4A '12)*. ACM, New York, NY, USA, Article 23, 2 pages.

[4] Tiziana Armano, Anna Capietto, Sandro Coriasco, Nadir Murru, Alice Ruighi, and Eugenia Taranto. 2018. An Automatized Method Based on LaTeX for the Realization of Accessible PDF Documents Containing Formulae. In *Computers Helping People with Special Needs - 16th International Conference, ICCHP 2018, Linz, Austria, July 11-13, 2018, Proceedings, Part I*. 583–589.

[5] Sandra Bardot, Marcos Serrano, Bernard Oriola, and Christophe Jouffrais. 2017. Identifying How Visually Impaired People Explore Raised-line Diagrams to Improve the Design of Touch Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 550–555.

[6] Cristian Bernareggi, Christian Comaschi, Andrea Marcante, Piero Mussio, Loredana Parasiliti Provenza, and Sara Vanzi. 2008. A Multimodal Interactive System to Create and Explore Graph Structures. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems (CHI EA '08)*. ACM, New York, NY, USA, 2697–2702.

[7] Meera M. Blattner, Denise A. Sumikawa, and Robert M. Greenberg. 1989. Earcons and Icons: Their Structure and Common Design Principles. *SIGCHI Bull.* 21, 1 (Aug. 1989), 123–140.

[8] John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[9] Niccolò Cantù, Mattia Ducci, Dragan Ahmetovic, Cristian Bernareggi, and Sergio Mascetti. 2018. MathMelodies 2: a Mobile Assistive Application for People with Visual Impairments Developed with React Native. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 453–455.

[10] Davide Cervone, Peter Krautzberger, and Volker Sorge. 2016. Towards Universal Rendering in MathJax. In *Proceedings of the 13th Web for All Conference (W4A '16)*. ACM, New York, NY, USA, Article 4, 4 pages.

[11] Catherine S Fichten, Jennison V Asuncion, Maria Barile, Vittoria Ferraro, and Joan Wolforth. 2009. Accessibility of e-learning and computer and information technologies for students with visual impairments in postsecondary education. *Journal of Visual Impairment & Blindness* 103, 9 (2009), 543.

[12] John A. Gardner. 2002a. Access by Blind Students and Professionals to Mainstream Math and Science. In *Proc. of the 8th Int. Conf. on Computers Helping People with Special Needs*. Springer-Verlag.

---

[20] `https://www.w3.org/TR/pointerlock-2/`

[13] John A Gardner. 2002b. Access by blind students and professionals to mainstream math and science. In *International Conference on Computers for Handicapped Persons*. Springer, 502–507.

[14] John A. Gardner and Vladimir Bulatov. 2006. Scientific Diagrams Made Easy with IVEOTM. In *Proc. of the 10th Int. Conf. on Computers Helping People with Special Needs*. Springer-Verlag.

[15] Andrea Gerino, Nicolo Alabastro, Cristian Bernareggi, Dragan Ahmetovic, and Sergio Mascetti. 2014. Mathmelodies: inclusive design of a didactic game to practice mathematics. In *International Conference on Computers Helping People With Special Needs*. Springer, Cham, 564–571.

[16] Andrea Gerino, Lorenzo Picinali, Cristian Bernareggi, and Sergio Mascetti. 2015. Eyes-free Exploration of Shapes with Invisible Puzzle. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '15)*. ACM, New York, NY, USA, 425–426.

[17] Cagatay Goncu and Kim Marriott. 2015. GraCALC: An Accessible Graphing Calculator. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '15)*. ACM, New York, NY, USA, 311–312.

[18] T. Götzelmann. 2018. Visually Augmented Audio-Tactile Graphics for Visually Impaired People. *ACM Trans. Access. Comput.* 11, 2, Article 8 (June 2018), 31 pages.

[19] João Guerreiro, Dragan Ahmetovic, Kris M Kitani, and Chieko Asakawa. 2017. Virtual navigation for blind people: Building sequential representations of the real-world. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 280–289.

[20] Timothy Hickey, Qun Ju, and Maarten H Van Emden. 2001. Interval arithmetic: From principles to implementation. *Journal of the ACM (JACM)* 48, 5 (2001), 1038–1068.

[21] Chandrika Jayant. 2006. A survey of math accessibility for blind persons and an investigation on text/math separation. *Seattle: University of Washington* (2006).

[22] Da-jung Kim and Youn-kyung Lim. 2011. Handscope: Enabling Blind People to Experience Statistical Graphics on Websites Through Haptics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2039–2042.

[23] Edward Kim and Kathleen F McCoy. 2018. Multimodal Deep Learning using Images and Text for Information Graphic Classification. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 143–148.

[24] Stephen E. Krufka and Kenneth E. Barner. 2005. Automatic Production of Tactile Graphics from Scalable Vector Graphics. In *Proc. of the 7th Int. Conf. on Computers and Accessibility*. ACM.

[25] Richard E. Ladner, Melody Y. Ivory, Rajesh Rao, Sheryl Burgstahler, Dan Comden, Sangyun Hahn, Matthew Renzelmann, Satria Krisnandi, Mahalakshmi Ramasamy, Beverly Slabosky, Andrew Martin, Amelia Lacenski, Stuart Olsen, and Dmitri Groce. 2005. Automating Tactile Graphics Translation. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '05)*. ACM, New York, NY, USA, 150–157.

[26] Sergio Mascetti, Andrea Gerino, Cristian Bernareggi, and Lorenzo Picinali. 2017. On the Evaluation of Novel Sonification Techniques for Non-Visual Shape Exploration. *ACM Trans. Access. Comput.* 9, 4, Article 13 (April 2017), 28 pages.

[27] Priscilla Moraes, Gabriel Sina, Kathleen McCoy, and Sandra Carberry. 2014. Evaluating the accessibility of line graphs through textual summaries for visually impaired users. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. ACM, 83–90.

[28] Arshad Nasser, Kening Zhu, and P. V.M Rao. 2018. ColorTact: A Finger Wearable Audio-tactile Device Using Customizable Color Tagging. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp '18)*. ACM, New York, NY, USA, 178–181.

[29] Rameshsharma Ramloll, Wai Yu, Stephen Brewster, Beate Riedel, Mike Burton, and Gisela Dimigen. 2000. Constructing Sonified Haptic Line Graphs for the Blind Student: First Steps. In *Proc. of the 4th Int. Conf. on Assistive Technologies*. ACM.

[30] Jeff Sauro. 2011. *A practical guide to the system usability scale: Background, benchmarks & best practices*. Measuring Usability LLC Denver, CO.

[31] Volker Sorge, Charles Chen, T. V. Raman, and David Tseng. 2014. Towards Making Mathematics a First Class Citizen in General Screen Readers. In *Proceedings of the 11th Web for All Conference (W4A '14)*. ACM, New York, NY, USA, Article 40, 10 pages.

[32] Ryo Suzuki, Abigale Stangl, Mark D. Gross, and Tom Yeh. 2017. FluxMarker: Enhancing Tactile Graphics with Dynamic Tactile Markers. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. ACM, New York, NY, USA, 190–199.

[33] Marzia Taibbi, Cristian Bernareggi, Andrea Gerino, Dragan Ahmetovic, and Sergio Mascetti. 2014. Audiofunctions: Eyes-free exploration of mathematical functions on tablets. In *International Conference on Computers Helping People With Special Needs*. Springer, Cham, 537–544.

[34] Brianna J Tomlinson, Prakriti Kaini, Siyan Zhou, Taliesin L Smith, Emily B Moore, and Bruce N Walker. 2018. Design and Evaluation of a Multimodal Science Simulation. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 438–440.

[35] Patrice Voss, Maryse Lassonde, Frederic Gougoux, Madeleine Fortin, Jean-Paul Guillemot, and Franco Lepore. 2004. Early-and late-onset blind individuals show supra-normal auditory abilities in far-space. *Current Biology* 14, 19 (2004), 1734–1738.

[36] Bruce N Walker and Joshua T Cothran. 2003. Sonification Sandbox: A graphical toolkit for auditory graphs. Georgia Institute of Technology.

[37] Torsten Wörtwein, Boris Schauerte, Karin E Müller, and Rainer Stiefelhagen. 2015. Interactive web-based image sonification for the blind. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 375–376.