Accessible PDF: $2 \ngtr 1$

Neil Soiffer
Talking Cat Software.
soiffer@alum.mit.edu

Abstract

Some people feel the accessibility of PDF documents is inferior to that of other formats such as HTML and Word. Theoretically it is possible to make PDF documents very accessible, approaching the accessibility of content in HTML and docx formats. When it comes to mathematics, PDF documents commonly suffer from inferior accessibility due to the complexities of PDF's internal structure: two parallel trees for content and markup rather than a single tree incorporating both. Making math accessible in PDF is technically demanding, as the trees must be tightly interlinked for fully accessible results. An important new feature in PDF 2.0 leads to a technically simple solution for content creators of mathematical expressions in PDF. It is also a solution that is easily handled by assistive technologies.

1 Background

To understand math accessibility in PDF and why it is harder than math accessibility on the web, it is important to understand how accessibility in PDF works. That in turns requires understanding the format of PDF documents. Understanding the reason for why PDF is the way it is requires a little history detour…

The history begins with the invention of laser printers in the late 1970s and early 1980s. A solution that emerged for dealing with the drawing graphics and fonts on devices with unknown resolution was the PostScript language. As the name implies, it was a postfix programming language. A trivial example is shown in Figure 1.

```
/Times-Roman 10 selectfont     Select the font and its size
72 504 moveto                  Move to a position (lower left is origin)
This is PostScript!) show      Draw the text (at current position)
showpage                       Print the page
```

Figure 1: PostScript Example

```
11 0 obj                              Start 11th indirect object
<< /Length 95 >>                      Dictionary index for object
stream                                Begin stream
BT                                    Begin text object
/F8 10 Tf                             Select the font and its size
72 504 Td                             Move to a position
[(This)-333(is)-334(PDF!)]TJ          Draw the text
ET                                    End text object
endstream                             End stream
endobj                                End indirect object
```

Figure 2: PDF Example*

PDF (Portable Document Format) was released in 1993 as a platform-independent format for sharing documents. The equivalent PDF is shown in Figure 2.

The PDF above is just for the one (text) "object" in a bigger PDF document. It is merely 8 out of 165 lines of the uncompressed version of the entire file. Most of the rest of the file is consumed by the embedded font with the remainder devoted to building the structure of the PDF document.

As can be seen, at its core, PDF is similar to PostScript. PDF optimized PostScript for displaying arbitrary pages in large documents by adding document structure to index into the pages and their drawing commands quickly. It removed programming language constructs from PostScript such as loops and conditionals. Most importantly, state was removed so that each page can be drawn in isolation and does not depend upon the state after drawing the previous page. Typically, PDF files include not just commands (including vector graphics), but also the fonts used in the document.

Because PDF was meant to be drawn to a page, accessibility was not a consideration in the original design. Without heuristics such as looking for larger fonts (headings) and numbering/bullets (lists), these low-level drawing commands are completely opaque to screen readers. In fact, there is no requirement that the drawing commands be in reading order. An attempt to read the text as it occurs in the flow of the PDF document can be (and often is) nonsensical without other information.

As the Web became a popular way to read PDF documents, PDF accessibility became important. Also important was relaxing the "looks the same everywhere" rule because on smaller screen, it was often useful to allow the PDF document to "reflow" to match the current window width. In 2000, PDF 1.3 was published. PDF 1.3 added a way to represent the logical structure of a document. A part of logical structure is the reading order of the document. That's crucial to accessibility, reflow, and export to other formats (including copy and paste).

Prior to 1.3, PDF essentially consisted of a relatively flat tree whose main nodes were the pages to be drawn. As shown in Figure 2, the text was embedded inside of a text object, one of several types of objects in PDF. 1.3 adds a logical "structure tree". The leaves of the structure tree point into text streams which contain markers such as BMC/BDC and EMC (begin/end marked/demarcated content). Marked content is nested corresponding to the nesting in the structure tree. The begin commands include a unique number within the page that is referenced by the structure element that points to it. Marked content must be in reading order. PDF's structure tree is similar to HTML's tags. Unlike HTML where the textual content is a direct child of some elements, in PDF, the leaf structure elements point off to the drawing commands in a separate page tree. Many of the standard structure "types" in PDF have corresponding HTML tags such as paragraph, headers, lists, list items, emphasis, etc. PDF that contains a structure tree designed to make the document accessible is often referred to as "tagged PDF".

* Many applications will compress various parts of a PDF document they generate including streams. This example shows the uncompressed stream. A brief summary of the syntax and some PDF commands presents in the examples is given in the Appendix.

Figure 3 shows the structure tree and part of the object stream for a simple document consisting of two headings (“/H1”) and one subheading (“/H2”), where there is a paragraph (“/P”) of text below each heading.

```
5 0 obj
<< /Type /ObjStm /Length 987 /N 12 /First 77 >>
stream
6 0 7 96 8 364 9 389 10 406 11 501 12 562 13 608 14 668 15 729 16 789 17 850
<< /K [ 10 ] … /Type /StructTreeRoot >>
<< /K [ 10 0 R ] /ParentTree 8 0 R … >>
<<  /Annotation /Sect /Artifact /Sect … >>   % role mapping
<< /Nums [ 0 12 0 R ] >>
<< /Names [ ] >>
<< /K [ 11 0 R 13 0 R 14 0 R 15 0 R 16 0 R 17 0 R 18 0 R 19 0 R 23 0 R ]
   /P 6 0 R /S /Document /Type /StructElem >>   % object #10
<< /K [ 0 ] /P 10 0 R /Pg 28 0 R /S /H1 /Type /StructElem >>
[ 11 0 R 13 0 R 14 0 R 15 0 R 16 0 R 17 0 R 18 0 R 20 0 R 22 0 R 21 0 R
  24 0 R 25 0 R ]
<< /K [ 1 ] /P 10 0 R /Pg 28 0 R /S /P /Type /StructElem >>
<< /K [ 2 ] /P 10 0 R /Pg 28 0 R /S /H1 /Type /StructElem >>
<< /K [ 3 ] /P 10 0 R /Pg 28 0 R /S /P /Type /StructElem >>
<< /K [ 4 ] /P 10 0 R /Pg 28 0 R /S /H2 /Type /StructElem >>
<< /K [ 5 ] /P 10 0 R /Pg 28 0 R /S /P /Type /StructElem >>
endstream
endobj
…

stream
/H1 <</MCID 0>> BDC q
…
[(Fi)4(rst )-5(Headin)-4(g )3(\(L)-3(ev)-5(el 1)4(\))] TJ
…
EMC
/P <</MCID 1>> BDC
% draw paragraph content
EMC
/H1 <</MCID 2>> BDC
…
[(Sec)-3(on)3(d )-5(Headin)-4(g )3(\(L)-10(ev)-5(el 1)4(\))] TJ
…
```

Figure 3: Example showing both the structure tree and linked text stream

In the figure, the structure tree is defined first and is later followed by an object stream that draws the content. In the structure tree, the line that begins “`<< /K [ 0 ]`” has the following meaning:

- “<<…>>” is a dictionary containing key/value pairs (e.g., key “/S”, value “/H1”)
- /K [0] is an array (in this case with size 1) that refers to the children (“Kids”) of this node. The children can be another structure element or (as in this example) refer to marked content (“`<</MCID 0>>`”) in the content stream.
- “`/P` 10 0 R” is a reference to the parent object of this element (in this case, /Document)

- “/Pg 28 0 R” refers to the object number of the page containing the element
- “/S /H1” is the name of the structure (H1, P, etc.)
- “/Type /StructElem” identifies this dictionary is a structure element

In the content stream, “BDC” (Begin Demarcated Content) and “EMC” (End Marked Content) mark the beginning and ending of the text being pointed to from a structure tree element.

Although the details are a bit messy, the concept is relatively simple: the structure tree describes the logical structure and reading order by pointing to positions in the (marked) content stream. Those points contain the text to read. Note that just as text can be split into multiple objects, the structure tree can also be split across objects (via pointers to children structure elements) allowing for more local generation of an object presentation and its corresponding logical structure.

2 Math Accessibility

From the previous section, it should be clear that math accessibility in PDF requires using PDF’s structure tree. For math, PDF has a “formula” type that can be used to attach “alt” text to an equation – this is the equivalent of where math accessibility on the Web was prior to MathML being used by most AT for reading math in documents. Using static text for math doesn’t allow for braille generation, exploration of the equation, or tailored text to the reader’s needs/disabilities. Alt text also suffers with prosody and pronunciation problems, especially with respect to not always speaking the variable “a” with the long “a” sound in English.

To explore accessibility solutions for math in PDF, in 2006 we undertook an experiment to modify MathType’s PDF generation to generate begin/end marks in the content stream and MathML in the structure tree that pointed to those marks.[†] A sample Adobe Acrobat app was built that connected with MathPlayer to speak an expression in the tagged PDF and highlight what was being spoken. This demonstrated the feasibility of using MathML as a structure tree tag set and the work was brought to a standardization meeting for PDF the following year.

The success of the demonstration began a slow process of adding MathML into the PDF specification. The work was slowed in part because of the transition of PDF from Adobe to ISO (International Standards Organization). In 2008 ISO released ISO 32000 (ISO 2008). Following that, work began on a new PDF version: ISO 32000-2 [PDF 2.0] (ISO 2020). It was released in 2017 and the revision included recognition of the MathML namespace.

A separate PDF/UA (Universal Accessibility) committee released ISO 14281 (ISO 2014) in 2012 providing requirements for making PDF documents accessible. Because MathML was not officially part of PDF 1.0, the committee could only state that alt text is required along with stating a requirement that all mathematical characters have a Unicode mapping specified if one is not known from the font. Work on PDF/UA 2 is underway and the current draft calls for the use of MathML in one of two ways: direct mapping of MathML tags to the corresponding textual content or using Associated Files (AF). Both methods can be combined.

MathML contains a number of tags that describe notational layout. Examples include “mfrac” (fraction), “msup” (superscript), and “mrow” (horizontal juxtaposition). It also includes tags for textual content: “mi”, “mn”, and “mo” (identifier, number, operator). The MathML for the expression πr^2 is shown in Figure 4.

[†] The experiment did not reach full maturity and was never released as part of MathType.

```
<math>
  <mrow>
    <mi>&pi;</mi>
    <mo>&#x2062;</mo>   <!-- invisible times -->
    <msup>  <mi>r</mi> <mn>2</mn>  </msup>
  </mrow>
</math>
```

Figure 4: MathML example

The corresponding tagged PDF is shown in Figure 5. In the figure, the relevant part of the structure tree is shown followed by the relevant part of the content where the math is drawn. All math is enclosed in a “formula” tag. Its only child is the “math” tag, and so on. Each “StructElem” entry children (“/K […]”) points to either the marked content sequence containing the commands (element

```
% structure element objects
44 0 obj
<< /K [ 57 ] 0 R /P 23 0 R /S /Formula /Type /StructElem >>
endobj
57 0 obj
<< /K [ 60 ] 0 R /P 44 0 R /S /math /Type /StructElem >>
endobj
60 0 obj
<< /K [ 48 ] 0 R 61 0 R ] /P 57 0 R /S /mrow /Type /StructElem >>
endobj
48 0 obj
<< /K [ 24 ] /P 60 0 R /Pg 12 0 R /S /mi /Type /StructElem >>
endobj
61 0 obj
<< /K [ 49 0 R 50 0 R ] /P 60 0 R /S /msup /Type /StructElem >>
endobj
49 0 obj
<< /K [ 26 ] /P 61 0 R /Pg 12 0 R /S /mi /Type /StructElem >>
endobj
50 0 obj
<< /K [ 28 ] /P 61 0 R /Pg 12 0 R /S /mn /Type /StructElem >>
endobj
% content stream leaf example
/mi <</MCID 24>>BDC
BT
… draw ‘π’ …
ET
EMC
```

Figure 5: PDF example showing fully tagged MathML[‡]

such as “mi”) or to another StructElem (e.g, “/K 60 0 R”)

As can be seen, this produces a lot of fine grain structure. In particular, it requires the PDF generator to know about the layout of the math, something that is not always an option. For example, the generation of PDF from TeX cannot easily produce this markup. There are several translators that will translate a subset of TeX commands to MathML, but they work at a higher level than the low-level output generated by TeX (Fischer, 2020).

The PDF/UA 2 draft allows for a second approach: use an Associated File (AF) on the “Formula” structure element. The AF would typically be embedded in the PDF document as an object in the

[‡] Neither MathType nor Word include the invisible times in the PDF they generate.

content stream. An example is shown in Figure 6. The first object shown is the “Formula” object in the structure tree. It has an “AF” key that points to a file specification dictionary which in turn points off to the actual contents of the file (the MathML). At the time of writing, Adobe’s accessibility interface (Adobe, 2008) does not have a way to access an AF although it is available via the full interface; the PDF/UA 2 draft requires PDF processors to provide access to the AF, so Adobe’s API will likely incorporate this ability in the future.

```
…
44 0 obj   % structure element object for Formula
<< /K [ 45 0 R ] /P 23 0 R /AF 100 0 R /S /Formula /T /StructElem >>
endobj
…
100 0 obj  % File specification dictionary -- points to contents
<< /EF << /F 101 0 R >> /F (Eq1.mml) /UF (Eq1.mml)
   /AFRelationship /Supplement /Desc (MathML) /Type /Filespec >>

101 0 obj  % embedded file stream
<< /Subtype /application#2fmathml+xml /Length 100
   /Params <</ModDate (D:20210108143734)>> >>
stream
<math> <mrow> <mi>&pi;</mi> <mo>&#x2062;</mo>
<msup> <mi>r</mi> <mn>2</mn> </msup> </mrow> </math>
endstream
endobj
…
```

Figure 6: PDF example showing MathML in an AF

While the AF approach is often simpler and easier for software to generate, it has the drawback that synchronized highlighting of speech is much harder because there are no longer links from the MathML elements being spoken to the corresponding rendering in PDF. One solution is to create an overlay and re-render the math onto the overlay. This requires both a math rendering engine and also the ability to closely match the existing PDF drawing. Matching the PDF is possible because the fonts used are known by examining the PDF content. However, this approach places a large burden on AT.

3 Why Bother with PDF in 2021?

HTML is by far the most common way to deliver information on the Web. Using a “filetype:” search with Bing, there are 185 billion sites using HTML and 2 billion sites using PDF.[§] While the number of HTML pages dwarfs the number of PDF pages, it says nothing about the amount of information stored – PDF documents tend to be much longer than HTML pages. Another way to look at this question is an approach taken by Johnson (Johnson, 2015): compare file type usage over time. Using his numbers from the past and adding on numbers from this year, we can see that relative usage of PDF compared to non-HTML formats is large and has possibly increased. Table 1 (updated from (Johnson, 2015)) shows the relative percent of document formats on the web besides HTML.

Another way to look for whether PDF relevance is decreasing is to use Google trends. Figure 7 shows the results of people (worldwide) who searched for “PDF”. The numbers are noisy, but they indicate interest in PDF has not decreased.

[§] Google’s filetype search did not work for PDF. Using the slightly different search key “site:* filetype:html” (and the same for PDF), the results were significantly different than the Bing results: 10.5b and 1.2b. Hence, the numbers presented in this section should be taken with a grain (or maybe a kg) of salt.

Table 1: Relative PDF usage over time[**]

	2011, Apr	2013, Jan	2015, Feb	2021, Jan
PDF	81%	79%	79%	96%
DOCx	13%	17%	18%	3%
XLSx	3%	2%	2%	1%
PPTx	3%	1%	2%	1%

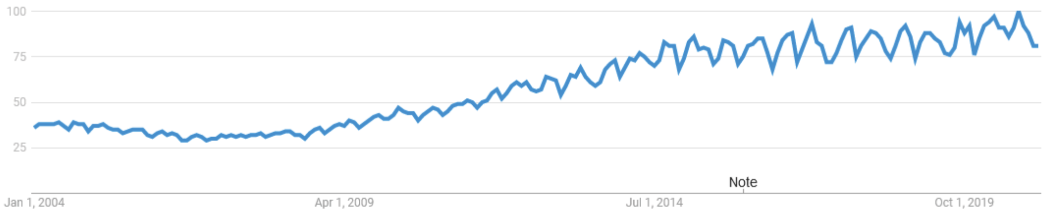


Figure 7: Google trends search results for "PDF"

These numbers encompass all types of material. This paper is focused on math, so STEM documents are the most relevant ones to consider when asking if PDF remains an important format. Probably the largest source of technical papers is arXiv; PDF remains the primary file format used by arXiv. On Google Scholar, PDF dominates also, accounting for ~79% of results.[††] At dl.acm.org, ~50% of the pages are PDF. Hence, PDF remains very important for technical material.

Summary

PDF accessibility requires a tight linking between two trees: the content stream and the structure tree. Math is highly structured, so this linking means lots of coordination between rendering the math and making the math accessibility. The burden is eased by using associated files, but the ability to synchronously highlight the math while it is spoken is greatly compromised. Contrast this to HTML, which uses a single tree for math that is relatively easily generated from a variety of sources. That tree serves as the source for generating speech and braille as well as the source for rendering. Creating accessible PDF documents is clearly harder than doing the same in HTML: 2 trees is not better than 1.

If an accessible PDF can be created, making the math accessible is not hard. In fact, since NVDA started using MathPlayer to speak math in HTML, Word, and PowerPoint documents in 2015, it has also had the ability to do the same for PDF documents (if the MathML is fully given in the structure tree). Unfortunately, with only sample PDF documents available that are properly tagged, this is a solution still waiting to be used.

[**] Values for 2011, 2013, and 2015 come from (Johnson, 2015). Data for 2021 are from a Google search using “site:* filetype:…”. Both old and new formats for Office apps were searched (e.g., doc and docx). The 2021 search technique may differ from the previous search technique.

[††] This was determined by using a filetype search on google at Google Scholar. Using “filetype:html” resulted in a large number of PDF results. The top 100 results were examined and 26 were HTML, with the remainder being PDF. The filetype results for HTML were multiplied by .26 when determining the 79% number. 8.1% were Word docs.

References

Adobe. (2008). *PDF Accessibility API Reference*. Retrieved January 4, 2021, from https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/access_v9.pdf
Fischer, U. (2020). Creating Accessible PDFs with LaTeX. *TUGboat*, *41*(1), 26-28.
International Organization for Standardization. (2008). *Document management — Portable document format — Part 1: PDF 1.7* (ISO 32000-1:2008). Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:32000:-2:ed-2:v1:en
International Organization for Standardization. (2014). *Document management applications — Electronic document file format enhancement for accessibility — Part 1: Use of ISO 32000-1 (PDF/UA-1)* ISO 14289-1:2014). Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:32000:-2:ed-2:v1:en
International Organization for Standardization. (2020). *Document management – Portable document format* (ISO/DIS Standard No. 32000-2, Second edition). Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:32000:-2:ed-2:v1:en
Johnson, D. (2015, October 7). *PDF still dominates electronic documents online*. Retrieved January 4, 2021, from http://duff-johnson.com/2015/10/07/pdf-still-dominates-electronic-documents-online/

Appendix

Several of the figures in this paper include PDF. Table 2 explains some of the syntax used in the examples so that they are more easily understood.

Table 2: Quick Guide to PDF syntax used in examples

Syntax	Meaning	Examples
%	comment	`% structure dictionary entry`
/	name, usually defined in spec	`/Type, /K`
(…)	string	`(x), (this)`
[…]	array	`[1], [<<…>>, 2]`

<<…>>	dictionary of key value pairs, order of key/value pairs is not important	`<< /Length 95 >>,`
		`<< /K [ 1 ]   /P 10    /Pg 19`
		`   /S /P      /Type /StructElem >>`

n 0 R	Reference to the n^{th} object	`100 0 R`

Conversion of School textbook PDF to EPUB3

Masakazu Suzuki
Science Accessibility Net
msuzuki@sciaccess.net

Abstract

We have been working on the digitization of school textbooks for math and science to convert them into accessible audio books: Multimedia DAISY and EPUB3 with aloud reading as media overlay for a long time (since 2009). The audio books as the result have been mainly distributed to the disabled students with dyslexia by the Japanese Society for Rehabilitation of Persons with Disabilities (JSRPD).

In the process, we have been working on the improvement of the audio-book quality and also on the investigation of the methods to reduce time/effort for producing the audio books from PDF.

In the keynote lecture, I will talk about our recent workflow for this job, in which the management of collaborating automatic processes with manual works is crucial. Unsolved problems and steps to improve in our workflow will be also discussed.

1 Introduction

In addition to braille, a LaTeX source or a document in which a LaTeX source is embedded usually plays an important role as accessible STEM (science, technology, engineering and math) contents for the blind. For instance, concerning PDF, you can produce an accessible content in the background of which the LaTeX source is embedded. Blind people can read it with a screen reader or a refreshable braille display. On the other hand, concerning the other-type print-disabled people such as ones with severe low-vision or dyslexia, audio-embedded e-books are useful/important. Multimedia DAISY (Digital Accessible Information System) or accessible EPUB3 is recently regarded as the most-promising international standards of such (accessible) e-books [1].

In European countries, if text-based information is properly stored, a TTS (text-to-speech) engine usually can read it out without any trouble. However, in Japanese, four different character sets are used simultaneously in print: Chinese characters (Kanji), Hiragana, Katakana and alphanumeric letters. While Hiragana and Katakana (or Kana, generically) are essentially kinds of phonetic symbols, a single Kanji or a compound of the characters usually has several ways of pronouncing, according to its context. In STEM, they are often read in a different manner from the usual. As the result, a TTS engine tends to make mistakes quite frequently in reading out Japanese texts. In particular, school textbooks should be read out correctly, and you do need to embed audio files of aloud-reading that is corrected manually in advance.

In Japan, “the Japanese Society for Rehabilitation of Persons with Disabilities (JSRPD)” has been providing print-disabled students with accessible e-textbooks in multimedia DAISY since 2008 [2]. In cooperation with many volunteer groups and non-profit organizations, they produced multimedia-DAISY versions of about 80% textbook titles used in elementary and junior-high school. In 2019, those textbooks were provided to about 12,000 print-disabled students (mostly ones with dyslexia). Incidentally, JSRPD is now preparing to change the format of those e-textbooks to accessible EPUB3 in the near future, which includes audio files of aloud-reading as media-overlay.

Our non-profit organization: “Science Accessibility Net (sAccessNet)” has also joined this project since 2009 and been mainly working on conversion of STEM textbooks into multimedia DAISY [3]. We have already released about 200 titles of such accessible e-textbooks by this year. This conversion from original PDF into multimedia DAISY consists of automatic (computerized) process and manual error-correction. This process has been improved year by year, and its production efficiency and quality are getting extremely better than in the early days.

In this keynote, I introduce briefly our latest workflow to produce STEM text books in multimedia DAISY/accessible EPUB3. In this lecture, hereafter, I refer to this process simply as “DAISY-making”; here, DAISY means both multimedia DAISY and EPUB3 with aloud reading as media overlay.

In Japan, so-called “Making Textbooks Barrier-Free Act” came into effect in 2009, and the government requires every publishers of school textbooks to provide digital version (PDF) of their original textbooks to groups/organizations who produce their accessible versions such as braille, large print, DAISY, etc. In those PDF, all text information is embedded, and you do not need to use OCR (optical character recognition) to recognize a scanned page image. We can assume that the text information can be extracted directly from the given PDF in the production process. As will be discussed later, we can also extract characters/symbols in math parts, and in principle, we do not need image-based OCR for the conversion (actually, however, image-based OCR is still used slightly even in the latest workflow).

In Japan, some people tend to be negative about audio-based STEM textbooks; they insist that it is hard for print-disabled student to understand math only with speech. However, here, I would like to point out that repeated listening of a spoken math formula often helps them with more deeply understanding it than simply looking at it, and audio-based textbooks should be helpful for math education.

2 Conversion Process of Japanese Texts and Graphics

As was discussed, in DAISY-making of a textbook in Japanese, it is important to embed the audio data of correct aloud-reading. Furthermore, we have other additional important tasks that should be achieved in the Japanese-text conversion.

Many Kanji characters have more complicated forms than Kana. The number of different Kanji is too many, and some of them resemble each other. Thus, dyslexic people are usually poor especially at reading Kanji. Someone says that Kanji looks just a kind of meaningless picture, not a letter. Even after reaching their adulthood, some of them can read only very-restricted simple ones. On the other hand, many of them can read Kana, and placing a pronunciation in Kana alongside Kanji usually helps them very much with understanding. Thus, there is a big demand for a textbook in which kana readings, called “Ruby,” are added beside all the Kanji characters in a piece of writing.

Different from European languages, there are no spaces between words in ordinary Japanese texts, and many dyslexic people have a difficulty in realizing where the breaks are. It often takes a very long time for them to realize, and that situation often brings about their print disability. Actually, a survey of children/students' (DAISY textbook users) demands by JSPRD shows that about 30% of them become able to read Japanese texts just by adding word breaks with spaces or vertical bars.

In the 6-dots braille system, a single cell can represent only 63 print characters, and there is no Kanji in Japanese braille. 46 single-cell braille are assigned to Kana (Katakana). Thus, in transcription, you have to change a Japanese print text to one in Kana only. In addition, word breaks become very ambiguous in the Kana-only text; the segmentation of a Japanese text usually depends on Kanji included in it. Thus, you also have to add the necessary word breaks in the Kana conversion. Japanese braille translators have a long history of development, in which a syntactic analysis technology has been widely used to convert a print text in Japanese correctly into one in Kana with word breaks. To convert a textbook into DAISY for the dyslexia, the application of that technology is also very effective to add Ruby beside all Kanji and word breaks. Thus, in the process of producing STEM textbooks in DAISY, sAccessNet uses an application to change automatically all Japanese texts in that manner. Certainly, however, the application often makes errors in the syntactic analysis, and we have to correct them manually.

A special characteristic of STEM textbooks in Japan is probably that graphics play a definitely important role. Many figures or other graphic representations are used in STEM textbooks in European countries as well as Japan. In those textbooks, however, thoughtful and scrupulous explanation for the graphics is usually given in articles so that readers can understand a content even if those graphics are hard to read directly. On the other hand, in Japan, authors usually lay emphasis on getting children/students to understand just with graphics. They tend to write necessary explanation directly in the graphics, and overlapped explanation in the article is often skipped. In many cases, it is almost impossible for a reader to understand a content without looking at the graphics. Thus, it is also very important to realize a function to read out text parts embedded in the graphics with highlighting them. ChattyInfty allows you to produce STEM contents in multimedia DAISY including graphics that have such function. I intend to demonstrate that in the keynote.

3 Conversion from PDF to Structured Text (IMLX)

In DAISY-making of a textbook in Japanese, we use software named "ChattyInfty," with which you can author easily various STEM contents including figures, tables, math expressions, chemical formulas or other technical notations [3]. ChattyInfty contents can be exported as various accessible formats including multimedia DAISY and accessible EPUB3. Editing a document structure such as multi-level sections (index in DAISY) or itemization, etc. is also very easy. In the Japanese version of ChattyInfty, a high-quality TTS engine is bundled, and it can read out an entire content including technical parts in a very natural manner. As you listen to that TTS voice, you can tune up the aloud reading or correct reading errors with the keyboard and mouse operation.

"IMLX" is a file format of ChattyInfty, which is a kind of XML. Here, I discuss a process to convert PDF into IMLX that is a structured text. In this conversion, our OCR system for a STEM document named "InftyReader" also carries out a main purpose [3]. In principle, this IMLX does not include audio files of aloud reading; it is essentially equivalent with a text-based EPUB3, and bidirectional conversion can be done between them.

In the latest way, our workflow of the conversion process is as follows: (here, LIF is "Layout Information File" in InftyReader.)

(1) Converting PDF into a file in SVG (Scalable Vector Graphics).

(2) Recognizing the SVG file as a pre-processing to get character information and to generate a LIF for it.

(3) Correcting errors in the LIF manually.

(4) Re-recognizing the SVG file with the corrected LIF and converting the result into IMLX.

(5) Correcting errors in the IMLX manually.

(6) Finally, converting the corrected IMLX file into a text-based EPUB3.

In this flow, the step (1): PDF to SVG conversion and the step (2): the LIF generation carry out core purposes.

In the step (1), PDF is converted into SVG consisting of three-type elements: character, image and path elements, where the path is a set of point sequences to draw lines, arcs and Bezier curves. A new PDF parser named "PDFcontentExtracter," which has been recently developed by Fujiyoshi, is very powerful in achieving this task [4]; it provides us with the font rect-area (rectangular area) together with their character code. (As for graphics in PDF, we also can get path information to make them outline (rect-area).) Many other PDF parsers provide us with only font rect-areas. To recognize correctly a math expression consisting of multiple characters/symbols, in conventional math OCR method, the rect-area information given by the parser has been usually combined with results of image-based OCR to estimate the real rect-area size of the original character image [5, 6, 7]. However, in such method, we cannot avoid a certain (large) number of recognition errors that are in most cases caused by misrecognition in the image-based OCR process. (In general, character/symbol images extracted from PDF are clear enough, but they often touch each other in a math expression in the original PDF; such touched characters usually bring about the misrecognition.) As for PDFcontentExtracter, it provides us with path information to make characters/symbols outline (rect-area) as well, and we always can get their exact font rect-area information, which is very effective to improve recognition accuracy for math expressions. Furthermore, in the SVG file, a page content is classified into character, image and path elements, and that is also very useful to develop an application to segment a complicated-layout page automatically into text areas, graphic areas, table areas, etc.

Elements defined in LIF is as follows:

Text (including math expressions), Graphics, Table, Logo, Character (dramatis personae), Caption, Page number.

That is, in the step (2), InftyReader divides a page automatically into the elements (areas) with these tags. In addition, text areas are classified into main articles and sub-texts such as balloons, and captions are labeled into different-types: ones for a table or a figure, dialogs by a character, etc. The main articles are further divided into paragraph blocks. The LIF also has the reading-order information of those elements including graphics and tables. "Infty Layout Editor" allows you to correct a way to segment, tags for each element and the element order.

In Japanese textbooks, the header number of sections/chapters, the title of balloons, etc. are all represented as colorful logos. Children/students listen to aloud reading embedded in a DAISY textbook in a class room as they look at it. Thus, the logos are not an ornament, but they play a very important role to specify which part is now read out. Dialogs among dramatis personae and/ mascots are also important to understand a content since a story is often developed just with the dialogs, or a key issue is explained in that manner. Thus, in DAISY making, information concerning "Who speaks of what?" is indispensable to understand clearly, and Dialog tags also show that attribute. For the present, unfortunately, InftyReader does not have a layout-analysis ability to assign those specific tags appropriately, and we have to do the LIF correction manually in the step (3).

In the step (4), the SVG generated by PDFcontentExtracter is combined with the corrected LIF, and InftyReader re-recognizes it and converts the result into IMLX. This job is automatically processed.

Its conversion accuracy is very high, and we no longer need to spend a lot of time and effort to correct the recognition result in the step (5) while that correction process is usually most dominant in ordinary workflow of PDF-DAISY conversion.

Concerning the step (6), ChattyInfty has a function to export, by one-click, IML as various accessible format such as human-readable TeX, text-based EPUB3, etc.

4 Production of Audio-Embedded DAISY/Accessible EPUB3

As was discussed, in Japanese, a TTS engine frequently makes reading errors, and we do need to embedded audio files of correct aloud-reading into DAISY as media overlay. In addition, a Japanese text should be converted so that Ruby (kana readings; see the section 2.) beside all Kanji and word breaks are added. To carry out these tasks, we have developed an application, in which a braille-translation engine works as back-end, to add automatically Ruby beside all Kanji and word breaks. We have also developed a user interface of ChattyInfty to allow a user to correct reading errors easily [8]. As was mentioned previously, in the Japanese version of ChattyInfty, a high-quality TTS engine ("AI Talk") is bundled, and it can read out an entire content including technical parts in a very natural manner. As you listen to that TTS voice, the interface allows you to tune up the quality of aloud reading such as intonation with the keyboard and mouse operation.

Concerning non-technical contents used in a daily life, AI Talk is very smart and read out them quite properly. However, as for other type contents such as STEM or history, a single Kanji or a compound of the characters are often read in a different manner from the usual, and AI Talk frequently fails at reading out them. Thus, we edit a specific reading dictionary for each field by picking up special terms included in them. By incorporating the dictionary in AI Talk, it becomes able to read out such contents more properly. However, even in this manner, we cannot avoid that a certain number of wrong aloud-reading occur. To solve this problem, in ChattyInfty, an interface is implemented to control how to read out each term locally if necessary [8]. We refer to this concept of assigning a pronunciation as "Yomi" (a Japanese word that means "a manner of reading aloud").

Concerning math expressions, structured math syntax is automatically converted into a correct spoken math by ChattyInfty, and it can read out any math expressions quite properly. You seldom need to correct manually. Actually, as far as math parts are concerned, we usually do not make any corrections in DAISY-making.

Concerning texts in graphics, in ChattyInfty, a function is implemented to assign aloud reading (Yomi) to any selected letter/phrase/sentence/math expression embedded as a text in graphics. When exporting a content in DAISY, we place an image corresponding to the selected part over the text (for reading out) in the graphics and assign SPAN ID to the image. Those images as well as texts in an article are linked to their audio files of aloud reading as media overlay so that a popular DAISY browser can play back the content properly. When browsing the graphics, the audio files are played back in a proper order together with synchronized highlighting of texts/math expressions in the graphics. For the present, there is no standard manner to assign SPAN ID to each SVG element and to link them to their audio file. Thus, we are tentatively compelled to take such a tricky way. We wish the DAISY Consortium would establish the standard way of media overlay for SVG elements in the near future.

That is the entire workflow in which we produce an accessible textbook in multimedia DAISY or audio-embedded EPUB3 from PDF. It should be particularly emphasized that you do not need to edit audio files, themselves directly in the workflow. The audio file of aloud reading is generated from corrected IMLX which can be authored just with keyboard or mouse like an ordinary word processor. That feature should reduce the level of difficulty and be a key issue to increase the number of people who are involved in DAISY-making, I believe.

5 References

1. DAISY Consortium (n.d.), "Home, DAISY Consortium,"
   Available at: http://www.daisy.org/ (accessed 10 January 2021).
2. Japanese Society for Rehabilitation of Persons with Disabilities (n.d.),
   "Multimedia-DAISY Textbooks," Available at:
   http://www.dinf.ne.jp/doc/daisy/book/daisytext.html (Japanese only, accessed 10 January 2021).
3. sAccessNet (n.d.), "About sAccessNet,"
   Available at: http://www.sciaccess.net/en/ (accessed 10 January 2021).
4. Kohase, K., Nakamura, S. and Fujiyoshi, A. (2020),
   "Layout Analysis of PDF Documents by Two-Dimensional Grammars for the Production of Accessible Textbooks,"
   Proc. the 17th International Conference on Computers Helping People with Special Needs, Springer, Lecture Notes in Computer Sciences 12376, pp.321-328.
5. Baker, J., Sexton, A. and Sorge, S. (2008),
   "Extracting Precise Data from PDF Documents for Mathematical Formula Recognition,"
   Proc. 8th International Workshop on Document Analysis Systems, Nara, DOI: 10.1.1.159.8601.
6. Baker, J., Sexton, A. and Sorge, V. (2009),
   "A Linear Grammar Approach to Mathematical Formula Recognition from PDF,"
   Proc. International Conference on Intelligent Computer Mathematics 2009, Springer, Lecture Notes in Computer Sciences 5625, pp.201-216.
7. Suzuki, M. and Yamaguchi, K. (2016),
   "Recognition of E-Born PDF Including Mathematical Formulas"
   Proc. the 15th International Conference on Computers Helping People with Special Needs, Linz, Springer, Lecture Notes in Computer Sciences 9758, pp.35-42.
8. Suzuki, M., Kanahori, T. and Yamaguchi, K. (2014),
   "Necessities for Math-Access Tools with Speech"
   Proc. the Annual CSUN assistive Technology Conference, California State University - Northridge, Journal on technology and persons with disabilities, Vol.1, pp.123-134.

Extracting Precise Coordinate Information of Components from E-Born PDF Files

Shunsuke Nakamura, Kento Kohase, and Akio Fujiyoshi

Graduate School of Science and Engineering, Ibaraki University, Hitachi, Ibaraki, Japan
{20nm726r,20nm711h,akio.fujiyoshi.cs}@vc.ibaraki.ac.jp

Abstract

This study introduces a tool to extract information of components from PDF files. Recently most PDF files become e-born, that are electronically produced by a word processor, desktop publishing software or LaTeX system. Since an e-born PDF file contains all necessary information how to print it out, the precise coordinate information of components can theoretically be extracted. However, there are some difficulties caused by multiple use of coordinate transformations, abnormal definition of fonts, relative arrangement of objects, etc. By pursuing the rendering process of a PDF document, the tool can extract the precise coordinate information on the base coordinate system of the document.

1 Introduction

In our daily life, we often use digital documents in the Portable Document Format (PDF). Recently most PDF files become e-born, that are electronically produced by a word processor, desktop publishing software or LaTeX system. Since character information (character code and font name) can be extracted from an e-born PDF file, the information can be used by screen readers and OCR software. For the recognition of STEM documents with mathematical formulae, however, precise coordinate information (position and size) of components is also necessary because the relative position of symbols must be precisely recognized.

This study introduces a tool to extract the precise coordinate information of components from PDF files. Though an e-born PDF file contains all necessary information how to print it out, the extraction of the precise coordinate information of components is difficult because of multiple use of coordinate transformations, abnormal definition of fonts, relative arrangement of objects, etc. The tool pursues the rendering process of a PDF document and can extract the precise coordinate information on the base coordinate system of the document.

The tool, named as “PDFcontentExtracter”, is a member of simple processing tools for PDF files developed in the authors’ laboratory [1]. The tool is developed in Java using Apache PDFBox library [2]. Apache PDFBox library is an open source Java library for creation of new PDF documents and manipulation of existing documents, published under the Apache License v2.0. The tool uses the library to obtain raw data of components from a PDF file. The tool can also be used to extract images in a PDF file. All images are converted to PNG format. In addition, for the purpose of confirmation of extraction, the tool can reconstruct the document in SVG format from all extracted information of components of a PDF file. The simple processing tools for PDF files are available on the website of authors’ laboratory: http://apricot.cis.ibaraki.ac.jp/PDFtools/.

2 Overview of PDF Documents

2.1 Types of Components

As shown in Figure 1, the main components of a PDF document are characters, images, paths, and shadings. Since shadings are mostly used for decoration and seem not so important, the tool extracts the coordinate information of components of the other three types. The coordinate information of paths is important for STEM documents with mathematical formulae because they are used as fraction lines and a part of radical signs.

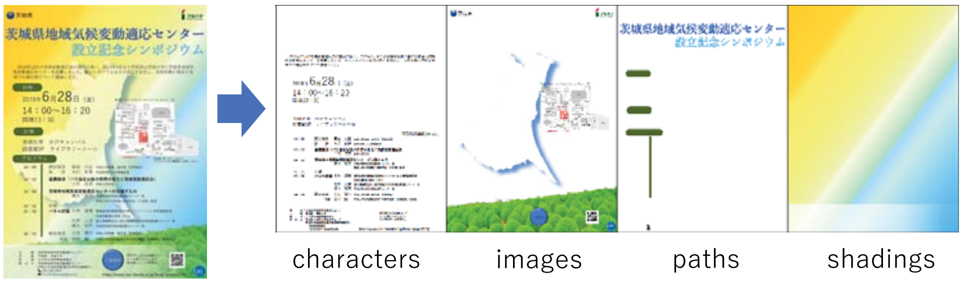


Figure 1: Main Components of a PDF File

2.2 Coordinate Systems

In a PDF document, there are several coordinate systems in which the coordinates specifies how graphical objects should be drawn. Transformations among coordinate systems can be defined by transformation matrices, which specify linear mappings of two-dimensional coordinates such as translation, scaling, rotation, reflection, and skewing.

The base coordinate system of a page of a PDF document is called "*user space*". The origin of user space corresponds to the lower-left corner, and the length of a unit along both x and y axes is set by 1/72 inch. A rectangle in user space, called "CropBox", specifies the visible area of a printed page.

Since coordinate systems can be modified using the coordinate transformation operator, several coordinate systems exist in a same page of a PDF document. A coordinate system for text is called "text space", and character glyphs in a font are defined in "glyph space." Glyph spaces are relatively placed in a text space by a combinations of text-positioning parameters such as character spacing, word spacing, horizontal scaling, text leading, and text rise. For most types of fonts, 1 unit of text space equals 1000 units of glyph space.

Each image has its own coordinate system, called "image space." An image occupies a rectangle in image space w unit width and h unit height, where w and h are the width and the height of the image. The origin of image space is at the upper-left corner of the image.

A minipage within another page can also be defined with its own coordinate system, called "form space."

The tool transforms all coordinate systems of the components of a page of a PDF document into user space of the page.

3 Coordinate Information Extracted by the Tool

The tool extracts the coordinate information of a PDF document for the three types of components (characters, images, and paths) separately. All coordinates are transformed as in user space (the base coordinate system).

For characters, two types of rectangle boxes are extracted by the tool. Figure 2 shows the coordinate information of characters extracted from a PDF document with a mathematical formula produced by LaTeX system. The blue box, called “fontBBox”, represents the smallest common rectangle bound of drawn areas of all glyphs in a font, while the red box represents the actual rectangle bound of drawn area of the glyph. The fraction line and a part of the radical sign are paths, and their shapes and bounding boxes are extracted, drawn as thick orange boxes in the figure.

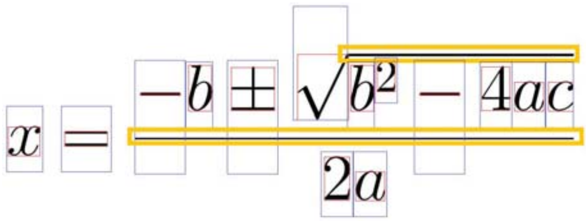


Figure 2: Coordinate Information for Characters Extracted by the Tool

For images, two types of rectangle boxes are extracted by the tool. Figure 3 shows the coordinate information of an image extracted from a PDF document. The blue box represents the actual rectangle bound of drawn area of the image, while the green box represents the imaginary rectangle bound of image with the same scale and position but without rotation, reflection and clipping.

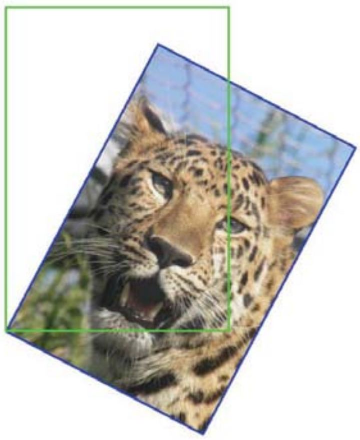


Figure 3: Coordinate Information for an Image Extracted by the Tool

4 Simple Processing Tools for PDF Files

The tool, named as “PDFcontentExtracter”, is a member of simple processing tools for PDF files developed in the authors’ laboratory [1]. Other tools are “PDFcontentEraser” and “PDF-fontChanger.”

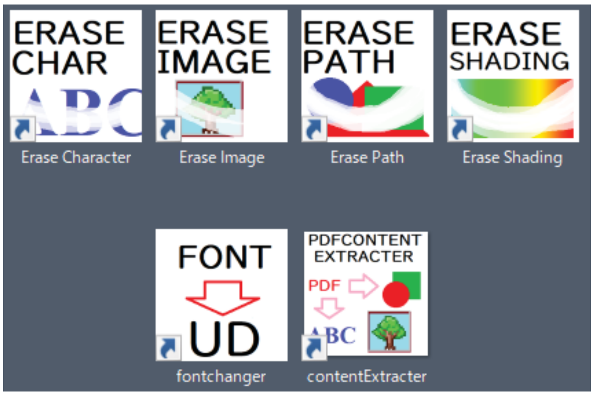


Figure 4: The Icons of the Simple Tools

PDFcontentEraser is a tool to remove a certain type of components in a PDF file. This tool can selectively erase all components of a certain type.

PDFfontChanger is a tool to change a selection of fonts in a document. As a default setting, all fonts are replaced by “Universal Design Font for Digital Textbooks” (UD font) [3] developed by Morisawa Inc.

The usage is very simple. When they are installed on a PC with Windows, icons are placed on the desktop as shown in Figure 4. A PDF file is processed by a drag-and-drop operation to these icons.

The simple processing tools for PDF files are available on the website of authors’ laboratory: http://apricot.cis.ibaraki.ac.jp/PDFtools/.

References

[1] Shunsuke Nakamura, Kento Kohase, and Akio Fujiyoshi, A Series of Simple Processing Tools for PDF Files for People with Print Disabilities, Proceedings of the 17th International Conference on Computers Helping People with Special Needs (ICCHP 2020), LNCS 12376, pp. 314-320, 2020/09

[2] Apache PDFBox, https://pdfbox.apache.org/

[3] Morisawa Inc., Universal Design Fonts (in Japanese), https://www.morisawa.co.jp/fonts/udfont/

Are Statistics Courses Accessible?Part II: Revisited with 2020 hindsight

A. Jonathan R. Godfrey

Massey University, Palmerston North, New Zealand
a.j.godfrey@massey.ac.nz

Abstract

Statistics courses in the modern era rely less and less on pen and paper and more and more on the use of computers and specific software. This updated paper reconsiders the state of the accessibility of statistics courses for vision impaired students in 2020.

There are four key elements contributing to this accessibility; graphics, software, tables, and mathematical formulae. Each element is addressed but greater emphasis is given to the accessibility issues for statistical software and the solutions offered through its most recent developments. The author calls upon personal experiences as a blind person using statistical software over the last twenty-five years, initially as a student and currently as a university lecturer. The current state of statistical software offers choice and opportunity as against the previous 'choose the least bad option'. The best software continues to open up opportunities that are beyond anyone's expectations of ten years ago.

1 Introduction

This work unashamedly builds on a similarly titled paper presented in late 2009 at the first Workshop on E-Inclusion in Mathematical Sciences (WEIMS) (see Godfrey 2009). Huge gains have been made in the overall access to statistics courses, but as I now suggest, the majority of impact is a consequence of how the statistical world has changed, as compared to the rate of development of relevant adaptive tools used by vision-impaired people. Much of the text from Godfrey (2009) is therefore reproduced with out of date material removed or updated accordingly.

The idea of having to take a statistics course often induces expressions of fear and trepidation. Many of the first year students we encounter hate the prospect of taking a statistics course and phrases like "I suck at math", "I was useless in stats classes at school" and "I'm no good with numbers" are all too common. Anecdotal evidence suggests the blind students have fears too, but these are different, and in my opinion, should be different. It is all too clear that the dominating problem for the blind student is "how am I going to do what they do?"

In Godfrey (2009), I contended that "The author argues that the very real concerns blind students face are not adequately countered by sufficient solutions and encouragement; resources are lacking; role models are few and far between; and the current way Statistics is taught is making things worse not better." Thankfully, times have changed in so many ways, but while the times are different, they are not uniformly better.

Introductory statistics classes are taught to a very wide range of students. Subject areas at many universities that expect or encourage their majoring students to take an introductory statistics course include all business majors, all technology and engineering majors, psychology, geography, economics, biological sciences such as ecology and veterinary science; Starting in 2020, all science students at Massey University are expected to take a course in statistics.

The timing of the introductory statistics course is far from consistent across subject areas and degree programmes. It's often hidden away and guised as something like "Research Methodology for . . . " or ". . . Analysis". Sometimes these courses are left to postgraduate level but most are compulsory elements of the majoring requirements for the related discipline. The point here is that while death and taxes are inevitable, having to take a statistics course isn't far off. In fact, to be sure of not having to take a statistics course means the intending university student probably needs to enrol in a law degree, the fine arts, or one of the Humanities subjects.

Aside from the challenges in actually dealing with numbers, statistics lecturers force students to create and interpret graphs, use a (usually new) software application, and process the mathematical formulae with all that Greek. In addition, many statistics courses also force their students to look up statistical tables. Most mathematics courses do not deal with all of these key elements of the standard introductory statistics course. The vision impaired student, be they blind or having some residual vision, must be ready to overcome the challenges that each of these four elements presents before the semester begins. Let us now address these four elements in turn.

2 Element 1: Graphs

Godfrey (2009) said "There is no doubt that assessing, interpreting, and creating graphics has its challenges for the blind statistician. Working with graphs is also a challenge when assessment of knowledge is undertaken. How does the assessor know whether it is the student who knows that an outlier is highlighted in the histogram, or whether they were told it was there by their sighted reader/writer? Provision of tactile graphics is probably the only solution that will offer direct access to the graph to be examined." Work on producing tactile graphics is well-advanced, but the availability and therefore uptake of tactile media frequently remains out of reach for many blind students today. The willingness to request tactile material seems to be lacking (anecdotal evidence), even though the tools to create tactile representations of graphs on the fly have existed for five years (Godfrey and Murrell 2016). Anecdotal evidence suggests that in addition to this lack of willingness , there is a shortage of personally owned braille embossers in the hands of students who might benefit from tactile images.

Efforts to provide a solution based on the exploration of graphs via a combination of synthetic speech and enlargement of portions of graphs (Godfrey, Murrell, and Sorge 2018) show promise to help anyone who understands a graph's structure, to gain understanding of the particular data instance. This work builds on attempts to automatically generate text descriptions of graphs as they are created (Godfrey 2012).

Despite the existence of these modern tools, blind students continue to rely on sighted individuals to be able to give them the knowledge they would be able to gain were they sighted or had direct access to the graphs through such media as tactile graphs. This remains a concern for any blind person when producing graphs to meet publishing standards, but the vast majority of graphs in statistical analyses form part of the preparatory work, not just part of the final presentation. Ultimately, there is no problem with a blind student asking a sighted person if the graph is showing what was intended. Sighted students ask each other for their opinions, so why shouldn't a blind person do likewise?

Modern teaching practices in statistics courses are increasingly focused on use of graphs as an intermediate step for developing ideas about a given data set. Without adequate substitutes for the graphs being used by sighted people, blind people are at an obvious disadvantage.

Godfrey (2009) stated "There are, however, alternatives to all graphics. Many graphs were designed to be used as highlights for quickly demonstrating a point discovered via analysis by other means. Calculation of the mean and the median as well as midpoints between quartiles would be used to gauge the skewness of a sample before graphs like histograms were so simply created by software. Now that software can create the graph so easily, we rely on this as the method for determining skewness. Exploratory data analysis techniques are prominent in introductory courses sometimes taking up as much as one quarter of the overall work. The old-fashioned ways are still relevant today but are less efficient for the sighted user. They are, of course, a better option for the blind user to consider, but seldom are they taught in the 21st century. Many of these numerical substitutes for graphs are now given practically no exposure courtesy of the ever-increasing use of graphics in every facet of statistics education. Hypothesis tests exist for analyzing the existence of skewness, but hypothesis tests are usually taught well after exploratory data analysis through graphs." Little has changed.

The Godfrey (2009) claim that "Blind students need to learn how to do things differently to their sighted peers if they are to use statistical analyses in their ongoing study or careers. The standard introductory course may not actually provide the necessary content required by the blind student." This remains relevant in 2021.

3 Element 2: Statistical Software

Statistical software has been an important part of most statistics courses in the 21st century. The choices made by teaching staff have a dramatic impact on the approach to teaching in 2021. The most modern courses make use of the software that interacts well with the other digital tools used in the education process.

Godfrey (2009) unashamedly promoted use of R (R Core Team 2020) for offering the best access to software for blind people. Promotion of this access was given by Godfrey (2013) to the R development community. This superiority remains today, but the latest developments available by and for R users are solving many old problems, while also creating new ones.

Access to standard classroom presentation material should not be an issue in 2021 unless the teaching staff are using what is now old-fashioned methods. However, if the staff use the latest interactive tools in R, there is a chance that blind students will be left in the position of being told what happened while sighted students see it for themselves. Teaching staff therefore need to think carefully about how to explain material in words that these demonstrations display. Demonstration using physical objects may well prove an invaluable approach even if it is seen as a retrograde or archaic solution.

The second (and probably most concerning) source of software issues for blind people is the use of software to generate statistical analyses for themselves. The material taught in introductory courses is changing as we are able to rely more and more on statistical software. Even introductory courses are now using code to analyse data in R, in preference to other software that relies on menus and dialogue boxes. Manipulation of data and the creation of graphics are much easier now too, and this might explain why we now have a greater focus on exploratory data analysis techniques in statistics courses. In addition, the days of generating a few histograms, boxplots, and maybe a scatter plot or two are behind us. As an example, the following code generates a graph (intentionally not included) that was well within the expectation of a first year statistics student at Massey University in 2020.

```
>  library(ggplot2)
>  Graph1 = ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() +
>      facet_grid(.~year) + geom_smooth(method="lm")    +
>      xlab("Engine displacement (litres)") + ylab("Highway miles per gallon")
```

N.B. The data used in this example is an original component of the `ggplot2` package (Wickham et al. 2020) used to generate the graph. Removing the irritation of the use of metric and imperial measurement scales is used as an example for improving the graph.

The R code used above is not all that human-readable especially to an R novice, but the `BrailleR` package (Godfrey et al. 2020) will go a long way to making sense of it, and therefore the associated graph. A sighted person who has access to the graph might well work out what each component of the code generates, but a reader of this paper is now in the same position as a blind student would find themselves. The `BrailleR` package generates the following description:

```
> library(BrailleR)
> VI(Graph1)
```

```
This is an untitled chart with no subtitle or caption.
The chart is comprised of 2 panels containing sub-charts, arranged horizontally.
The panels represent different values of year.
Each sub-chart has x-axis 'displ' with labels 2, 3, 4, 5, 6 and 7.
Each sub-chart has y-axis 'hwy' with labels 10, 20, 30 and 40.
Each sub-chart has 2 layers.
Panel 1 represents data for year = 1999.
Layer 1 of panel 1 is a set of 117 points.
Layer 2 of panel 1 is a smoothed curve using method 'lm' with confidence intervals.
Panel 2 represents data for year = 2008.
Layer 1 of panel 2 is a set of 117 points.
Layer 2 of panel 2 is a smoothed curve using method 'lm' with confidence intervals.
```

What we learn from `BrailleR` is that the code has actually generated two graphs, not just one, and that each of them has two "layers" of information. It is clear from the descriptions that a blind student could now know what the graph contains, but still does not get the benefits of the interpretation of its content. The `BrailleR` package is therefore likely to remain "under development" as efforts are made to catch up with the latest options being given to sighted R users.

The increasing use of markdown in conjunction with R is a huge leap forward in access to statistical analysis of data. Markdown files are plain text files and are processed into (very accessible) HTML, epub3 (also very accessible), or many other (less accessible) file formats (including the pdf version of this paper in the workshop proceedings). The use of R and markdown (Allaire et al. 2020) avoids the pain of copying and pasting text output and graphs, which is comparatively difficult for many blind people. This paper was written using R markdown in preference to LaTeX or MS Word. Reading the HTML version found at https://R-Resources.massey.ac.nz/papers/DEIMS21Access/ will show that the standard heading styles are included; easily read mathematics and alt tags for graphics are also features of R markdown documents. These are done as the default action so little to no user knowledge of accessibility is necessary. Given so many authors are now using R markdown and other associated tools as their normal way of working, the range of accessible statistics texts that demonstrate use of R is broadening. Many of these texts include a web-based search engine which makes the subject index somewhat redundant.

3.1 Software selection has an impact on the blind student

Many subject areas will choose a particular statistical software solution because it does everything that is wanted within a course. This in turn has an impact on what is taught within a course too. Some statistical software will do some tasks much better than others, but the use of

R has increased across science disciplines over the last ten years. It is much easier to administer a course when all students are using the same application. Many courses that use R must now contend with a plethora of operating systems and even use of virtual machines.

The last ten years has seen little change in the accessibility of statistical software. While personal preference plays a role in making this proclamation, I still believe R is the best option for any blind person, with only two other options offering anything close to what I consider access.

Ten years of attempting to gain any traction with statistical software developers has been almost fruitless. The advances in R are courtesy of a large and extremely active open source community. The only other options that offer the blind user any hope of equivalent access as enjoyed by their sighted peers remains SAS and Stata. Rather unfortunately, neither of these options has much traction in the teaching of undergraduates in any discipline nowadays, and only SAS can claim to take access for blind users seriously.

Obviously students should not be restricted in their course selections on the basis of the software used in that course, and allowances must be made so that a blind student can use alternative software that is accessible. If the student cannot access the chosen software for a course then the onus is on the course staff to ensure that whichever application they choose as a substitute can fulfil all the tasks of the course's preferred software. R will meet that need.

Godfrey (2009) (correctly) prophesized, "Most commercial software is produced in the United States. Given the U.S. has anti-discrimination legislation and is a comparatively litigious society, it is surprising that software developers continue to disregard the needs of blind users. In the author's opinion, these laws should be used to protect the interests of blind people and their access to software but it is difficult to see that this will happen. It is certainly not a short to medium term solution."

3.2 Supporting resources

It seems pointless to have accessible support material for inaccessible software, or inaccessible support material for accessible software. In general though, the documentation required for R is available in completely accessible HTML. Legacy documentation is reducing in value as the R ecosystem is evolving.

This is not an R specific phenomenon; SAS also makes its support material available in HTML. It is now common to find documentation for many software solutions across STEM disciplines in HTML. The standard adaptive software used by most blind people is extremely capable of handling HTML content so blind people are better off today than we have ever been.

Blind students and staff must be ready to adapt if the proposed course does not use one of the few accessible statistical software options. It is fortunate that R as the best option has a burgeoning community of blind users who are willing to share their experiences. The quantity and quality of the massive amount of publicly available accessible documentation presents a potential hazard though; novice users must somehow determine which material will assist and what is a distraction.

4 Element 3: Tables

Godfrey (2009) described scientific calculators used by sighted students as a common alternative to the old-fashioned statistical tables still in use. Times have changed and even use of those calculators in statistics courses is decreasing in popularity nowadays.

Godfrey (2009) stated “The student who does not have ready access to the simple tools will become comparatively inefficient.” Even though what is considered “simple” may have changed, this message remains sound. To be efficient and effective, a blind student may need to put extra time into the tools that are going to prove the simplest for them to manage; this decision could well be different to what a sighted student would choose if no advice was sought. My advice to sighted and blind students is practically the same now.

Godfrey (2009) also stated “Be ready to use tables before the course starts. It is unlikely that a software solution will be appropriate for use in assessment exercises such as tests or exams without prior negotiation.” Prior negotiation might be required, but the era of “reasonable accommodation” is upon us in so many situations in life, including education. I would now argue that making use of R or other suitable software (perhaps under supervision or with constraints) should be seen as a reasonable accommodation for a blind student even in circumstances when other students must do manual calculations.

5 Element 4: Mathematical Formulae

The situation promoted in Godfrey (2009) “Unfortunately there is no escaping horrible formulae, although the reliance on doing all the substitutions and integral calculus that was part and parcel of the introductory course in the 20th century is reducing. There is no way of avoiding the various symbols that arise in a statistics course. The intending statistics student needs to know how they will present work in a format the lecturer can handle.” The evolution of statistics courses mentioned previously has reduced, but not eliminated, the mathematical load on students.

Given the relative leaps forward in access to mathematical content in HTML via tools including Mathjax in particular, and the incredible increase in available accessible mathematical content, blind people working with mathematical content are substantially better off today than we have ever been before.

Accessibility of mathematical software is not a direct focus of this article, but options do exist should the statistical software not provide a solution for the more mathematical coursework at higher levels. This would include the need to perform analytic integration and differentiation for example, neither of which is a common tool for statistical software.

While authoring mathematical content is now relatively easy for a blind person, it does require effort. Use of standard LaTeX for mathematical expressions makes creation of documents in Microsoft Word (MathType is required) or markdown within reach of most students (blind or sighted), so the need to recommend that a blind student entering university should learn everything about LaTeX can now be set aside. One major criticism of LaTeX was always that it generated an inaccessible document so the blind author could not independently verify the outcome; this has been worked on, but the access is still below that offered by Microsoft Word or HTML generated from markdown. As an illustration, this paper was prepared in markdown so that it could be read by the author in HTML and submitted for publication in the organisers’ preferred format, because that is the way I do almost all of my work . Processing of a single file into multiple formats has vastly improved my ability to create documents independently.

6 It’s far from bad news

Aside from the fact that it is possible to avoid or meet the challenges of the four key elements as described above, there are reasons why blind students can succeed where their sighted peers may not. Statistics is all about the ability to predict the future with knowledge of the past; to apply probability; to deal with randomness and understand variability to name just a few points.

Blind people live life encountering things unexpectedly, even if they have some vague idea of where something should be (estimation) and there is always some amount of inaccuracy (error).

Part of any successful teacher-student relationship is the ability to connect the theory in a course to real life examples that resonate with the students. The experiences of the blind student may not match those of their sighted peers. In the author's experiences, this mismatch exists in many different ways. For example, international students do not understand any examples based on events in New Zealand's history without some serious explanation, but all students relate to world news events. Finding examples that match a blind student's experiences will be difficult for the lecturer who has little knowledge of the blind person's world. This can be overcome through developing rapport, but this requires some proactive effort by the student.

I firmly believe that some statistical ideas are best learnt from one's blind peers. Some notions are so easy to frame up for blind students if only the right scenario can be found to help its explanation. Generally speaking, blind people have been able to form networks where they can pick up information from one another. This is especially true for assistance with our specific software just to name one example. In my opinion, those of us at the forefront of new endeavours by blind people must expect to support those that come along behind us. Thankfully, many blind people do this all the time.

7 Statistical software: The litany of failure

The inaccessibility of software has been recorded in many ways, including Godfrey and Loots (2014) and Godfrey (2013), but perhaps the personal commitment to attend conferences for the R development community and to make direct contact with developers has increased the general knowledge of the potential for R to be a viable solution for blind people. Godfrey and Loots (2014) compared four commonly used statistical software options; R came out on top. Although R as the base product was considered superior, Godfrey (2013) lamented the inaccessibility of the RStudio integrated development environment (IDE) which was becoming increasingly popular in the R user community.

Several statistical software companies needed to be contacted to make the best attempts to ensure fairness and accuracy in Godfrey and Loots (2014). Remarkably little has happened to change the situation between 2015 and 2020. In early 2020 though, RStudio, the developer of the previously inaccessible IDE of the same name, announced it was going to put effort into improving the IDE's accessibility. This has proven to be welcome news to blind R users who want to work in the same fashion as their classmates and work colleagues. Initial work has shown great promise, in part due to the hiring of a blind person to take a key role in the project.

8 Role models do exist!

In 2009, I had no idea that I was the only totally blind person to gain full time employment as a Lecturer in statistics. I was an undergraduate in a pre-internet era; the arrival of the internet did not lead to my searching for others like me, until I stumbled upon the email lists promoted by the National Federation of the Blind for various interest groups, one of which was the Blindmath list. I found that I was soon contributing as much as I was receiving from joining that community. I believe my engagement in that community led to my attendance at WEIMS in 2009 and sharing my experiences in Godfrey (2009).

I was approached in 2010 by another blind person who wanted to get a job as a Lecturer in Statistics. He had gone looking for anyone to talk to and discovered that I was the only person to reach out to; this was quite a surprise to me. I always assumed that someone else had already been there and done that before me; I assumed that I just couldn't find out who the others were, not that there were no others to find. The sharing of ideas with someone who could verify my ideas, challenge some of them, and complement others, was hugely beneficial in a professional sense and culminated in two major publications based on our experiences with statistical software Godfrey and Loots (2014) and as students and lecturers Godfrey and Loots (2015).

Over the years, I've kept in touch with a handful of blind people as they complete a PhD in Statistics, some blind people wanting to start an academic career in another STEM discipline or something closely related, and some others who forged their STEM careers before or at about the same time as I did. All of these people now form a community of role models and I no longer feel any pressure to be the only person proving that blind people can become statisticians. I'm proud that blind people working with statistical data are no longer a novelty.

As has now been demonstrated over and over again, there is potential for blind people to gain employment that makes use of their ability to analyse data. While some blind people have forged careers in academic institutions across a wide range of disciplines, others have found success in private industry.

9 Concluding remarks

Ten years ago, I discovered that I had a role to play in helping improve access to statistics education for blind people. Over those years, I've learned a lot from experimentation, trying to show blind people what they can learn, and seeing a lot of good ideas fall flat because they made work for the blind person or the staff that supported them.

In the 20th century, access to printed information was the primary limitation on the blind student. This is a considerably smaller problem for blind students undertaking statistics courses in 2021 than it was even ten years ago. In the author's opinion, it is the modern ways of teaching statistics that are driving the inaccessibility in the 21st century. The ability of the blind student to take courses in qualitative disciplines and succeed is increasing with the improvements in technology. In many sciences, the opposite is true as we lose ground due to advancements in software and other technologies that are not created with the blind user in mind.

It is possible for blind students to succeed in statistics courses, as long as the right choices in software and the way work will be undertaken are made. We live in information-rich times, and the most current work being done in statistics that is relevant to introductory and advanced students is being presented in a very accessible format. Reliance on obtaining information from the inaccessible pdf format, even with the best enhancements available today, is an inferior way of working and leaves blind students at risk of suffering in a similar fashion to many others who preceded them. It will still take some serious preparation before the course starts, some additional work during the course, and some reasonable accommodations being made by the course staff. As much as the blind student is afraid of failing statistics, most academic staff are afraid of failing capable students.

It was necessary to prove that some things could be done, even if those actions were not eventually taken up by the thousands of blind students around the world. Numerous glass ceilings have been broken and some hugely innovative tools have been developed. The most enduring successes are those tools that provide improved access to blind people and have become embedded into the everyday software and practices being used by everyone. In statistics, this includes improvements in R and SAS, and in mathematics this includes MathJax. I put the availability of R markdown at the forefront of accessible tools that has made such a difference for R users, with both the consumption of printed information and the production of statistical analyses in HTML being key to this success.

With respect to other printed information, we need to be sure that research efforts get put into the tools that are the most practical and useful for blind people, instead of those tools that are the most commonly used by publishers. In my opinion, the efforts to improve access to content in pdf are important. The problem is that unless they become built-in so that they are not optional extras that require authors and publishers to learn how to build in access, they will not gain sufficient traction and therefore not be valued by blind consumers. Whether we like it or not, younger blind people do not want to read material in a fixed page format that is second rate when there is slightly better HTML based solutions ready to hand. They want access on a plethora of devices, running all operating systems, and in my experience, they want them to work right out of the box. It is difficult to see what (if any) benefits a blind person will enjoy from the most accessible pdf that do not already exist when working with documents available in accessible HTML.

The next ten years is going to be very exciting for blind people in STEM. We need to make sure the tools that currently work well continue to do so, and we also need to continue to develop more tools that improve the ability of blind people to succeed in more STEM fields. Access to information is a necessary problem to overcome, but access to modern tools will also continue to pose challenges.

References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020. *Rmarkdown: Dynamic Documents for R*. `https://github.com/rstudio/rmarkdown`.

Godfrey, A. Jonathan R. 2009. "Are Statistics Courses Accessible?" In *Proceedings of the Workshop on E-Inclusion in Mathematics and Science 2009*, 72–80. Fukuoka, Japan.

———. 2012. "The BrailleR Project." In *Proceedings of Digitization and E-Inclusion in Mathematics and Science*, edited by Katsuhito Yamaguchi and Masakazu Suzuki, 89–95. Tokyo, Japan. `http://workshop.sciaccess.net/DEIMS2012/Proceedings.zip`.

———. 2013. "Statistical Software from a Blind Person's Perspective: R Is the Best, but We Can Make It Better." *The R Journal* 5 (1):73–79. `http://journal.r-project.org/archive/2013-1/godfrey.pdf`.

Godfrey, A. Jonathan R., and M. Theodor Loots. 2014. "Statistical Software (R, SAS, SPSS, and Minitab) for Blind Students and Practitioners." *Journal of Statistical Software, Software Reviews* 58 (1):1–25. `http://www.jstatsoft.org/v58/s01`.

———. 2015. "Advice from Blind Teachers on How to Teach Statistics to Blind Students." *Journal of Statistics Education* 23 (3):1–28. `http://www.amstat.org/publications/jse/v23n3/godfrey.pdf`.

Godfrey, A. Jonathan R., and Paul Murrell. 2016. "Statistical Graphs Made Tactile." In *Proceedings of the International Workshop on Digitization and E-Inclusion in Mathematics and Science 2016*, edited by Katsuhito Yamaguchi and Masakazu Suzuki, 69–74. Kanegawa, Japan. `http://workshop.sciaccess.net/DEIMS2016/index.html`.

Godfrey, A. Jonathan R., Paul Murrell, and Volker Sorge. 2018. "An Accessible Interaction Model for Data Visualisation in Statistics." In *Computers Helping People with Special Needs - 16th International Conference, ICCHP 2018, Linz, Austria, July 11-13, 2018, Proceedings, Part I*, 590–97. https://doi.org/10.1007/978-3-319-94277-3/_92.

Godfrey, A. Jonathan R., Debra Warren, Paul Murrell, Timothy Bilton, and Volker Sorge. 2020. *BrailleR: Improved Access for Blind Users*. https://github.com/ajrgodfrey/BrailleR.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Realization of Inclusive Environment in Online STEM Lectures Using PowerPoint

Toshihiko KOMADA

Junior College Funabashi Campus, Nihon University, Japan
komada.toshihiko@nihon-u.ac.jp

Abstract

Many STEM educational materials for online lectures are provided as video, PDF, and other electronic format via the Internet network. In this situation, there is a difficulty for visually impaired students to access scientific element of STEM educational materials. I have developed PowerPoint add-on's to add speech output for such elements into the PowerPoint materials. By using the tool for STEM educational materials created with PowerPoint, visually impaired students become able to access the materials with a slight burden of manual processing by faculty staff. This tool realizes more inclusive environment in online STEM lectures.

1 Introduction

Due to the pandemic of COVID-19, recently, online lectures are widely given all around the world. Many of the educational materials for online lectures (for sighted students) are created with PowerPoint and provided as lecture videos in MP4 and/or PDF; otherwise, the slides are displayed on a computer screen in online real-time lectures. In case that lectures are face-to-face an in inclusive class, faculty staff can explain directly the lecture materials as necessary even if they cannot be accessed by visually impaired students. On the other hand, in case of online lectures, visually impaired students should access the entire educational content for themselves at home. It is already given how to create PowerPoint content so that visually impaired students become able to access its elements except for scientific parts such as formulas, diagrams, and tables [1, 2, 3]. However, there remains a difficulty since they can hardly access important scientific elements included in the educational materials for STEM (science, technology, engineering and mathematics), which is created with PowerPoint, with ordinary assistive tools such as a screen reader or a braille translator.

A solution to the difficulty for visually impaired students is to add audio (aloud reading) information to the materials. "ChattyInfty" [4] is software for visually impaired people to edit and to read STEM documents, which already has a function of reading out the STEM materials including math formulas with synthesized speech output. It also has a function to export a ChattyInfty document to audio-embedded EPUB3 and HTML5 (HTML5+JavaScript+MP3). “InftyReader” [4], which is OCR software for STEM documents, can convert a document in image or PDF into ChattyInfty format, multimedia DAISY and other accessible formats. They should be useful to make PowerPoint STEM contents accessible; however, the collaboration between PowerPoint and ChattyInfty/InftyReader has not yet been realized.

In this study, by linking them, I develop a tool that enables visually impaired students to access STEM educational content for sighted students created with PowerPoint.

2 Method

I implement an automatic-processing mechanism to convert text description of formulas, diagrams, and tables created by ChattyInfty into synthesized speech and to insert it into PowerPoint materials. In terms of a math part in an already-existing STEM document, you can convert it into ChattyInfty once with InftyReader OCR and insert synthesized speech with ChattyInfty into PowerPoint materials.

It is realized as a PowerPoint add-on using VBA (Visual Basic for Application). The outline of the add-on is as follows (Figure 1).

1. STEM elements are copied as image data from existing PowerPoint materials into the clipboard.
2. The image information is recognized with the math OCR software, InftyReader, and the recognition result of the text including scientific information is output to ChattyInfty format.
3. The result obtained with InftyReader is checked and edited with ChattyInfty.
4. The generated text information is converted to audio information (MP3 file) by ChattyInfty.
5. Insert this MP3 file into the PowerPoint materials. This MP3 file is linked to the IML (Infty XML) file of the recognition result.
6. If changes/corrections are necessary in the inserted audio information, open the IML file linked to the MP3 with ChattyInfty and change/correct its element according to the steps 3, 4, and 5.

Instead of existing PowerPoint materials, in the step 1, it is possible to copy image data of STEM elements from existing PDF materials to the clipboard with the snapshot function in Acrobat Reader. In the step 3, instead of the steps 1 and 2, you may create a content by manually inputting mathematical expressions or texts to a new ChattyInfty document, or by importing them from existing ChattyInfty/TeX documents into the new ChattyInfty document.

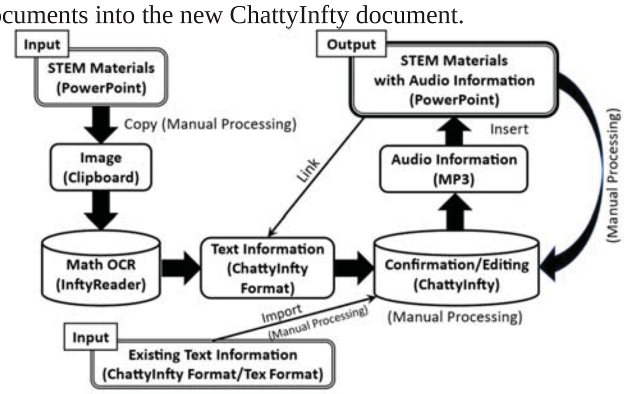


Fig. 1. An outline of an automatic-processing mechanism of a PowerPoint add-on using VBA. Converting STEM documents into audio information and inserting it into PowerPoint materials.

By making use of this mechanism, visually impaired students become able to access STEM elements in PowerPoint slides and MP4 video generated from them.

Incidentally, a similar mechanism, inserting "formula" from existing materials into MS-Word by making use of InftyReader OCR has already been realized as a MS-Word add-on with VBA [4].

3 Conclusion and Future Plans

The tool given here allows visually impaired students to access STEM content for online lectures created with PowerPoint, which is usually difficult for visually impaired students to access.

Future plans are as follows.

1) Realize further automatic processing of the tool: Currently, we need to select manually each math part to add aloud-reading for them. It requires a certain effort to prepare for the contents. It should be improved so that the process is done automatically.

2) Make the tool multilingual: The tool is developed for Japanese in first stage. The tool is implemented by making use of ChattyInfty/InftyReader as PowerPoint VBA. Some other local languages such as English, Italian and Vietnamese are available in ChattyInfty/InftyReader [5], and it is possible to localize the tool by writing down VBA in each language.

3) Seamless transformation between the PowerPoint format and the ChattyInfty format: If this function is realized, it is possible to export a content produced with PowerPoint to various accessible format such as ChattyInfty, mp4, audio-embedded EPUB3 and audio-embedded HTML5. A content produced with ChattyInfty can be also converted into accessible PowerPoint directly. It is efficient way to provide educational materials in STEM for inclusive education.

References

[1] Accessible PowerPoint template sampler, Microsoft, https://go.microsoft.com/fwlink/?linkid=851258

[2] Make your PowerPoint presentations accessible to people with disabilities, Microsoft, https://go.microsoft.com/fwlink/?linkid=851258

[3] PowerPoint Accessibility, CSUN (Universal Design Center) , https://www.csun.edu/universaldesign-center/PowerPoint

[4] Science Accessibility Net, https://www.sciaccess.net/en/index.html

[5] Katsuhito Yamaguchi and Masakazu Suzuki, "An accessible STEM editor customizable for various local languages", Journal of Enabling Technologies, Emerald Publishing Ltd., 13, No.4 (2019) pp. 240-250.

Dictation of mathematics in natural Italian language

Dragan Ahmetovic[2], Tiziana Armano[1], Cristian Bernareggi[1], Anna Capietto[1], Sandro Coriasco[1], Mattia Ducci[1], Alessandro Mazzei[3], and Nadir Murru[1]

[1]Università degli Studi di Torino, Department of Mathematics G. Peano, Italy
[2]Università degli Studi di Milano, Department of Computer Science, Italy
[3]Università degli Studi di Torino, Department of Computer Science, Italy

Abstract

System based on speech interaction are becoming widespread in many application fields. Nonetheless, they have not yet become widely used in applications for STEM. This is mostly due to the difficulties in recognizing mathematical notation in natural language. This works investigates dictation of mathematics in natural Italian language. It draws conclusions about some characteristics of the language that have to be taken into account in the design of a system that enables dictation of mathematics. In the end, an extension of such a prototype system, based on the conclusions of this study, is illustrated.

Contents

1 Introduction 1

2 Related Work 2

3 Analysis of the Italian language for the dictation of mathematics 3

4 Results 3

5 Discussion 5

6 SpeechMatE 6

7 Consclusions and Future Works 7

1 Introduction

The last decade has witnessed significant advances in speech recognition, which have paved the way for the development of applications operable through speech commands including personal and home assistants based on dialogue interfaces, speech-driven interaction with mainstream operating systems (e.g. MS Cortana in MS Windows) and speech-to-text for word processing. Indeed, a large population can benefit from systems operated by speech. On one side, everyone can get advantage of speech interaction in activities where hands are not free. On the other side, benefits brought by speech interaction are of paramount importance for millions of people who have severe motor disabilities (*e.g.* tetraplegia), temporary or permanent motor impairments (*e.g.* tendinitis, arthritis, muscoloskeletal diseases), motor disorders (*e.g.* developmental coordination, hand disorders, stereotyped movements), upper limb disabilities and amputations [12].

Nonetheless, so far, speech interaction has not yet become common practice in educational applications, especially in those concerning STEM subjects. Many different reasons, related to the dictation of mathematical notations, account for this. First, mathematical notations have a visual nature and they are represented in a bi-dimensional layout [11]. By contrast, speech flow is sequential. Hence, for an expression to be dictated, a reader has to mentally convert the notation from a bi-dimensional representation to a sequential one according to often unnatural reading rules (e.g. by using many pauses for grouping elements) [7]. Second, recent speech recognizers are highly accurate, but the recognition results are not 100% error-free. Whilst a low error rate is acceptable in the dictation of text, it significantly hampers the dictation of mathematical expressions, for which, even slight variants of symbols can totally change the semantics [1]. Third, reading of mathematics in natural language is very ambiguous. Indeed, many symbols are not expressed or they are tacitly identified by pauses in the speech flow [6, 14]. Fourth, doing maths exercises implies transforming maths expressions by navigating symbols back and forth and by editing subexpressions. This process turns out to be cumbersome through speech commands only [1, 2, 8]. Fifth, even though mathematical notation can be considered universal, when it comes to dictate maths expressions, the characteristics of each language and of each speaker play a crucial role. Nowadays, the systems which enable dictation of maths expressions are designed to strike a balance between dictation of mathematics in natural language and recognizing the utterances unambiguously.

This work investigates the characteristics of the dictation of mathematical expressions in the Italian language. It is a preparatory study for the design and development of a system that enables the dictation and editing of mathematics in the Italian language. A prototype system for dictation of mathematics in Italian, which is being developed in our working group, has been extended taking into account the results emerged.

2 Related Work

This section introduces the systems which have been developed to dictate mathematics. TalkMaths [14, 15] is a prototype application which transforms arithmetic, algebraic and trigonometric expressions from spoken English into LaTeX or MathML. Speech recognition is achieved through Dragon Naturally Speaking[1] (DNS) and the editing process is assisted by an intelligent interface based on a statistical model [1]. This system recognizes the English language only and the dictation is based on pauses, which slow down the dictation process [7].

Mathifier [3] is an open source software module that converts maths expressions from English into LaTeX. It includes a dictionary, a language model and an acoustic model to recognize mathematical English utterances. Speech is recognized through Sphinx-4 [13]. This system has been designed for recognizing maths only in English, and so far Sphinx does not recognize the Italian language reliably.

CamMath [7] is a prototype software module designed to prove the benefits of continuous speech over discrete utterance of mathematical expressions in English. It is a front-end for Scientific Notebook[2], and it uses Microsoft speech recognition platform. This system does not recognizes mathematics in Italian.

Google EquatIO[3] is a service for the dictation of very simple maths expressions in English in MS Word and in GSuite applications.

Metroplex MathTalk[4] is a commercial application that enables speech input of arithmetic, algebra, calculus and statistics in English. Recognition is performed by DNS and it implements a dictation model based on pauses, which slow down the dictation process.

1 www.nuance.com
2 www.mackichan.com/
3 https://www.texthelp.com/en-us/products/equatio/
4 www.metroplexvoice.com/

Bernareggi et al. [5] studied the dictation of elementary mathematics in Italian in the LAMBDA system [4], a commercial maths editor used by visually impaired people in Italy, through DNS. The dictation rules are the ones adopted in the LAMBDA system, which adopt special symbols close Braille code. This is far from the Italian natural language.

To the best of our knowledge, a reliable system for the dictation of mathematics in natural Italian language has not yet been designed and developed.

3 Analysis of the Italian language for the dictation of mathematics

The analysis of the dictation of mathematics in Italian is conducted by going through four main stages. First, a list of mathematical expressions is compiled according to three criteria: (i) it includes expressions which cover a secondary school program of mathematics and geometry in Italy; (ii) the complexity of each expression, defined as the number of symbols and nested subexpressions, mirrors the one of maths expressions presented in Italian secondary school books; (iii) in view of future comparisons to English, the expressions are similar or identical to those presented in the Handbook of Spoken Mathematics [6]. Selection of the expressions is driven by the teaching experience of a secondary school maths teacher and of the working group members who have a long-term experience in teaching mathematicss (e.g. university maths professors and researchers). The complexity of the expressions is determined according to the complexity of the expressions randomly chosen for each topic from two secondary school exercise books [10, 9]. The resulting list consists of one hundred and ten expressions organised in six categories: **set theory**, **logic**, **arithmetic and algebra**, **trigonometry algorithms exponentials**, **calculus**, and **geometry**. This list contains forty-eight expressions identical to those presented in the Handbook of Spoken Mathematics and ninety different maths symbols which are used in Italian secondary schools.

Second, ten volunteer readers are recruited in order to read out loud the expressions in the list and record them in an audio file. This group of volunteers includes eight persons with a degree in mathematics and two with a degree in engineering; eight of them are teachers and two of them have teaching experiences. These readers are required to read each expression from the list in natural language likewise a teacher reads mathematics in a classroom.

Third, in order to generate a corpus of maths expressions which can be analysed, audio recordings (about 6 hours) are listened, transcribed into plain text and organised in a CSV document. An excerpt of the resulting corpus is reported.

Fourth, the analysis of this corpus is conducted. On one side, in order to have a feel about the linguistic syntactic complexity of the sentences pronounced by the readers, we used a statistical dependency syntactic parser[5], tuned on Italian. On the other side, in order to find out characteristics of the dictation which cannot emerge through automatic processing, all the transcribed expressions were read and examined by the members of the working group.

4 Results

By processing a small number of the simplest sentences (e.g. $2 \leq a \wedge a \geq b$ read as "due minore uguale a a e a maggiore uguale di b") with the syntactic parser, we found that a typical error is the misunderstanding of the correct part of speech for the symbols in the expression. Indeed,

[5] http://lindat.mff.cuni.cz/services/udpipe/

Expr	$A = \{a \mid a \in \mathbb{N} \cup \{0, -1, -2\}\}$	$a[b + c - e(f - g)]$	$\frac{a+b}{d}$
P1	definiamo l'insieme a grande uguale aperta graffa l'elemento a piccolo tale che a piccolo appartiene a n insieme dei numeri naturali unito all'insieme formato dagli elementi 0 meno 1 e meno 2 chiudiamo la graffa e chiudiamo anche la graffa grande	a per aperta quadra b più c meno e che moltiplica aperta tonda f meno g chiusa tonda chiusa quadra	frazione al numeratore a più b al denominatore d
P5	insieme a maiuscolo uguale a apro graffa a tale che a appartiene a n unito all'insieme apro graffa 0 virgola meno 1 virgola meno 2 chiusa graffa chiusa graffa	a per aperta quadra b più c meno e per aperta tonda f meno g chiusa tonda chiusa quadra	a più b fratto d

Table 1: Two sample records of the corpus. The first row shows three expressions that are dictated by the volunteer participants. the second and the third row report the transcription of the audio recording for two participants to this tudy.

the parser analyses some symbols, (e.g. "a") as a function word of Italian (preposition for "a") . Moreover, by listening to the audio recording of these expressions, we observed that the prepositions "a" and "di" are pronounced with the same sound of the following letter. Based on these considerations we identify a first group of ambiguities: **prepositions followed by a letter**. These can be hardly recognized by a speech recognizer and they are not identified by a syntactic parser for the Italian language.

The analysis conducted by examining the characteristics of the expressions let single out three more different types of ambiguities: **tacit symbols** (e.g. parentheses in $f(x)$ read as "f di x"; the subscript in a_0 read as "a zero"); **tacit upper case modifiers** (e.g. upper case letters are not pronounced in $A = \{a \mid a \in \mathbb{R}\}$ read as "a tale che a appartiene r") and **pauses or changes in reading speed** which are used to identify delimiters (e.g. parentheses, the end of an exponent, end of the argument of a function). Each expression in the corpus was annotated with the number and the type of ambiguity. For example, the expression $a = a(a + 1) - 2$, read as "a uguale a a pause a+1 pause meno 2" contains one ambiguity concerning the preposition "a" and one related to the use of pauses which implicitly identify the parentheses. Based on these annotations, the distribution of the ambiguities in each category is analysed. For each expression e, for each ambiguity a let r_{ea} be: $r_{ea} = \frac{n_{ea}}{w}$ where n_{ea} is the number of ambiguities of type a in the expression e and w is the number of uttered words. So, n_{ea} is the number of ambiguities of type a in the expression e normalized with respect to the number of uttered words. For each category c, for each type of ambiguity a, let r_{ac} be: $r_{ac} = \frac{\sum_e = 1^{n_c} r_{ea}}{n_c}$ where n_c is the number of expressions in the category c. So, r_{ac} is the number of ambiguities of type a in the category c normalized with respect to the number of the expressions in the category c. By using r_{ea} and r_{ac} it is possible to compare how ambiguities are distributed in different categories. The following figure illustrates the distribution of ambiguities.

Two more characteristics emerged from the analysis of the corpus: use of articles and synonyms. Both for articles and synonyms no general rule can be stated. Articles are used by all the readers. They are sometimes used before function names (e.g. $\cos(x)$ read as "il coseno di x"), before set names (e.g. $A \cup B$ read as "l'insieme A unito l'insieme b"), before roots or fractions (e.g. $\sqrt{x}$ read as "la radice di x", $\frac{1}{n}$ read as "la frazione uno su n"), or they may be missing (e.g. $\ln(x)$ read as "logaritmo naturale di x"). For what concerns the use of synonyms (i.e. equivalent forms to express the same symbol), a reader uses different synonyms both in the same expression and in different expressions. Analogously, two different readers may dictate

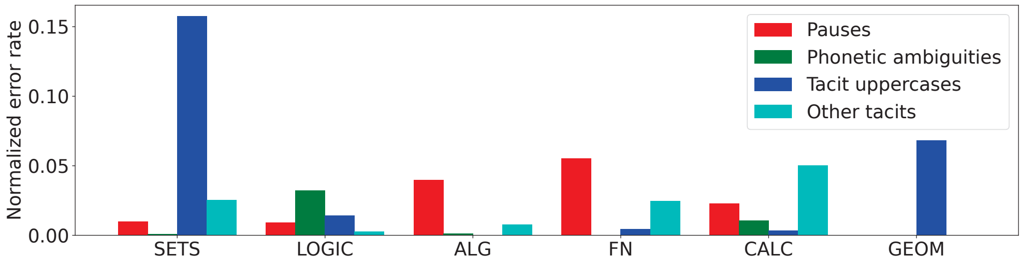


Figure 1: Bar chart of ambiguities in each category. The red bar represents pauses, the green bar the ambiguities due to prepositions bollowed by a letter, the blue bar the tacit upper case modifier and the light blue bar represents tacit symbols.

(	)	$a \cdot b$
aperta parentesi	chiusa parentesi	a per b
aperta tonda	chiusa tonda	a che moltiplica b
tonda	chiusa tonda	a moltiplicato b
apro tonda	chiudo tonda	a prodotto b

Table 2: Synonyms for symbols (in the first row).

the same symbol the same way or with different synonyms. Hence, in order to catalogue all the synonyms used by the readers, for each of the ninety symbols, all the synonyms are identified and grouped in a table. For example, the following table reports the synonyms for open and close round bracket, and for the product.

5 Discussion

The analysis conducted highlights some characteristics of the Italian language which have to be taken into account in the design of a system that aims at recognizing dictation of mathematics in a natural form. First, a preposition followed by a letter cannot be easily recognized by a speech recognizer (e.g. “a=a+1” can be read as “a uguale a a”) and, even if the sentence is properly recognized, when the proposition is omitted, the sentence can be ambiguous (e.g. “a minore di meno 1” means $a < -1$ if the proposition is omitted or $a < d - 1$ if it is not omitted in the dictation). Hence, either a guideline for dictation is given to the reader or the speech recognizer and the parser should implement techniques to disambiguate. Second, as shown in Figure 1, ambiguities due to tacit upper case modifier and tacit symbols are more present in some categories (e.g. tacit upper case in sets and geometry, tacit symbols in trigonometry, exponentials and calculus). Hence, a parser can apply context analysis to suggest what a reader means with an ambiguous phrase. Third, the problem of pauses or changes of reading speed affects especially reading in algebra, trigonometry, logarithms and exponentials. This can be tackled either by introducing a stop word that delimits the blocks read between pauses or at a different speed or by analysing pauses in the speech recognition process. Fourth, in order to ensure as natural as possible dictation, articles and synonyms must be allowed in the dictation process and properly processed by a parser.

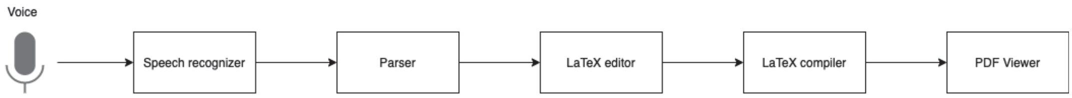


Figure 2: A high level representation of the components of SpeechMatE

6 SpeechMatE

SpeechMatE is a prototype system under development by our working group (a work on an early prototype is under submission). The ultimate goal of SpeechMatE is to enable hands-free dictation and editing of mathematical expressions through speech interaction in the Italian language. At this stage of development, SpeechMatE consists of five components: a *speech recognizer*, a *parser*, a *LaTeX editor*, a *LaTeX compiler*, and a *PDF viewer*. The user can dictate mathematical expressions and operate through speech commands. Speech flow is processed by a speech recognizer (Google Cloud Service[6]), which converts speech into text in Italian. A parser processes this text and it generates either the corresponding LaTeX expressions or it triggers a script when a command is recognized. The LaTeX expression is input in a LaTeX editor and viewed in a PDF viewer.

The analysis of dictation of mathematics in Italian has driven an extension of SpeechMatE with respect to the language recognized. In addition to arithmetic and algebra, SpeechMatE has been extended with trigonometry, calculus and sets. Mathematical expressions that can be dictated are defined as sequences of:

Atomic symbols including for example Latin letters, Greek letters, numbers, parentheses;

Operators that are applied to one or more mathematical expressions which may be separated by reserved words and ended by the stop word *fine* (i.e. *stop*).

The stop word **fine** is used to overcome the ambiguities caused by pauses or changes in dictation speed. For example, this stop word is used to delimit the end of of a function argument, of an exponent, of a root. Being aware of the ambiguities generated by the use of prepositions, maths expressions in SpeechMatE must be dictated without prepositions. The problem of tacit upper case letters is particularly relevant in some contexts (e.g. in set theory and geometry) where many upper case letters are dictated in one expression. Hence it is chosen to enable dictation in upper case letters by setting an upper case mode by a speech command (i.e. *maiuscole*). At the present stage of development, the problem of tacit symbols has not yet been tackled. It can be studied as a future work by extending the corpus and investigating a prediction model. For what concerns articles and synonyms, SpeechMatE aims to enable the most natural dictation. Articles are often, but not always, used by all readers, as above observed. Hence, SpeechMatE enables dictation with or without articles. As far as synonyms are concerned, SpeechMatE implements the table of synonyms drawn up after the analysis.
Some examples of expressions (in Italian) which can be dictated with SpeechMatE follow.
"due che moltiplica aperta parentesi quadra 1 più tonda tre meno 4 chiusa tonda alla seconda chiusa quadra"

$$2 \cdot [1 + (3 - 4)^2]$$

[6] cloud.google.com/speech-to-text/

"a più la frazione numeratore due più dodici denominatore 5 fine minore uguale zero"

$$a + \frac{2+12}{5} \leq 0$$

"frazione numeratore x alla 4 fine meno b denominatore frazione numeratore x più uno denominatore x meno uno fine meno b fine"

$$\frac{x^4 - b}{\frac{x+1}{x-1} - b}$$

Two videos[7] show the process of dictation in SpeechMatE.

7 Consclusions and Future Works

This work has investigated the main characteristics of dictation of mathematical expressions in the Italian language. The analysis is conducted on a corpus of 110 expressions read by 10 volunteers with teaching experience in mathematics (about 6 hours of recordings). The knowledge acquired has been applied to the design of SpeechMatE, an application for dictating and editing mathematical expressions in Italian. We aim to extend this work along three directions.

First, extension of the corpus and of the recordings. Even though this work has drawn some very informative conclusions, a larger corpus would enable a statistically significant analysis. Such analysis would pave the way for investigation on predictive methods.

Second, design of a usability test to validate the design choices for SpeechMatE both with respect to the language and to the interaction model based on speech input and visual feedback.

Third, outline of guidelines for teachers to read mathematics in Italian unambiguously. Based on the analysis presented in this work, we aim to prepare a set of guidelines for dictating mathematics unambiguously in Italian in order to facilitate teachers to aurally present maths in the classes. Students with sight impairment and dyslexia can benefit from unambiguous presentation of mathematical content.

References

[1] Dilaksha Attanayake, James Denholm-Price, Gordon Hunter, Eckhard Pflügel, and Angela Michelle Wigmore. Intelligent assistive interfaces for editing mathematics. In *Intelligent Environments (Workshops)*, pages 286–297, 2012.

[2] Dilaksha Attanayake, Eckhard Pfluegel, Gordon Hunter, and James Denholm-Price. Swims (speech-based web interface for mathematics using statistical language models): An intelligent editing assistant for mathematical text. In *2012 Eighth International Conference on Intelligent Environments*, pages 327–330. IEEE, 2012.

[3] Salim N Batlouni, Hala S Karaki, Fadi A Zaraket, and Fadi N Karameh. Mathifier—speech recognition of math equations. In *18th International Conference on Electronics, Circuits, and Systems*, pages 301–304. IEEE, 2011.

[4] Cristian Bernareggi. Non-sequential mathematical notations in the lambda system. In *ICCHP*, pages 389–395. Springer, 2010.

[5] Cristian Bernareggi and Valeria Brigatti. Writing mathematics by speech: A case study for visually impaired. In *ICCHP*, pages 879–882. Springer, 2008.

[7] http://bit.ly/3fslvF
http://bit.ly/2WCCYT0

[6] Lawrence A Chang et al. Handbook for spoken mathematics:(larry's speakeasy). 1983.

[7] Cameron Elliott and Jeff Bilmes. Computer based mathematics using continuous speech recognition. *Vocal Interaction in Assistive Technologies, Games and More*, 2007.

[8] Marina Isaac, Eckhard Pflügel, Gordon Hunter, and James Denholm-Price. Intuitive nuis for speech editing of structured content (work in progress). 2015.

[9] Graziella Barozzi Massimo Bergamini. *Matematica verde*. Zanichelli, 2014.

[10] Graziella Barozzi Massimo Bergamini. *Matematica azzurro*. Zanichelli, 2016.

[11] J. Mazur. *Enlightening Symbols: A Short History of Mathematical Notation and Its Hidden Powers*. Princeton University Press, 2016.

[12] World Health Organization et al. *World report on disability 2011*. World Health Organization, 2011.

[13] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition, 2004.

[14] Angela Wigmore, Gordon Hunter, Eckhard Pflügel, James Denholm-Price, and Vincent Binelli. Using automatic speech recognition to dictate mathematical expressions: The development of the "talkmaths" application at kingston university. *Journal of Computers in Mathematics and Science Teaching*, 28(2):177–189, 2009.

[15] Angela M Wigmore, Eckhard Pflugel, Gordon JA Hunter, James Denholm-Price, and Martin Colbert. Talkmaths better! evaluating and improving an intelligent interface for creating and editing mathematical text. In *6th International Conference on Intelligent Environments*, pages 307–310. IEEE, 2010.

New Features in SZS LaTeX Editor

Giuseppe Melfi[1], Thorsten Schwarz[1], and Rainer Stiefelhagen[1]

Karlsruhe Institute of Technology, Study Centre for the Visually Impaired
giuseppe.melfi@kit.edu

Abstract

In this paper, we discuss the reasons which led us to propose LaTeX as a mathematical notation to visually impaired students and the new features introduced after the first evaluation of the SZS-Editor. As for the prototype, the development of the new version was conducted in close relationship with students who use the editor regularly. In particular, a plug-in for the management of matrices has been introduced and the auto-completion has been considerably improved.

1 Introduction

Education in mathematics is important at all school levels, from primary level to the university. At the high school level, knowledge of math is the basis for many subjects in particular in mathematics, sciences, and engineering fields. It is well known that mathematical notation for the visually impaired students is a challenge which has not been completely solved by the current assistive technologies (ATs) yet.

There are two main aspects regarding mathematical notations that have to be considered to produce educational materials and ATs fully accessible and easy to use. The first one is the large set of different symbols used in math, that for example makes it impossible to design a symbol mapping one-to-one in Braille. The second aspect is the large use of subscripts, superscripts, and multi-line notation, while the screen readers are based on the linearization of the information.

2 State of the Art in R&D

In response to the problem experienced by people with visual impairment accessing mathematical materials, different solutions with different approaches have been developed in the last decades. Currently, there is not a global standard for mathematics notation for Braille. Each country has its own notation and in some cases also more than one. There are notations based both on six-dot Braille and eight-dot Braille.

One of these notations is the well known Nemeth Braille code [9], where contractions, indicators, meaningful white spaces, context-sensitive semantics, and other strategies are largely used. With this Braille code it is possible to represent a wide range of mathematical expressions. However, in order to master the code, it is necessary to learn an equally large set of rules. In other cases, a compact Braille math code was developed and distributed together with a complete system including plug-in, editor or reader like for the LAMBDA project [2] and the LEAN Math Notation [4]. Unfortunately, not all the visually impaired students are fluent in Braille and in any case the users have to learn a large set of new symbols, which poses an additional challenge for a student who is already involved with the tasks of the university. Furthermore, a symbolic notation specifically developed for visually impaired users can hinder inclusive education. The direct collaboration with a professor or a fellow student when using a specific Braille notation system could be difficult, particularly during a laboratory or a practice.

To avoid the use of a notation specific for blind, the use of LaTeX was proposed [3, 5, 8] as a possible solution of the mathematical notation for people with visual impairment. LaTeX is widespread, in particular in the university context, which facilitates the communication and interoperability with sighted persons. In addition, the tags of the markup language are quite easy to learn. The drawback of the use of LaTeX is that reading and writing long formulas is extremely cumbersome due to linearization. Ahmetovic et al. [1] with their third version of package Axessibility, proposed LaTeX as alternative text to solve the problem of the accessibility of formulas inside the PDF. Pepino et al. [10], and more recently Manzoor et al. [6], proposed solutions based on LaTeX to the problem of the mathematical notation for visually impaired students.

In our previous work [7], we tested the accessibility of some LaTeX editors with the ATs currently used by our students. The results, also available at this web page[1], show that a more accessible interface was needed. In addition, complex interfaces, with many elements often unnecessary for our students, disorient some users, especially if they are learning LaTeX for the first time. For this reason, we have chosen to develop our own editor, a simplified interface adapted to the students' needs. The results of the prototype test are available in the same previous work.

3 SZS-Editor for LaTeX

The SZS[2] LaTeX Editor is cross-platform and it is built to handle standard LaTeX input files. This preserves the inclusion of students who can share their files with their colleagues and vice-versa, and work on the same document in a collaborative way. The main features of the editor in its first version were the following:

Hidden LaTeX header. In a typical LaTeX document, it is possible to distinguish two different main blocks of code. In the header, the global properties of the document are described, while the body contains the actual content of the document. The header in our editor is automatically hidden to facilitate the revision and to ease the use of the editor by beginners in LaTeX. This feature allows the user to focus on the main content of the document. The header is still editable with a second window accessible from the menu.

Simplified log of compiling. The standard output of the LaTeX compiler is verbose and uncomfortable to quickly check it with a screen reader or a magnifier. The SZS-Editor summarizes the log in a few lines.

Go-to-line functionality. If an error occurs in the LaTeX compiling, the number of the line where the error occurred is present in the log compile. Opening a small interface, with a shortcut, the user can move the cursor to the line required.

Auto-completion LaTeX-code. Especially for beginners, it is easy to make mistakes while writing LaTeX code such as forgetting to close a bracket or inserting a dollar symbol to close the mathematics environment. The editor inserts these symbols automatically.

Code folding. With this feature, the user can selectively hide and display sections of the currently edited file which is particularly useful during the revision of a document. The editor can fold the code between two dollar symbols '$...$' (used in LaTeX to open and close the mathematics environment) or between two brackets '{...}'. For example, it is possible to fold the following long equation: `$y=(\frac{\sqrt{a}}{b})^{\frac{1+\omega}{z}}$` as `$EQ1$` or `$y=(\frac{ARGV1}{b})^{ARGV2}$`. This feature is also useful to copy/paste/cut some parts of

[1]SZS LaTeX Editor http://services.szs.kit.edu/szsLaTeX/

[2]SZS: German acronym for Study Centre for the Visually Impaired - Karlsruhe

the LaTeX code in the document and for learning and understanding a long formula, folding one by one every part of it.

Focus manager. During the development of the GUI, particular attention was paid to move the focus automatically through the interface to facilitate the use of the editor. For example, pressing the compile button the focus is automatically moved to the frame with the compiling log output. Furthermore, in the PDF preview window, there is a button to come back to the main window with the focus already in the text frame with the LaTeX code.

4 The new features

After the publication of the first version of the editor, several students started using the software. This allowed us to collect interesting feedback to improve the editor.

4.1 The matrix plug-in

One of the most interesting requests was for a matrix management tool, directly inside the LaTeX editor. For blind students, to navigate trough the matrix's elements, to use in the calculation, is very time consuming and very error-prone. To illustrate how the plug-in works, let's suppose that the student receives the exercises to be performed in a LaTeX file and that

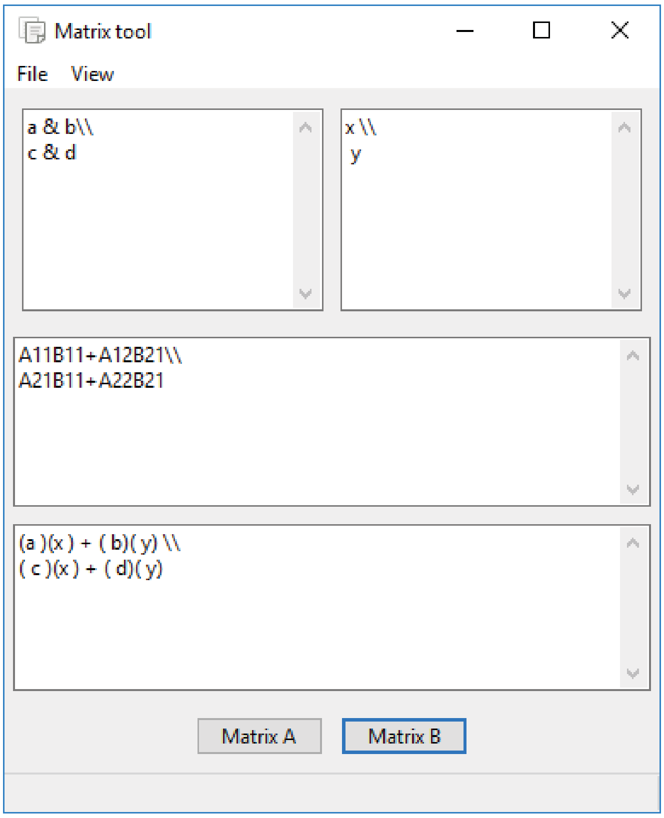


Figure 1: Matrix dialog box example

among these exercises there is the product of the A and B matrices. In our example the A matrix in LaTeX looks like:

```
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
```

and B like:

```
\begin{pmatrix}
x\\
y
\end{pmatrix}
```

To load a matrix into the plug-in, it is sufficient to select the corresponding lines of LaTeX code and press *Alt+l*. At this point, the student will open the matrix dialog box, where there are four text fields. One for matrix A, one for matrix B, one to write in LaTeX the operation to perform (in our example `A11B11+A12B21\\ A21B11+A22B21`), and the last field where the result will be shown after to have pressed *Ctrl+Enter*. The matrix dialog box is shown in Figure 1.

This plug-in is an auxiliary tool that does not do the calculation itself, and there should be no reason why this should not be used also in the exams. The plug-in also offers the possibility to consult the A and B matrices in tabular form.

4.2 Synchronization of the PDF preview

Some of our students prepare their exams in study groups with sighted fellow students. In these contexts, the editor has proven to be an effective interface that improves mathematical communication between students. The blind student can write the desired mathematical expression in LaTeX and show to the sighted student the PDF preview of the compiled LaTeX code. The PDF preview was initially designed primarily for partially sighted students, so the preview window was opened in full-screen mode. But in the context previously described, the

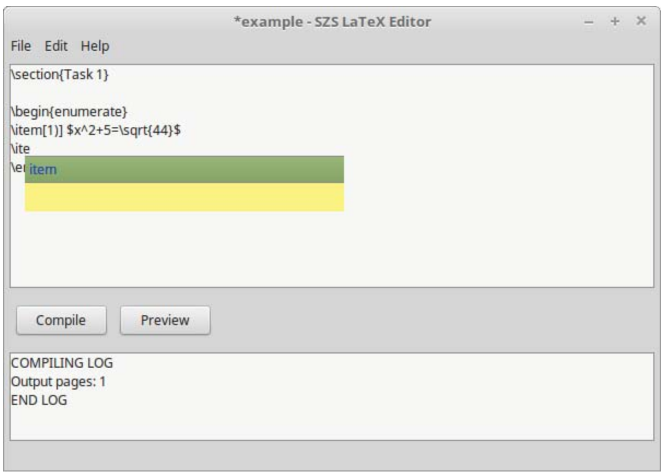


Figure 2: Suggestion list example

full-screen mode is uncomfortable. Therefore, a second layout has been implemented. It provides side-by-side windows containing the LaTeX code and the PDF preview. In addition, the synchronization of the preview with the current position of the cursor in the LaTeX code field has been added. The feature was required by both blind and partially sighted students.

4.3 Auto-completion LaTeX-code

In the first version of the editor, the automatic completion of the code was limited. In the current version, when the user starts writing a command for which one or more auto-complete suggestions are available, a short sound notifies the user of the opening of the suggestion list. At this point, the user can continue writing by ignoring the drop-down menu, or she/he can enter the list of suggestions by pressing *Tab*, select the suggestion, and complete the command pressing *Enter*. An example of a suggestion list opened typing a LaTeX command is shown in Figure 2.

This feature has been particularly appreciated by some students, especially because to add or remove a command from the suggestion list, it is sufficient to edit an XML configuration file. Interesting is the experience expressed by a student who reported having customized his configuration file to suggest whole blocks of code instead of single commands. The student uses these blocks as templates for elements like lists, matrices, and tables.

5 Conclusion

The publication of the first version of the LaTeX editor was welcomed with particular enthusiasm by some of our students. This prompted us to further develop the software, according to user requirements. We have therefore introduced a plug-in for matrix management, a second windows layout for more comfortable use in inclusive study contexts between blind and sighted students, the synchronization of the PDF preview, and the improvement of the auto-completion feature. We aim to introduce new features in co-design with blind students, plan to evaluate the new version, and collect the feedback of users. We will focus in particular on the difficulties they encounter and on the cognitive load of working with LaTeX.

References

[1] Ahmetovic, D., Armano, T., Bernareggi, C., Capietto, A., Coriasco, S., Doubrov, B., ... & Murru, N. (2020, September). Automatic Tagging of Formulae in PDF Documents and Assistive Technologies for Visually Impaired People: The LaTeX Package axessibility 3.0. In ICCHP (p. 69).

[2] Alistair, E., McCartney, H., Fogarolo, F.: Lambda: A multimodal approach to making mathematics accessible to blind students. In: 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 48-54. (2006)

[3] Borsero, M., Murru N., Ruighi, A.: LaTeX as solution to the problem of access to texts with formulas by visually impaired people. In: GuIT Meeting 2016, ArsTeXnica. 22, pp. 12-18 (2016). http://www.integr-abile.unito.it/articoli/LaTeX-acc.pdf

[4] Gardner, J., Christensen, C.: More Accessible Math, the LEAN Math Notation. In: K. Miesenberger et al. (Eds.): ICCHP 2012, Part I, LNCS, Vol. 7382, pp. 124-129. Springer, Heidelberg (2012)

[5] Gonzúrová, W., Hrabák, P.: Blind Friendly LaTeX. In: Miesenberger K., Karshmer A., Penaz P., Zagler W. (eds.) Computers Helping People with Special Needs. ICCHP 2012, LNCS, vol 7382, pp. 138–141. Springer, Heidelberg (2012)

[6] Manzoor, A., Arooj, S., Zulfiqar, S., Parvez, M., Shahid, S., Karim, A.: ALAP: Accessible LaTeX Based Mathematical Document Authoring and Presentation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, pages 504:1–504:12, New York, NY, USA, 2019. ACM.

[7] Melfi, G., Schwarz, T., Stiefelhagen, R.: An Inclusive and Accessible LaTeX Editor. In: Miesenberger K., Kouroupetroglou G. (eds) Computers Helping People with Special Needs. ICCHP 2018. Lecture Notes in Computer Science, vol 10896. Springer, Cham (2018) https://doi.org/10.1007/978-3-319-94277-3_90

[8] Meyer zu Bexten, E., Jung, M.: LaTeX at the University of Applied Sciences Giessen-Friedberg - Experiences at the Institute for Visually Impaired Students. In: K. Miesenberger, J. Klaus, W. Zagler (eds.) Computers Helping People with Special Needs. ICCHP 2002, LNCS, vol 2398, pp. 508–509. Springer, Heidelberg (2002)

[9] Nemeth, A.: The Nemeth Braille Code for Mathematics and Science Notation 1972 Revision. American Printing House for the Blind (1972)

[10] Pepino, A., Freda, C., Ferraro, F., Pagliara, S., Zanfardino, F.: "BlindMath" a New Scientific Editor for Blind Students. In: K. Miesenberger et al. (eds.): ICCHP 2006, LNCS, vol. 4061, pp. 1171-1174. Springer, Heidelberg (2006)

Formalizing Visualization Semantics for Accessibility

Tomas Murillo-Morales and Klaus Miesenberger

Institut Integriert Studieren, Johannes Kepler University
Linz, Austria
Tomas.Murillo_Morales@jku.at
Klaus.Miesenberger@jku.at

Abstract

This article introduces a hierarchy of ontologies for visualization with applications in accessibility to statistical charts. Formally defining graphical content and its underlying semantics enables interactive systems, such as natural language interfaces (NLIs) to efficiently retrieve information of interest for the user as well as inputting additional bits of information, both user- and system-initiated, into the graphic. In turn, such access mechanisms allow blind persons to query and navigate statistical charts and other commonly visually displayed data in an effective manner.

1 Introduction

Graphic representations of data, also known as *diagrams*, exploit the natural perceptual, cognitive, and memorial capacities of human beings in order to efficiently communicate numerical data [14]. This has made diagrams such as bar and line charts the most common way of summarizing large amounts of data in all kind of print and online publications. However, blind persons are generally excluded from accessing diagrams due to the current lack of satisfactory accessible counterparts.

Accordingly, much research attention has been given to finding non-visual alternatives to diagrams that are (1) as close as functionally equivalent to the original graphic as possible and (2) easy to create by a sighted author or, ideally, automatically generated. State-of-the-art methods for non-visual access to diagrams include refreshable tactile versions [20], hybrid techniques that combine touch and sound e.g. [10], interactive software e.g. [4], and natural language interfaces (NLIs) e.g. [18]. What these methods have in common is that they are computer-based and therefore the original graphic needs to be converted to a suitable digital format before being consumed by the end system. Therefore, the first step towards designing efficient interactive methods for non-visual access to diagrams consists on defining a suitable data model underpinning the necessary aspects of information visualization that will be requested by their human readers.

2 Problem Statement

Structuring and providing formal semantics to raster diagrams enables their translation into different non-visual modalities such as sound, speech, or haptic alternatives. Most graphic software applications, such as Microsoft Excel and Apple Numbers, let users create diagrams stemming directly from in-app data tables which are encoded into proprietary object models (OMs) file formats. OMs are independent from the structures used to persist their underlying tabular data, which is generally encoded in comma-separated values (CSV) format. In addition to proprietary formats, open formats, most commonly based on XML, are also frequently employed. Examples include GraphML [3] and GXL (Graph EXchange Language) [11], which characterize link diagrams. Last, many applications use their own XML-based graphic syntax languages e.g. [4] and [9]. Beside modelling languages, there also exist environments for popular programming languages used in statistical computing that allow technical users to author diagrams of arbitrary complexity. One of the most advanced environments of this kind is the `ggplot` library for the R programming language [23].

However, these formats are generally specific to a diagram type such as link diagrams (GXL, GraphML) or statistical charts (Excel's OM), or to a specific domain such as chemical formulae, which limits their potential applicability to a broader set of diagrammatically-represented domains. In addition, these formats do not generally provide any native means for embedding complex metadata or higher-level semantics to the graphic. Conversely, leveraging structured formalized representations of the high-level concepts depicted in images has been shown to improve the solving of complex problems supported by graphical means such as semantic image retrieval and automatic image captioning [12, 15], as well as dialogue-based navigation of photographs [21]. Furthermore, they allow for automatically disambiguating between various kinds of knowledge that can be represented by a diagram in different ways, thereby enhancing diagram interpretation [5].

Therefore, if any sort of complex interaction between a user and a diagram is to take place in a computer system, it is paramount that high-level conceptual information is added to the raw tabular data and graphical primitives depicted by the graphic. Complex, interactive software enabling blind and visually impaired persons to interact and understand diagrammatically represented information is a definite example of such kind of system.

3 A Hierarchy of Ontologies for Visualization

The idea of systematically formalizing diagram elements and their semantics into a set of general principles is coherent with current research in visualization semantics. Some well-known approaches to describing diagram semantics include the works of Richards [22] and, more recently, Wilkinson [24], in which so-called *metalanguages* for discussing diagrams are laid out. Among current methods for structuring conceptual models, ontologies are arguably the most appropriate means of embedding high-level semantics to a graphic [16, 21]. In addition, through vocabulary reuse ontologies foster system interoperability, schema credibility, and prevent designer efforts to be wasted in ensuring that the schema is well defined and properly hosted [2].

We have therefore designed a hierarchy of conceptual models stemming from a categorical adoption of the agreements found within these and other metalanguages. The end result is a set of ontologies specified in the RDF (Resource Description Framework[1]) vocabulary. Even if it has been developed with a special focus on accessibility and dialogue-based navigation of diagrams, the knowledge base is layered in such a manner that the lower levels of the hierarchy may be employed to underpin disparate diagrammatic representations regardless of application. A layered approach to ontology design stimulates maximal reuse of domain ontologies while not falling into under-specification pitfalls [7] and being consistent with the widely-shared notion of an aggregate-composite approach to graphic meaning with higher-level semantic concepts taking precedence over lower-level ones see e.g. Engelhardt's "recursive nature of graphics" [8]. The suggested model of formalized visualization semantics [2] is structured hierarchically in five

[1]For an introduction to domain modelling in RDF, see https://www.w3.org/TR/rdf-primer/

[2]The aggregated, up-to-date vocabulary is published at https://ontovis.integriert-studieren.jku.at/ontovis-full/. Please note that the knowledge base may still be under development at the time of reading.

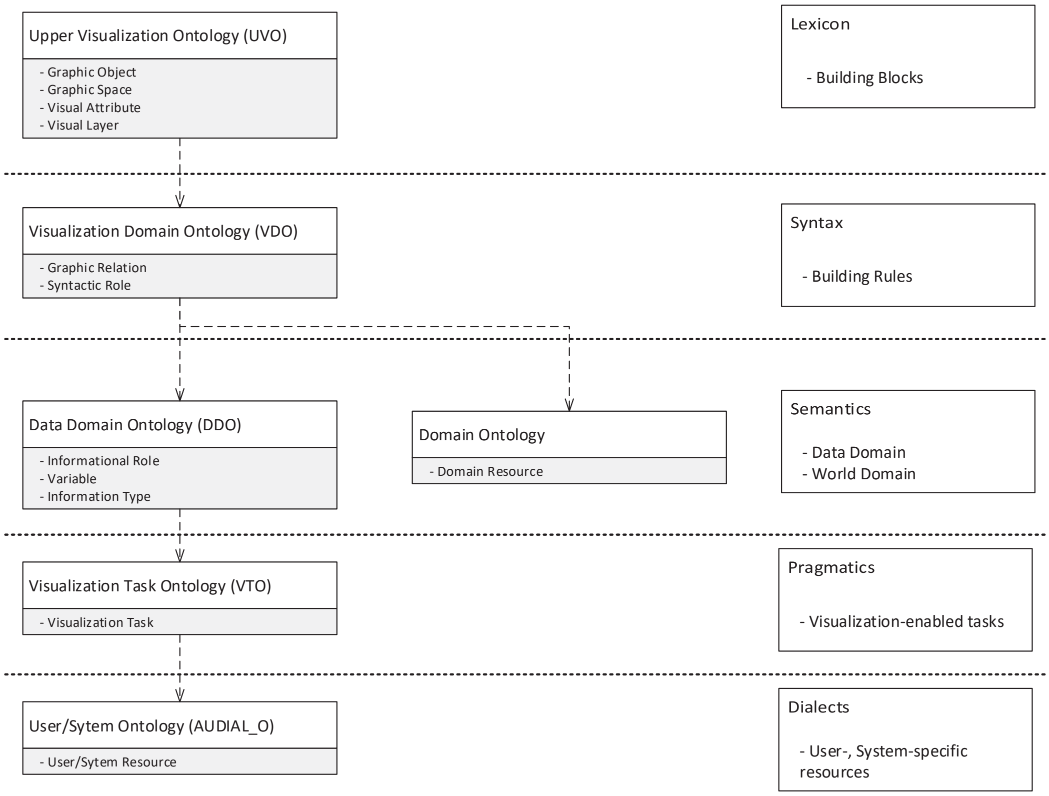


Figure 1: Hierarchical five-layer ontology pattern for visualization

levels as depicted in Figure 1. Each level groups logically resources corresponding to a different conceptual level, as follows:

- **Upper Visualization Ontology *(UVO)***. Deriving a parallel world view to that of descriptive linguistics, visualization semantics are built on top of an inventory of available vocabulary, known as lexicon. The lexicon of visualization is likewise comprised of what could be deemed as "nouns" (*graphic objects* and *graphic spaces*) and "adjectives" (*visual attributes* and *visual layers*) that provide additional visual information about graphic objects and spaces. The UVO is therefore the foundation upon which higher level constructs depend for their interpretation.

- **Visualization Domain Ontology *(VDO)***. This level underpins properties of the visual primitives defined in the UVO that enrich them with the semantic knowledge necessary for the correct interpretation of the diagram. Carrying on with the likeliness of visual and linguistic structures we could portray VDO as including "verbs" (*graphic relations*) semantically linking graphic elements together and syntactic knowledge about the roles of each resource taking part in a graphic relation (*syntactic roles*).

- **Data Domain Ontology *(DDO)***. UVO and VDO describe visualization itself, not the latent data conveyed by the diagram nor higher level concepts grounded on graphic objects and relationships. The next level of the semantic hierarchy, the DDO, underpins the raw tabular structure of the diagram information by means of ontological resources, such as *variable names*, *informational roles* of graphic objects, and *information types.* In addition, higher-level semantics of this information may be necessary for its proper understanding relative to the perspective of the viewer, which may be included in the knowledge base by means of object properties connecting resources from external domain ontologies with the resources described in the visualization ontologies of the hierarchy. This approach parallels that of Communicative Images, in which individuals appearing on a photograph are abstracted by means of suitable domain ontologies providing vocabulary and background knowledge about them [21].

- **Visualization Task Ontology *(VTO)***. This ontology describes the most relevant vocabulary of user tasks that can be performed on diagrammatically depicted data (*visualization tasks*). The vocabulary of tasks depends on the data depicted by the diagram more than on how the data is diagrammatically depicted. Moreover, a focus on analytic primacy instead of on representational primacy greatly benefits end-users of information visualization by not relying on the author's own insights and expressive capability within a particular tool [1]. For these reasons, VTO lays on top of DDO in the visualization hierarchy. Nevertheless, some user tasks, such as retrieving the slope/trend of a line segment, may rely on how the data is particularly visualized.

- **User/System Ontologies**. The knowledge base underpinning a diagram's semantics may be accessed in very different ways depending on the specific needs of the end user and the client application. Therefore, the topmost level of the hierarchy consists of user- and system-specific resources which help with handling the knowledge contained in the bottom ontologies by end users and applications. For example, an accessible Natural Language Interface (NLI) to diagrams may define resources at this level in order to offer its users mechanisms that aim to compensate for their lack of sight [17].

4 Semantically-Enhanced Graphics

The formal resources contained in the knowledge base described in the previous section may be instantiated accordingly and embedded in a diagram in order to enhance its raw visual features with actionable semantics, resulting in a so-called *Semantically-Enhanced Graphic.* The semantic resources of the graphic may then be retrieved with the purpose of, for example, making it accessible to blind users. AUDiaL (Accessible Universal Diagrams through Language) is a web-based Natural Language Interface (NLI) prototype to Semantically-Enhanced statistical charts that has shown promising results in enabling blind users to access diagrams in a convenient, natural, and inexpensive manner [18]. An example of a simple Semantically-Enhanced chart that may be processed by AUDiaL so that it can be accessed non-visually is given next.

4.1 An Illustrative Example

In the following we outline the process of semantically annotating a simple statistical chart in SVG format by linking its graphical primitives to resources of the knowledge base introduced in this paper. This process, which some authors have named "bridging the semantic gap" [13], can be performed manually with specialized authoring tools, known as semantic image annotators (see [6] for a survey on semantic image annotation tools). The base diagram that will be semantically annotated is shown in Figure 2.

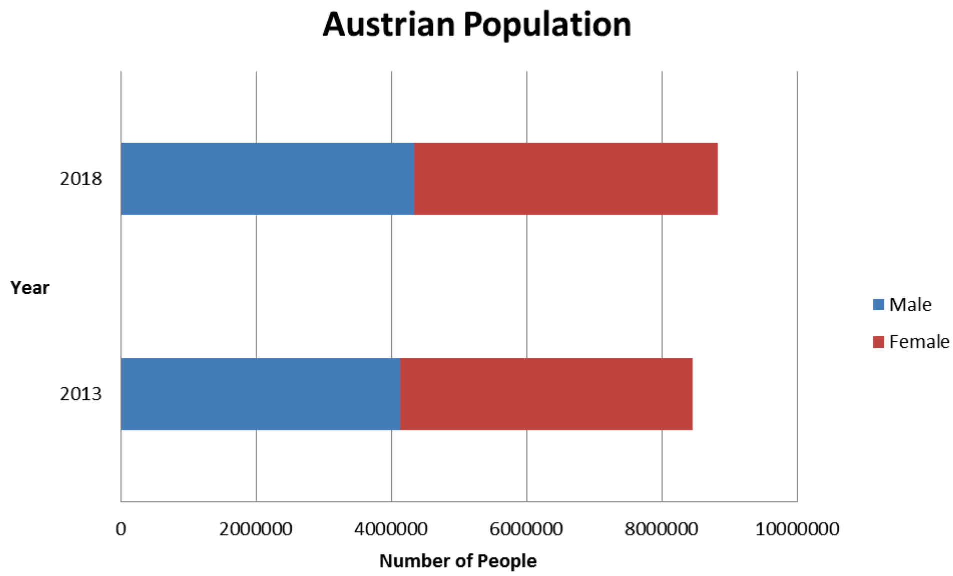


Figure 2: A simple SVG stacked bar chart showing the total population of Austria, divided by gender, for the years 2013 and 2018

Figure 3 depicts the formal relationship between a number of selected constituent elements of the diagram from Figure 2 and UVO resources. Recall from Section 3 that UVO characterizes the foundational visual primitives of the diagram. Namely, several Composite and Elementary Graphic Objects present in the bar chart are shown being associated to instances of corresponding UVO classes. Such association takes place by means of `uvo:hasSVGElement` datatype property occurrences, which relate ontological resources (triple's subject) to SVG shapes as given by their `id` attribute (triple's object). Semantic Annotator for Inkscape (SAI, [19]) is a semantic image annotator that has the ability of automatically generating these and other necessary property occurrences in a transparent manner to the human annotator. Three Graphic Spaces are additionally displayed in Figure 3. These are `GS1`, an Arbitrary Graphic Space in which the main diagram and its legend reside; `GS2`, an Elementary Metric Space in which one dimension (the length of the bars) is subject to strict interpretation; and `GS3`, a Meaningful Graphic Space that hosts the constituent elements of the diagram's legend.

Example annotations involving the next two levels of the hierarchy illustrated in Figure 1 (namely, VDO and DDO) for the bar chart are shown in Figure 4. These include Graphic Relations between Graphic Objects (e.g. the containment Graphic Relation `GR1` or a number of labeling relations), Syntactic Roles of Graphic Objects taking part in Graphic Relations (e.g. the `vdo:Label_SR` role used for labels), and Informational Roles of Graphic Objects, among others. Note that labeling is a special kind of Graphic Relation which, unlike most Graphic Relation instances e.g. `GR1`, is not characterized by instantiating a subclass of the `vdo:Graphic_Relation` class. This is because labeling is a relationship of a higher concep-

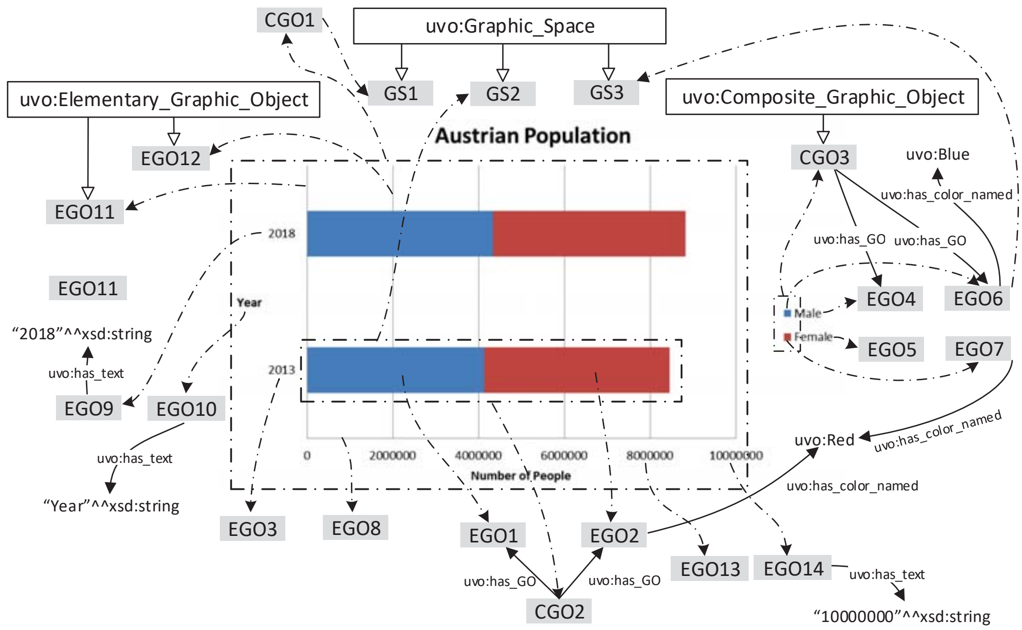


Figure 3: Semantic annotation of some selected diagram elements with resources from the Upper Visualization Ontology (UVO). Grey boxes represent named UVO class instances. Dashed boxes group SVG shapes into composite graphic objects. Dashed arrows indicate association between SVG shapes and formal resources of the UVO. Solid arrows represent other property occurrences. Note that only a subset of all possible annotations are shown in this figure.

tual level than other Graphic Relations underpinned in VDO that can be derived from a basic object-to-object Graphic Relation between the label and the node it labels. Most commonly, this foundational Graphic Relation is either one of clustering, where the label is placed in close proximity to the labeled nodes; or one of linking, where the label and the nodes it labels are connected by connectors. However, given the prevalence of labels in diagrams, most authors give dedicated attention to it in their models and taxonomies for visualization. Therefore, occurrences of the `vdo:is_labeled_by` object property (and its inverse property, `vdo:is_label_for`) have been included to VDO to axiomatically connect label and node (i.e. labeled) Graphic Objects taking part in labeling relationships, as shown in Figure 4.

This example also serves to illustrate how UVO and VDO leverage the so-called principle of compositionality of meaning, also known as *Frege's principle* [8], which defines graphic syntax in a recursive manner such that a collection of perceivable objects arranged in a given structure in two-dimensional space may function as a single graphic object at a higher level of abstraction. More specifically, the syntax of diagrams under Frege's principle can be decomposed as follows:

1. A *Graphic Representation* (diagram) is a *Graphic Object*

2. A *Graphic Object* may be:

    (a) An *Elementary Graphic Object*

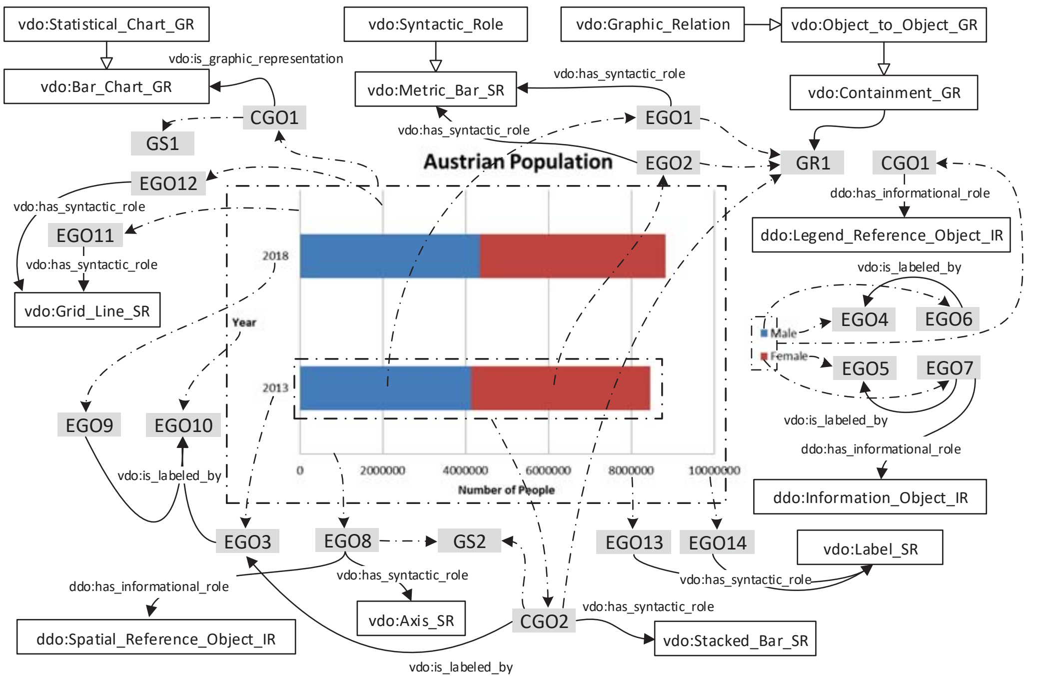


Figure 4: Semantic annotation of some selected diagram elements with resources from the Visualization Domain Ontology (VDO) and Data Domain Ontology (DDO). These include Graphic Relations, and Syntactic and Informational Roles of Graphic Objects, among others. This figure follows the same format as Figure 3. Note that only a subset of all possible annotations are shown in this figure.

(b) A *Composite Graphic Object*, made up of:

  i. A *Graphic Space* that is occupied by it
  ii. A set of Graphic Objects, which are contained within that Graphic Space, and
  iii. A set of *Graphic Relations* in which these graphic objects are involved

In our example graphic, the topmost Graphic Object of the diagram (not shown) encompasses the bar chart as whole. In turn, it can be decomposed into a Graphic Space (`GS1`), and a set of Graphic Objects; namely, the bar chart itself (`CGO1`), its legend (`CGO3`), and its title (not shown). If we now choose to decompose `CGO3` one more level, it is made up of a Graphic Space (`GS3`), and four Elementary Graphic Objects (`EGO4`, `EGO5`, `EGO6`, and `EGO7`), which are taking part in two labelling Graphic Relations (not depicted in Figure 4). Each Composite Graphic Object may be decomposed in this manner until all atomic Elementary Graphic Objects that make up the graphic (corresponding to basic SVG shapes) are reached.

Resources of the next level of the hierarchy shown in Figure 1 i.e. the Visualization Task Ontology (VTO), cannot be related to Graphic Objects since they underpin tasks that users may perform on the depicted data, not data itself or its visualization. Tasks characterized by VTO are subclasses of the `vto:Visualization_Task` class, which include both analytical and navigational tasks that can be supported by diagrammatically displayed information. Analytical tasks are low-level endeavors, more specific and focused in nature than high-level goals, which capture the majority of sighted people's activities while employing information visualization tools for understanding data [1]. Examples include finding anomalies/outliers in a certain attribute of a set of data cases (`vto:Find_Anomalies_Task`), retrieving the nodes adjacent to a given node of a link diagram (`vto:Retrieve_Adjacent_Task`), or simply computing the arithmetic mean of a set of attributes (`vto:Compute_Average_Task`). On the other hand, navigational tasks support the specific needs of blind persons accessing diagrams by characterizing a number of low-level activities that may be employed to explore the diagram non-visually. These include tasks meant to go through the diagram's elements sequentially (e.g. `vto:Move_Next_Task` for moving to the next node in a navigational sequence), tasks underpinning navigational shortcuts (e.g. `vto:Go_to_Highest_Task` for quick jumping to a salient node having the maximum value of a certain attribute), and tasks that aid in preventing disorientation during navigation (e.g. `vto:Where_Task` which informs users about their current location).

Given the expressive limitations of Description Logics such as RDF, VTO merely characterizes tasks, but it is not able to execute or process them for a specific diagram being consumed by a client application. End applications, such as AUDiaL, may employ their own formal languages to action Semantically-Enhanced Graphics. In order to do so they may have to define additionally their own RDF constructs at the User/System conceptual level depicted in Figure 1. For example, AUDiaL makes extensive use of a proprietary datatype property (`aud:task_has_verbalization`) in order to associate VTO tasks to phrases in natural language. If a user employs a phrase appearing in the subject of an occurrence of this property, its corresponding task may be executed accordingly. AUDiaL further employs resources at this level to enable users to add their own labels and other annotations to the graphic in order to compensate for their lack of sight, see [17] for details.

5 Conclusions

Whereas methods that automatically deal with simple perceptual tasks on images, such as classification and object detection, have experienced a thrilling increase in quality and efficiency, user tasks at higher conceptual levels, such as visual question answering in diagrams, are still not achievable by these technologies at a usable level. The integration of a suitable vocabulary of well-known *a priori* knowledge into the primitive graphical components of a diagram enables the realization of cognitive tasks, such as question answering, on diagrammatically displayed information.

Such a formal vocabulary for describing visualization semantics needs to conform to the different conceptual levels underpinning the descriptive dissonance expressed by the semantic gap between low-level concepts (e.g. image primitives) and high-level concepts (e.g. domain knowledge and tasks). In this paper we introduce a formal knowledge base in the form of a hierarchy of ontologies that aims to bridge this gap for diagrammatically displayed data, such as statistical charts. The annotation of raw graphical data with semantic information results in a so-called *Semantically-Enhanced Graphic*. We have furthermore outlined a particular use case for Semantically-Enhanced Graphics; namely, Natural Language Interfaces (NLI) to diagrams have the potential of enabling blind persons to access statistical charts non-visually in an efficient, effective, and user-friendly manner. AUDiaL [18] is a prototype of such application that we will continue exploring and working on.

References

[1] Robert Amar, James Eagan, and John Stasko. Low-level components of analytic activity in information visualization. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, INFOVIS '05, page 15, Washington, DC, USA, 2005. IEEE Computer Society.

[2] Phil Archer, Nikolaos Loutas, and Stijn Goedertier. Cookbook for translating relational data models to rdf schemas.

[3] Ulrik Brandes, Markus Eiglsperger, Jürgen Lerner, and Christian Pich. Graph markup language (graphml). In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, Discrete Mathematics and Its Applications, pages 517–541. CRC Press, Hoboken, 2013.

[4] Andy Brown, Steve Pettifer, and Robert Stevens. Evaluation of a non-visual molecule browser. In Julie A. Jacko and Andrew Sears, editors, *Assets '04 Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*, pages 40–47. ACM, New York, NY, USA, 2003.

[5] Peter Cheng, James Cupit, and Niger R. Shadbolt. Supporting diagrammatic knowledge acquisition: An ontological analysis of cartesian graphs. *International Journal of Human-Computer Studies*, 54(4):457–494, 2001.

[6] Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. A survey of semantic image and video annotation tools. In Georgios Paliouras, Constantine D. Spyropoulos, and G. Tsatsaronis, editors, *Knowledge-driven multimedia information extraction and ontology evolution*, Lecture notes in computer science. State-of-the-art survey, pages 196–239. Springer, Heidelberg, 2011.

[7] Michel Dumontier and Natalia Villanueva-Rosales. Three-layer owl ontology design. In Bernardo Cuenca Grau, Vasant Honavar, Anne Schlicht, and Frank Wolter, editors, *Proceedings of the 2nd International Workshop on Modular Ontologies*, 2007.

[8] Jörg von Engelhardt. *The Language of graphics: A framework for the analysis of syntax and meaning in maps, charts and diagrams*, volume 2002-03 of *ILLC dissertation series*. Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam, 2002.

[9] Jinglun Gao. *Information graphic classification, decomposition and alternative representation.* Phd dissertation, University of Delaware, 2013.

[10] Cagatay Goncu and Kim Marriott. Gravvitas: Generic multi-touch presentation of accessible graphics. In Pedro Campos, editor, *Human-Computer Interaction – INTERACT 2011 Proceedings of the 13th IFIP TC 13 International Conference*, Lecture Notes in Computer Science, pages 30–48. Springer, Heidelberg, 2011.

[11] Richard C. Holt, Andy Schürr, Susan Elliott Sim, and Andreas Winter. Gxl: A graph-based standard exchange format for reengineering. *Science of Computer Programming*, 60(2):149–170, 2006.

[12] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, and David A. Shamma. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[13] Halina Kwaśnicka and L. C. Jain. Semantic gap in image and video analysis: An introduction. In Halina Kwaśnicka and L. C. Jain, editors, *Bridging the semantic gap in image and video analysis*, Intelligent systems reference library, pages 1–6. Springer, Cham, Switzerland, 2018.

[14] Stephan Lewandowsky and Ian Spence. The perception of statistical graphs. *Sociological Methods & Research*, 18(2-3):200–242, 1989.

[15] Yuee Liu. *Ontology-based image annotation.* Phd thesis, Queensland University of Technology, Queensland, 2010.

[16] Tomás Murillo-Morales and Klaus Miesenberger. Ontology-based semantic support to improve accessibility of graphics. In C. Sik-Lányi, E.-J. Hoogerwerf, and K. Miesenberger, editors, *Studies*

in health technology and informatics, volume 217, pages 255–260. IOS Press, 2015.

[17] Tomas Murillo-Morales and Klaus Miesenberger. Techniques for improved speech-based access to diagrammatic representations. In Klaus Miesenberger and Georgios Kouroupetroglou, editors, *Computers Helping People with Special Needs. 16th International Conference*, Lecture Notes in Computer Science, pages 636–643. Springer, 2018.

[18] Tomás Murillo-Morales and Klaus Miesenberger. Audial: A natural language interface to make statistical charts accessible to blind persons. In Klaus Miesenberger, Roberto Manduchi, Mario Covarrubias Rodriguez, and Petr Peňáz, editors, *Computers Helping People with Special Needs*, Lecture Notes in Computer Science, pages 373–384. Springer, Cham, 2020.

[19] Tomás Murillo-Morales, Jaromir Plhák, and Klaus Miesenberger. Authoring semantic annotations for non-visual access to graphics. *The Journal on Technology and Persons with Disabilities*, 6:398–413, 2018.

[20] Sile O'Modhrain, Nicholas A. Giudice, John A. Gardner, and Gordon E. Legge. Designing media for visually-impaired users of refreshable touch displays: Possibilities and pitfalls. *IEEE transactions on haptics*, 8(3):248–257, 2015.

[21] Jaromir Plhák. *Dialogue-based Exploration of Graphics for Users with a Visual Disability.* Doctoral thesis, Masaryk University, Faculty of Informatics, Brno, 1/1/2014.

[22] Clive James. Richards. *Diagrammatics : an investigation aimed at providing a theoretical framework for studying diagrams and for establishing a taxonomy of their fundamental modes of graphic organization: Ph.D Thesis.* PhD thesis, Royal College of Art, 1984.

[23] Hadley Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, 2010.

[24] Leland Wilkinson. *The grammar of graphics.* Statistics and computing. Springer, New York, 2nd ed. edition, 2005.

Use of Mobile Devices for Haptic Interaction for classifying 3D Geometrical Objects

Aashish Kumar Verma, Klaus Miesenberger, Gregor Pertl and Kerstin Reithmayr
Johannes Kepler University Linz, Institut Integriert Studieren, Austria
{aashish.verma, klaus.miesenberger}@jku.at, gregor-pertl@gmx.at, kerstin.reithmayr@gmail.com

Abstract

Technology is the main medium to spread inclusive education and reduce digital divide in almost all domains of disability. Accessibility measures, as e.g. alternative text, long description, often including raw table data, haptic graphics, do exist but in many aspects do not provide a good alternative user experience. Other, more sophisticated methods tend to be complex or expensive. We propose a new method for blind and low vision persons (BLV) to access 3D geometrical shapes using haptic/vibrating features of of-the-shelf mobile devices based on using a mobile application. The application converts 3D geometrical shapes into a 3D virtual model. The user can move the mobile device and thereby virtually explore and feel the 3D geometrical shapes. We develop a low cost inclusive application using augmented reality where sighted user can access objects/models visually (also using haptics if they want) together with BLVs, who explore the 3D geometrical objects. Each 3D shapes can be assigned with different vibration patterns and height with respect to the geometrical shape. This approach provides an easy to implement, inexpensive and of the shelf experimental setting for improving and augmenting the user experience in accessing 3D geometrical shapes. The approach is also considered for experimenting in STEM education and or other subjects using graphical representation. We expect a broad field for ideation, experimenting and sharing to support new approaches for accessibility and inclusive education and cooperation.

1 Introduction

Mobile applications support BLV in many domains of daily life like to read mails, send replies to their mails, navigation, chatting, online shopping, creating and editing documents, and access internet etc. They have been early adaptors and power users in the digital revolution. Therefore, we can expect them do have access to mobile devices and have reasonable skills in using them with the embedded or connected assistive technologies, what also holds true at global level [e.g. (WHO, 2011), (World bank)].

The state of the art (see below) underlines that the standard way of using mobile devices and accessing information of BLV is based on screen readers and text to speech engines (TTS). Accessing text is straight forward. Other means of access, as e.g. haptic graphics still do need a considerable amount of work for preparation of materials, education and training for users and often expensive devices. Better production facilities (e.g. 2/3D printing, 2/3D modelling software) improved the situation but still these approaches of access are limited. Also different haptic devices are available as assistive technology for blind persons to analyze 3D virtual models. They create a touch experience by adding vibrations, forces or motions to the user's hand. These devices give an ability to the user to touch and manipulate virtual 3D models. Generally, these haptic devices are costly (Yu, 2002), so it is rarely used by blind persons in their daily life; it is still more available in lab-situations only. And finally, an according didactics, education and training to learn coping with more complex graphics as well as standards, methods and tools in preparing graphics are not really at hand. (Müller K., 2010) This situation, as further examined in the state of the art section, demands for a broadly available, easy to use and cost-effective testing bed to allow educators, trainers and even users to experiment with virtual 3D access to come to new user-driven approaches.

Based on this analysis we propose to use of the shelf mobile devices and mobile applications providing a haptic experience to explore, explain, understand and use 3D virtual models. We use 3D geometrical shapes as a first example of widely used graphical representations.

2 State of the Art

The state of the art in making graphics accessible can be summarized as a) providing alternative text (Bigham, 2006), which tends to be too short to convey the information sufficiently, b) a long description, which tends to be of lower usability and lacking in a comparable experience (Hoffman, 2003), c) providing access to raw data (e.g. tables) (Hoffman, 2003), what is not really easy to use in terms of developing mental models d) producing haptic 2D graphics based on the shapes of visual graphics (Xu, 2011), which are labour-intense and dependent on use and know-how of standards, e) providing 3D printed models (Götzelmann, 2016), which become easier to access but also would need according standards and training, f) combinations of audio and haptic representation (Sánchez, 2011) and finally g) virtual haptic representation using haptic-force-feedback devices (Yu, 2002).

Haptic devices are used as assistive technology to help persons with visual impairment, motor disability, and hearing impairment. Refreshable braille display (Abdelkader, 2007) is a kind of haptic device which is used by blind persons to read Braille encoded text through tactile feedback. Electro-stimulation wearable gloves developed by Meers et al. (Meers, 2005) which helps blind in navigation and avoiding obstacles.

Tang et al. (Tang, 2006) developed an oral-tactile mouthpiece which helps visually challenged persons in navigation. Force feedback joysticks (Abdelkader, 2007) are based on the haptic technology. It assists in better wheelchair control for the person with motor disability. Haptic based prosthesis assists in better grasp, grip and improve manipulative ability of persons with hand functioning problem. These prostheses are based on mechanoreceptors. Electromyographic (EMG) (Rivera, 2014) [4] based haptic technology aids people with motor disability to control wheelchairs and ICT based devices.

Phantom from SensAble Technologies (Yu, 2002) is a haptic based technology helps blind persons to access graphic display. Calle et. al (Sjöström, 2009) mention this technology a new way to access graphical information for blind persons. Logitech WingMan Force Feedback mouse (Yu, 2002) has shown great potential in rendering 2D plots for visually impaired people. The virtual reality kit (Haptic gloves help blind people to 'see' art, 2018) from Neural Digital Technologies and Geometry Prague help blind and visually impaired people to see art in the form of 3D virtual models. This virtual kit contains the haptic gloves. The state of the art consists of

- Creation of 3D virtual models
- Use of haptic gloves to feel the artwork
- Gloves produce vibrations and send sense of touch to the brain

This analysis let us conclude that approaches tend to be limited in scope and availability. In particular they miss the possibility of a low cost and low-tech approach by non-technical experts (e.g. trainers, educators, care givers, users) for experimenting and ideation at broader scale. Therefore, the goal of our research is defined as facilitating

- approaches to virtual haptic access to graphical information, starting with 3D geometrical shapes,
- adding additional cues e.g. using speech and non-speech audio,
- using of the shelf haptic features included in mobile devices,
- user- and expert driven experimenting in accessing, exploring and using graphical information.

3 Design and Method

We have used this new approach to convert the 3D geometrical shapes into a 3D virtual model:

- Create 3D virtual model of 3D geometrical shapes using augmented reality
- Use of mobile to feel the different 3D geometrical shapes
- Mobile produces vibrations and send sensation of touch to the brain

We create a new mobile applications based on this approach which allows a fast and easy 3D representation of objects and models for both visual and haptic access. This mobile application in a first approach to classify 3D geometric objects using mobile as haptic device. We have used Unity3D and Vuforia to create augmented reality of 3D geometry model. We work with computer science students and students with disabilities following a user centered design and development approach.

We have interviewed teachers of BLV students for requirement gathering. Our questions were related to current issues facing BLV students during their education. These teachers belonged to developed countries and developing countries. We have found out these main issues:

Lack of clarity of simple concepts due to lack of educational resources: As per teachers, it is difficult for BLV students to understand the concept. For instance, if they read in the book “Duck swims in the water”. This information is only available in the text for them. It is difficult for BLV students to understand how a duck swims in the water. When teacher asked the same question for them, how the duck swims in the water? Does it swim like a fish in the water or swim on the surface of the water? The answer from most of the BLV students was “It swims like a fish inside the water”. This shows the lack of clarity of BLV students due to lack of educational resources.

Lack of fast upgradation of Braille books: Braille books are costly as compared to normal textbooks. Some BLV students are following old textbooks due to their high cost. It leads to different problems. For instance, it is difficult for them to note their homework and solve the problem together with other BLV students because of change in the page numbers of braille books of new edition etc.

In the new era, ICT (Information and communication technology) provides audiobooks to them. It is easy to edit and upgrade these books. However, it does not contain every single detail as we have given the duck example above. Therefore, there is a need for ICT graphical technology, which is accessible and helps BLV students to understand the various concepts of STEM education.

On the other side, students with sight are moving towards virtual and augmented reality to understand the complex concept of STEM education. Most graphical content of these mixed reality applications are not accessible for BLV students. These mixed reality applications are mostly useful for entertainment, education, marketing and surgeries. As mentioned in the state of the art, there are different technologies available to access them graphically. The cost of these assistive technologies are costly for BLV students.

The main challenge was, how we can make these 3D virtual models more accessible and in a cheap way. Consequently, we have tried to use mobile phones to access 3D graphical virtual models.

Started with mobile-based android application: We have started our research with android application. Initially, we have generated 3D virtual geometrical models like cube, sphere and pyramid. However, it was difficult for BVP to locate the 3D model in the 3D space. Therefore, we have these research questions further.

- How can BVP locate the 3D virtual model in the 3D space?
- How can BVP feel 3D virtual models with the help of mobile?

Locate the 3D model in virtual space with the help of reference sheet: We have used reference sheet to locate the augmented 3D model. With computer vision, if you try to see the reference sheet with the camera of a mobile. It creates a 3D augmented model on the reference sheet as given in the figure 1. This reference sheet could be a part of the STEM notebook or it could be a separate sheet of paper. The BVP can simply place a reference sheet on the table and move the camera of the mobile towards it to locate a 3D augmented model. This procedure helps BVP to locate the 3D model in 3D space.

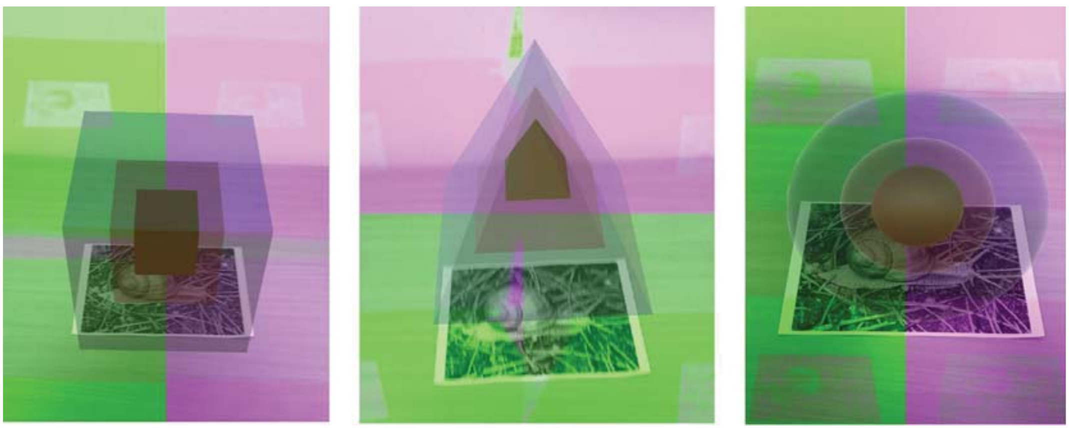


Figure 1: This figure shows augmented 3D models of basic geometric shapes (cuboid, pyramid and sphere from left to right) on the reference sheet for both visual and haptic access. These are screenshots of android-based mobile application. We have used Unity **(Unity)** and Vuforia **(Vuforia Engine)** to make this mobile-based application.

Experiment to feel 3D model with the mobile haptic feedback: We have done many trials to feel the 3D model as 3D shape with the help of different ranges of mobile vibrations. We have taken an example of a 3D model of cuboid. To present a 3D model of the cuboid, we have taken three sets of 3D cuboids. These cuboids of decreasing size placed one inside each other as shown in the figure 2. We have named these three cuboids as outer cuboid, middle cuboid and inner cuboid for our explanation. The space between the outer cuboid and middle cuboid is represented with light vibration, the space between middle cuboid and inner cuboid has medium vibration. Additionally, the inner cuboid has intense vibration. This kind of vibration practice feels the cuboid shape to the application user without looking at the mobile screen. We have done the above experiments with the sighted user by closing their eyes and without looking at the screen to feel the cuboid shape. We believe that blind and low vision people could also experience the cuboid with these sets of vibrations. Similarly, we have created other geometrical shapes (pyramid and sphere) as given figure 1.

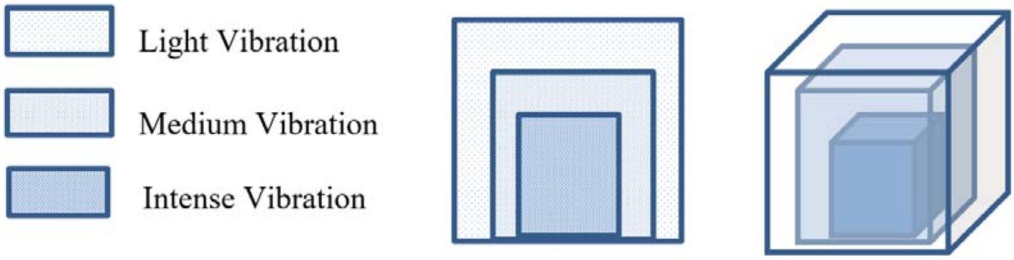


Figure 2: Cuboid geometrical shape represented with outer cuboid, middle cuboid and inner cuboid. The different parts of cuboid has light, medium and intense vibration.

Technology Used: We have used Unity version 2018.4.11 with the module of Vuforia Augmented reality support. The core functionality comes from the basic features of Vuforia combined with Unity colliders (Unity Colliders Documentation) on the camera and the instantiated objects. Under the Vuforia module, we able to create an AR Camera, which represents the device's camera. It has a small box collider as a trigger attached. The each of the shapes consist each out of 3 layers with each one having a collider. Unity's handheld.vibrate (Handheld.vibrate) method is used to vibrate the mobile for haptic feedback.

Testing: The testing has been done with following questions:

1. Can you guess the geometrical shape between sphere, cuboid and pyramid?
2. Can you differentiate between the sphere and cuboid?
3. Can you differentiate between the sphere and pyramid?
4. Can you differentiate between the cuboid and pyramid?

We were not able to do testing with BLV. Therefore, we have done testing with sighted people without looking at the screen. The results are as follows:

- They can guess only a pyramid correctly out of sphere, cuboid and pyramid with the haptic feedback through mobile.
- It is difficult for them to differentiate between the sphere and cuboid.
- Most of them are able to differentiate between the sphere and pyramid.
- They are able to differentiate between the cuboid and pyramid.

Consequently, we can say that it is difficult for them to understand the curves through these set of vibrations. The mobile has its limitations with vibrations like every technology has its own limitations. Therefore, there is a need of guidelines to use mobile as haptic device to feel 3D virtual models. We believe that we can use mobile as haptic device to access different virtual 3D model in STEM education.

4 Conclusion

This research intends to provide a technical framework allowing an experimenting and testing bed for practitioners (trainers, educators, users) for new, low-cost and easy to implement approaches helping BLV to use mobile devices as the haptic tool to study graphical data interactively with the 3D virtual model, starting with 3D geometrical shapes open for augmenting speech and non-speech audio, which of course can also be considered as a new and interesting approach in education in general. As compared to previous technologies, it is cheap as there is no need of tactile print or need to buy the separate haptic device. The approach is portable and at hand when needed. The approach is not intended as a final solution per se but as a framework allowing crowed-sources experimenting and playing to come to viable solutions for BLV and other groups of learners. As a base for a more user-driven approach this should support better access in the long run also by provoking interdisciplinary R&D. Additionally, this work motivates researchers to use mobile as the haptic device for different 2D or 3D virtual models.

References

Abdelkader, M. B. (2007). Haptic Feedback and A Motor Disability: Preliminary Study. *HuMaN 07 proceedings.*

Bigham, J. K. (2006). WebInSight:: making web images accessible. *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility.* Association for Computing Machinery, New York, NY, United States.

Carlisle, D. (2010, April). *graphicx: Enhanced support for graphics.* Retrieved from http://www.ctan.org/tex-archive/ help/Catalogue/entries/graphicx.html

Götzelmann, T. (2016). LucentMaps: 3D Printed Audiovisual Tactile Maps for Blind and Visually Impaired People. *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)* (pp. 81-90). Proceedings of the 18th International ACM SIGACCESS Confere Association for Computing Machinery, New York, NY, USA. .

Haptic gloves help blind people to 'see' art. (2018, July 3). Retrieved from BBC News: https://www.bbc.com/news/av/technology-44606631/haptic-gloves-help-blind-people-to-see-art

Hoffman, B. H. (2003). THE REALIZATION OF ACCESSIBLITY: Guidelines for the Creation and Refinement of Universal Online Learning Applications. *Proceedings of EdMedia + Innovate Learning 2003* (pp. 3206-3213). Honolulu, Hawaii, USA: Association for the Advancement of Computing in Education (AACE), Waynesville, NC.

Meers, S. W. (2005). A substitute vision system for providing 3D perception and GPS navigation via electro-tactile stimulation. *IntConfSensTechnol*, (pp. 551-556).

Müller K., C. A. (2010). Improving the Accessibility of ASCII Graphics for the Blind Students. *Computers Helping People with Special Needs.* Vienna, Austria: Springer, Berlin, Heidelberg.

Rivera, L. D. (2014). Haptic and Gesture-Based Assistive Technologies for People with Motor Disabilities. In G. Kouroupetroglou, *Assistive Technologies and Computer Access for Motor Disabilities* (pp. 1-27). IGI Global.

Sánchez, J. M. (2011). Audiopolis, Navigation through a Virtual City Using Audio and Haptic Interfaces for People Who Are Blind. *International Conference on Universal Access in Human-Computer Interaction. UAHCI 2011* (pp. 362-371). Springer, Berlin, Heidelberg.

Sjöström, C. D.-G. (2009). Phantom-based haptic line graphics for blind persons. *Visual Impairment Research*, 13-32.

Tang, H. B. (2006). An Oral Tactile Interface for Blind Navigation. *Tang, H., Beebe, D.J.: An Oral Tactile Interface for Blind Navigation. IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society.* , 116-123.

Unity. (n.d.). Retrieved from https://unity.com/

Voronkov, A. (2004). *EasyChair conference system*. Retrieved from easychair.org

Voronkov, A. (2014). Keynote talk: EasyChair. In *Proceeedings of the 29th ACM/IEEE International Conference on Automated Software Engineering* (pp. 3-4). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2643085&dl=ACM&coll=DL

Voronkov, A., & Hoder, K. (n.d.). *Templates.* Retrieved from Templates for proceedings: https://easychair.org/proceedings/template.cgi?a=12732737

Vuforia Engine. (n.d.). Retrieved from https://developer.vuforia.com/

WHO. (2011). *World Report on DIsability.* Retrieved from WHO: https://www.who.int/publications-detail/world-report-on-disability

Wikipedia. (n.d.). *EasyChair.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/EasyChair

World bank. (n.d.). *Disability Inclusion.* Retrieved from World bank: https://www.worldbank.org/en/topic/disability

Xu, C. I. (2011). Tactile display for the visually impaired using TeslaTouch. *In CHI ’11 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’11)* (pp. 317-322). Association for Computing Machinery, New York, NY, US.

Yu, W. B. (2002). Comparing two haptic interfaces for multimodal graph rendering. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002.* Orlando, FL, USA: IEEE.

Reading Graphics by Audio-Touch

John Gardner
ViewPlus Technologies, Inc
John.gardner@viewplus.com

Abstract

A computer-literate blind person today can easily access static text, but non-text information remains difficult or impossible. We do not fully understand how to make some types of dynamic, interactive information truly accessible, but we have known for many years how to access almost all static information. Unfortunately, understanding has not yet resulted in actual practical ability to access much critically important STEM information. Well-made documents and good user technology today can provide reasonably good access to reading math, but ensuring that documents are published in these accessible formats remains a daunting challenge. And even at best, creating math is still clumsy for blind authors. Graphics however remains the greatest challenge. This paper includes an extensive review of current graphical access methods. It is also a progress report on the author's development an experimental web-based graphics accessibility application and an associated application enabling certain STEM graphics to be authored by and for blind people.

Introduction

Modern screen readers provide persons with visual impairments or other print disabilities support to read electronic literature. But while screen readers make text almost universally accessible, they generally provide little support for images. Informational graphics - charts, diagrams, and graphs - are critical for STEM (Science, Technology, Engineering, Math) subjects but are much more ubiquitous.

In everyday news and other written communications, graphics often contain more information than the words. Since this content is usually inaccessible, persons with visual and print impairments are being left behind in today's knowledge society.

There are a number of ways to "make graphical information accessible", and these are reviewed in the next section. Currently all require a major effort on the part of a sighted interpreter. Since losing his sight in mid-career, this author has devoted much of his time to developing technologies that enable better graphics access and that can lead to a future in which graphics can be accessible "out of the box" with no need for sighted assistance. If this cannot be achieved, people with print disabilities will never have truly equal access to information.

2.0 Overview of Graphics Accessibility

2.1 Word Description

The most common current practice for graphics accessibility is to provide a word description for the image. These descriptions are usually created by humans, but computer applications are becoming available that will create some simpler ones automatically. Facebook has a graphics describer usable by anybody, and Microsoft provides a free iPhone app [SeeingAI 2020] for blind users with many features including a scene describer. Google Chrome [Chrome 2020, Mazzoni 2019] currently provides word descriptions of graphics as an option to screen reader users. These descriptions are useful for common scenes but not for complex STEM diagrams.

Commercial software for creating these natural language descriptions is under active development, some already on the market. The most prominent is Tableau [Tableau 2020] is a major company with a commitment to accessibility with partners [AutomatedInsights 2020, NarrativeScience 2020, Arria 2020] marketing Tableau plug-ins that create natural language descriptions of data underlying graphical display. However, these applications are data interpreters, not graphics recognizers, and all produce only word descriptions. These are not useful for complex graphics, but words are not really adequate even for simple charts. Graphics convey much subtle but important information, such as the relationship of neighboring points or the trend of data in a bar or line chart. Word descriptions seldom provide the subtle trends or precision of the graphic.

2.2 Tactile graphics

Graphics for textbooks and other published material have traditionally been "made accessible" by providing a tactile copy. Guidelines [BANA 2017] for creating tactile diagrams are published by the Braille Authority of North America. A brief perusal of these guidelines makes it clear that officially-acceptable tactile versions of mainstream graphics can be made only by trained human transcribers and are consequently very expensive. Unfortunately, such “stand-alone” tactile diagrams are seldom useful anyhow. They must have braille labels, but only about 10% of blind people read braille [APH 2016]. Even worse, experts [Dietrich 2013, Hasty 2013] estimate that considerably fewer than 25% of braille readers can read a tactile diagram. Informal polls by the PI [Gardner 2015] of blind students and blind scientists and engineers at meetings of the National Federation of the Blind suggest an even smaller percentage. It is a classic “chicken and egg” dilemma. Few available tactile graphics means few blind people meet them often, hence few blind people can read them.

During his long and highly productive career, University of Washington Prof Richard Ladner (https://www.cs.washington.edu/people/faculty/ladner) developed several innovative applications that would semi-automatically create accessible versions of STEM diagrams. These include one that would parse diagrams in articles or even complete books and create tactile copies of most figures. To the knowledge of the author however, these have never been used outside his lab. This work was done a number of years ago, and the figures do not conform to the more recent BANA standards, so this development may never be used.

2.3 Non-verbal sonification

Gardner 1996, Bargi-Rangin 1997, Brown 2003, Cohen 2006] and many others have used x-y graph tone vs time plots and found them effective for making these graphs accessible. Meijer[Meijer 2020] has generalized this idea to show two-dimensional images as frequency distribution vs time plots that a few people have learned sufficiently to "see" their environment. Sonification works well for single x-y graphs, but Meijer's technique has too great a learning curve to make it practical for general use.

2.4 Keyboard browsing with verbal feedback

[Sorge 2016, Fitzpatrick 2017] investigated audio browsing using keyboard input for navigation. This technique can provide good access to well-structured graphics such as organic molecules and Org charts. A current example is a molecule-browser [Sorge 2015, Sorge 2016] allowing users to explore the structure of arbitrarily complex organic molecules.

3.0 Audio-tactile graphics

The audio-tactile method has been understood for decades [Parke 1988] to provide excellent access to complicated graphics by people with and without visual disabilities. It has a relatively simple and straightforward learning curve and requires no braille ability. But it does require some special hardware. The blind reader can use a smart pen to read some products (best known are the T-Maps made by the California Lighthouse for the Blind - https://lighthouse-sf.org/tmap) or use a touch display to explore tactiles while receiving spoken information.

Spatial exploration with audio feedback even without any tactile overlay can be useful to blind people with good spatial skills. Summers [SAS 2020] has demonstrated that many people can use a standard tablet to understand Well-designed statistical data displays. Haptic feedback [Yu 2003, Gorlewicz 2018] provides more spatial context and consequently better accessibility. The startup company Vital (vital 2020) is developing software to support access using a commercial tablet with haptic feedback. Current haptic technology is still fairly primitive, and much better touch feedback can be obtained with a good tactile overlay.

A number of audio-tactile products have been introduced in the past three decades [Parkes 1988, Loetzsch 1994, Loetzsch 1996, Gardner 2001, Landau 2003]. Only two companies today market audio-touch products using tactile overlays. TouchGraphics (TouchGraphics.com) creates custom products in a proprietary format for libraries, museums, transportation facilities, etc. The other product is ViewPlus IVEO.

The IVEO® technology was introduced in 2005 [Gardner 2005]. It is based on “smart” Scalable Vector Graphic (SVG). SVG is supported by all modern web browsers. Smart SVG contains rich metadata providing the semantics for generating audio explanations on components and their relations. Tactile copies are created from any IVEO SVG image by printing the file to any ViewPlus embosser. It is important to understand that BANA guidelines are intended specifically for stand-alone tactile diagrams. There are no guidelines for audio-tactile diagrams, largely because audio feedback greatly reduces the need for special rules.

ViewPlus' IVEO Reader, and ViewPlus printer drivers have filter options to improve tactile readability and our experience is that good mainstream figures seldom require human intervention to produce good tactile versions.

IVEO® provides several levels of access. If the user is just browsing they might only read the SVG title and possibly the summary description. If reading for content the user can browse through single elements of the SVG using the keyboard. But if detailed understanding of the layout is needed, excellent access can be achieved by the audio-tactile method. Audio-tactile reading requires a tactile replica aligned with the visual image and a touch-sensitive surface that communicates touches to the computer. The image is read by tapping on raised elements or intermediate spaces, to hear titles of objects or text labels. Maps, geographical information and other 2-dimensional displays use color or intensity to convey information. Tone sonification and the dot height/pattern of tactile objects convey this information to blind users.

The author has emphasized [Gardner 2009, Gardner 2012, Gardner 2014] that smart SVG is the only currently-understood graphics access method that can lead to a future in which published graphics can be fully accessible as-published.

4.0 Status of the author’s Web-based Audio-Tactile Graphics reader

The author is not a skilled software developer but can use Javascript. This is sufficient to research effectiveness of various audio, tactile, and braille feedback mechanisms. Instead of the normal model of develop, test, modify, test again etc, as a blind scientist he can integrate these tasks. His many years of involvement in product testing for ViewPlus has given him a broader perspective than his alone on needs of the blind community. So his expectation is that a final commercial release will have fairly similar properties to what he is developing in this research project.

4.1 Requirements for the SVG

Accessibility is most easily achievable when the SVG is well-structured. Major objects (eg the bars in bar charts, the elements of org charts and flow charts) need SVG titles, a simple task if those objects are single SVG elements. In addition, SVG text should be included as semantically-meaningful chunks. Typically, SVG today is created in well-structured form, so graphical objects are well-defined. This is true simply because well-structured SVG is easiest to create and most re-usable. Text is also included as proper SVG, but all too often it is not so well-structured. For example, it is very common for the axis labels on graphs to be included in a single text span, making it difficult for a user to read any label individually.

Most modern software produces reasonably well-structured SVG, because it is the best and most re-usable way to create graphics.

Some SVG graphics include the needed object titles and other required metadata. This is largely done to make these images accessible to search and classification, but some companies also understand how such inclusion can also make them accessible to blind users. For example, the company HighSoft includes much metadata in their charting/map-making applications.

Much of this important metadata is included naturally during the authoring process, but authoring software does not usually save it appropriately in the final graphic. Doing so would not require a serious change in the authoring software, but until/unless such information is required by law, it will be necessary for someone to add this information after-the-fact. Doing so is usually not difficult, and ViewPlus already has software available that can “make graphics accessible” by inserting such data.

4.2 The web-based Audio-Tactile Reader Application

The purpose of this application is to give a blind user all the software tools needed to read a decently-made SVG. Currently it is designed to work on all common platforms except iOS – which requires integrating Voiceover for speech. The current app relies instead on the HTML web speech API. The app has similar behavior in Windows and the Chrome OS and should work well with Linux. However it has not yet been tested with Linux. It works well with Chrome, Firefox, and Brave browsers and most functionality works in Edge. Edge does not properly support pointer move functionality and so sonification and drawing features do not work. This is a strange bug that just needs to be fixed in Edge.

Sighted people can hear audio information by clicking a mouse or pointer on the object or text of interest. Blind users can do the same if a tactile copy is overlaid and the visual image calibrated to align.

ViewPlus markets an external USB touchpad that was needed when computers did not have touch screens. This device is bulky, fragile, expensive, and not easily portable. This has been a major factor inhibiting usability of IVEO. Today a better choice for audio-tactile users is to have a laptop that is also an audio-tactile access device. A “2 in 1” laptop computer/tablet has a touch screen, and a tactile version of an SVG can be clipped to the screen and read. 2 in 1 computers cost little more than those with a conventional screen.

Windows cannot be configured to fully eliminate all of its “convenience” touch screen features such as edge swiping ttask bars and various other dialogs. These are annoyances, not conveniences for blind users. Nonetheless, a carefully-used Windows touch screen is quite usable for audio-touch purposes.

Major Features:

- A tactile image can be created simply by printing to any ViewPlus embosser.
- Screen images can be aligned quickly by touching two points on the tactile image.
- If an object has a title, it is read when the object is clicked.
- If the object has a description it is read by holding or double tapping.
- The SVG title and description are accessible through hot keys.
- All audio information is displayed in a info bar and can be browsed. The info bar can be displayed in an on-line braille display.
- Find object and find text features permit these to be found by directing the user to go right/left/up/down.
- Objects or selectable rectangles can be zoomed to full screen. These “views” can be saved and embossed.
- The SVG and all additional views are assigned a unique id that is embossed as a special “tactile bar code” allowing the image to be recalled later from the tactile copy.
- GIS images, other 2d data, bar, pie, and x-y graphs can be sonified.

4.3 Authoring graphs, geometric figures and other useful STEM images

A currently unmet need is for software permitting blind students to create mainstream-accessible graphs and other images. The author has branched the audio-tactile software to permit blind people to create x-y graphs of any function or set of functions that can be written in “calculator” notation. The resulting graphs are normal SVG images that can be embossed and read by the blind authorIn “sonification” mode, a tone plot of the x-y line can be played as a function of time. Or the line can be heard by running a finger along it. The latter feature is particularly useful for distinguishing which of several lines is the one of interest in multiple-line graphs.

A blind person can also create common geometric shapes – triangles, circles, ellipses, polygons of virtually any degree. The user can simply define the figure by inputting parameters from the keyboard or by touching the screen to show figure centers, radii, vertices, etc. Several convenient sonification options are available to aid the blind author. The most important is ability to find, feel, and identify figures after they have been created without having to repeatedly emboss.

This application has been requested by many VI teachers for students taking courses in STEM fields.

4.4 Displaying math equations

The author’s first contribution to math accessibility was DotsPlus[Gardner 1998], a combination of graphics and braille that reproduces the 2d physical layout of math equations. DotsPlus has limited use, but some braille readers find it to be a more accessible way to read math than having to learn another math code. It is also useful simply as a demonstration to congenitally blind people of the layout of conventional math.

DotsPlus currently can be produced by using a special ViewPlus-created format in the popular MathType application. It should not be difficult to make it an option in MathJax, but this is not something that this author has the skill to do.

4.5 Availability

At the time of writing, the author anticipates opening an alpha/usability test of both the reading and authoring applications soon. If sufficient interest is generated, commercial products could be developed and introduced within a year.

References

[APH 2016] American Printing House for the Blind, "Annual Report 2016: Distribution of Eligible Students Based on the Federal Quota Census of January 3, 2015 (Fiscal Year 2016)." Retrieved December 22, 2019 from http://www.aph.org/federal-quota/distribution-of-students-2016

[Arria 2020] "the Natural Language Insights extension" retrieved December 22, 2020 https://extensiongallery.tableau.com/products/47

[AutomatedInsights 2020] "Natural Language Processing Plug-in for Tableau", retrieved December 22, 2020 https://automatedinsights.com/partners/tableau/

[BANA 2017] Guidelines for Tactile Graphics" retrieved December 22, 2020 from "http://www.brailleauthority.org/tg/"

[Bargi-Rangin 1997] H. Bargi-Rangin, S. Sahyun, R. Lundquist, and J. Gardner, "TRIANGLE: Tri-Modal Access to Text, Tables, Equations, Graphs, Calculations and Figures," Proceedings of the 1997 International Conference on Technology and Persons with Disabilities, Los Angeles, March 1997

[Brown 2003] L. Brown, S. Brewster, S. Ramloll, R. Burton, and B. Riedel (2003). Design guidelines for audio presentation of graphs and tables. Proc. on Auditory Display.

[Chrome 2020] "Get Image Descriptions from Google", a guide for blind users retrieved December 22, 2020 https://support.google.com/chrome/answer/9311597?hl=en

[Cohen 2006] R. Cohen, A. Meacham, and J. Skaff (2006). Teaching graphs to visually impaired students using an active auditory interface. ACM SIGCSE Bulletin, 38(1):279-282.

[Dietrich 2013] Ms Gaeir Dietrich, private communication, Nov 27, 2013, confirming that non-braille readers and even most braille readers cannot understand tactile graphics. Ms Dietrich is a staff member in the High Tech Center Training Unit of the California Community Colleges and considered an expert on training blind students.

[Gardner 1996] J. A. Gardner, R. Lundquist, and S. Sahyun,"TRIANGLE: A Practical Application of Non-Speech Audio for Imparting Information", Proceedings of ICAD 96 International Conference on Auditory Display, San Francisco, CA, November, 1996

[Gardner 1998] John A. Gardner, "The DotsPlus Tactile Font Set", Journal of Visual Impairment and Blindness, December, 1998, pp. 836-840

[Gardner 2001] J. Gardner, & V. Bulatov, (2001). Smart Figures, Svg, And Accessible Web Graphics. Proceedings of the 2001 CSUN International Conference on Technology and Persons with Disabilities. Los Angeles, CA

[Gardner 2005] John A. Gardner, Vladimir Bulatov, and Holly Stowell, "The ViewPlus IVEO Technology for Universally Usable Graphical Information" Proceedings of the 2005 CSUN International Conference on Technology and People with Disabilities, Los Angeles, CA, 16-19 March, 2005 http://www.viewplus.com/about/abstracts/05csungardner2.html

[Gardner 2009] John A. Gardner, Vladimir Bulatov, and Robert A. Kelly, Making Journals Accessible to the Visually Impaired – The Future is Near. Learned Publishing 22(4) 2009 pp. 314-319. http://www.viewplus.com/about/abstracts/09learnpubgardner.html

[Gardner 2012] J. Gardner, “"Creating the Future", Outstanding Achievement Award Lecture, ASSETS, Boulder, CO, (Oct 22-24, 2012)

[Gardner 2014] J. Gardner. Can Mainstream Graphics be Accessible? Information Technology and Disabilities (an on-line Journal), V 14, No 1, March 2014 http://itd.athenpro.org/volume14/number1/gardner.html

[Gardner 2015] The PI conducted impromptu surveys while addressing the Student Division (on July 6) and the Science and Engineering Division (on July 7) of the National Federation of the Blind convention, Orlando, Florida, July 5-10, 2015. A sighted observer reported that fewer than 15 of the approximately 200 students and fewer than 25% of the Science and Engineering participants agreed that they were comfortable reading tactile graphics. The students screamed a thunderous “yes” when asked if they needed better graphics access.

[Gorlewicz 2018] Jenna Gorlewicz, Corrine Mueller and Hayden Dahmm, " Bringing Digital Graphics to Life: The Promise of Multimodal Touchscreens—Progress and Future Work", presented at Tactile Graphics in Education and Careers, National Federation of the Blind Jernigan Institute, Baltimore, MD, October 11–12, 2018

[Hasty 2013] Ms Lucia Hasty, private communication, Nov 27, 2013, estimating that fewer than 25% of braille readers could understand tactile graphics. Ms. Hasty is considered the foremost US expert on tactile graphics. She was the leader of theBraille Authority of North America committee that developed the tactile graphics guidelines published at http://www.brailleauthority.org/tg/

[Landau 2003] S. Landau, (2003). Use of the Talking Tactile Tablet in Mathematics Testing. Journal of Visual Impairment and Blindness, 97, 85–96.

[Loetzsch 1994] J. Loetzsch, (1994). Computer-Aided Access to Tactile Graphics for the Blind. Proceedings of the 4th International Conference on Computers Helping People, Vienna. 575–581.

[Mazzoni 2019] Mazzoni, Dominic , "Using AI to give people who are blind the “full picture”" retrieved December 22, 2019 https://www.blog.google/outreach-initiatives/accessibility/get-image-descriptions/

[Meijer 2020] Meijer, Peter, "Hearing a Printed Graph", retrieved December 22, 2020 https://www.seeingwithsound.com/prgraph.htm

[NarrativeScience 2020] "Narratives for Tableau", retrieved December 22, 2020 https://narrativescience.com/products/quill/extensions/tableau/

[Parkes 1988] D. Parkes, Nomad: an Audio-Tactile Tool for the Acquisition, Use and Management of Spatially Distributed Information by Partially Sighted and Blind Persons. Proceedings of the Second International Symposium on Maps and Graphics for Visually Handicapped People. King's College, University of London. 24 –29.

[Parkes 1991] D. Parkes, "Nomad: Enabling Access to Graphics and Text Based Information for Blind, Visually Impaired and Other Disability Groups", Conference Proceedings, Vol. 5. World congress on Technology 1991, Arlington, Virginia, pp. 690-714

[SAS 2020] "Accessibility at SAS", retrieved December 22, 2020 https://www.sas.com/en_us/company-information/accessibility.html

[SeeingAI 2020] "SeeingAI| Camera App for those with a visual impairment", retrieved December 2020 https://www.microsoft.com/en-us/ai/seeing-ai

[Sorge 2014] V. Sorge, C. Chen, T.V. Raman, and D. Tseng (2014). Towards making mathematics a first class citizen in general screen readers. Proc. of W4A. ACM.

[Sorge 2015] V. Sorge, M. Lee, and S. Wilkinson (2015). End-to-end solution for accessible chemical diagrams. In Proc. of W4A. ACM. Best Paper Award

[Tableau 2020] " Best practices for designing accessible views" retrieved December 22, 2020 https://help.tableau.com/current/pro/desktop/en-us/accessibility_best_practice.htm

[Vital 2020] " Equalizing Education through Accessible Experiences" retrieved December 2020 https://www.vital.education/

[Yu 2003] W. Yu, K. Kangas, and S. Brewster (2003). Web-based haptic applications for blind people to create virtual graphs. Proc. of HAPTICS, 318-325. IEEE.

An Object-oriented Graphic Description Language Available for Blind Users

Noboru Takagi[1], Takashi Suzuki[2], and Tomoyuki Araki[3]

[1] Toyama Prefectural University
Imizu, Toyama, Japan
takagi@pu-toyama.ac.jp
[2] Hiroshima Institute of Technology
Hiroshima, Japan
t.suzuki.rb@it-hiroshima.ac.jp
[3] Hiroshima Institute of Technolgoy
Hiroshima, Japan
t-araki@cc.it-hiroshima.ac.jp

Abstract

With the development of information processing systems such as screen readers, the accessibility of various information to visually impaired people has been dramatically improved in recent years. In addition, InftyReader, an OCR system for mathematical expression recognition, is making it easier to access scientific documents with mathematical expressions. However, there are still large barriers for blind people to access visual information such as graphs and diagrams. In particular, it is almost impossible for blind people to generate precise figures without the assistance of sighted people. Therefore, we are developing a graphic description language available for the blind. In conventional graphic description languages, it was necessary to accurately specify the coordinates of parameters when drawing elementary shapes. By introducing an object-oriented idea, our language enables users not to specify many coordinates. We have produced an experimental drawing assistant system using our language. In this paper, we outline our language and system, and show the results of experiments which were conducted to verify the effectiveness of our system.

1 Introduction

Visually impaired people often use tactile graphics when they access figures. Here, a tactile graphic is an image that use raised surface so that a visually impaired person can feel the graphic. However, they are not able to generate precise figures without supports by sighted people. Several systems have been developed in order for the visually impaired to draw figures independently. The systems [1, 2, 3] are WYSIWYG (What You See Is What You Get) type computer-aided systems using refreshable matrix tactile displays. However, this kind of systems do not make us draw figures precisely; it only enables us to draw figures roughly. The system by Fujiyoshi [4] has adopted a command input method, and therefore it makes users draw precise figures. However, because the grammar of the commands is not rich so that we need a long time to complete drawing even a simple figure. So, we still do not have any drawing assistant system that is easy to access for visually impaired people.

One of the authors is a blind physics teacher of a college. When he creates documents for his lecture, he often uses TikZ when drawing figures. TikZ is a language for producing vector graphics from a geometric or algebraic description, and it involves high-level functions to draw figures. So these functions enable visually impaired people to draw precise figures, if they learn the grammar of TikZ and have rich experience for TikZ. The grammar of TikZ is not premised on the use of visually impaired people, so it is necessary to calculate every parameter for drawing elementary shapes such as lines and circles etc., and the barrier of TikZ is still high for visually impaired users.

Base on the background above, we are developing a drawing assistant system available for the visually impaired. A characteristic feature of our system is to introduce a new graphic description language which is desinged so that visually impaired users are easy to access. We have developed an experimental lexical analyzer and parser for our language. Then, an experimental drawing assistant system has also been developed. In the drawing assistant system, a user first writes a code for a figure, then the user can check the figure by touching the tactile graphic displayed on a refreshable matrix display.

In this paper, we explain the graphic description language we have developed, and then the outline of our drawing assistant system and the effectiveness for our system is described.

2 Related Works

A blind co-author uses TeX to produce documents for his physics lectures and so he uses TikZ to draw figures because it is a graphics package of TeX. Since TikZ has advanced graphic functions, even a blind user can draw figures if the user learns the grammar of TikZ and has a rich experience for TikZ. However, the grammar of TikZ was not designed under the expectation of using by blind users, and so there exists still a high barrier for blind users. There are languages other than TikZ that describe graphics such as PostScript and Scalable Vector Graphics etc., but none of them was also designed under the expectation of using by blind users.

Conventional tactile graphics production software requires users to operate GUIs, and so blind users cannot use them. Edel [5] and Braille editing system [7] etc. exits as drawing software for sighted users. Also, Tiger Software Sweet [8] is a high-performance system that automatically analyzes bitmap images and create tactile graphics, but none of them can be used by blind users.

Refreshable tactile displays such as DV2 [6], Tactile2D etc. [9], and Graphiti [10] that presents tactile graphics by controlling the vatical movement of pins arranged on a matrix using piezoelectric elements are on the market. Using these devices, several WYSIWYG (What You See Is What You Get) type drawing systems have been developed for blind users [1, 2, 3]. However, the refreshable tactile graphics displays cannot present precise figures, and so blind users can only feel figures roughly. Hence the WYSIWYG type systems are not appropriate for creating documents that are shared with sighted students.

In addition to the systems mentioned before, a system called Bplot [4], which allows blind users to edit figures by writing a set of commnands using a text editor, has been developed. Here, Bplot adopts a plotter control command method for controlling the braille printers ESA721 and ESA600G, products of JTR Corporation in Japan. Bplot enables blind users to draw high-precision figures, but there is still a high barrier; Fujiyoshi [4] reported that even when drawing a simple flowchart, it took about 90 minutes on average to complete drawing the flowchart.

3 Object-oriented Graphic Description Language

When a blind teacher draws a precise figure that is presented to sighted students, we think that it is better to choose a graphic description language such as TikZ and Bplot rather than the WYSIWYG type drawing systems. However, in the current graphic description languages, there are some drawback which can be barriers for blind users.

1. When drawing an elementary shape, all its parameters should be calculated.
2. After an elementary shape has been drawn, it is impossible to edit the shape again, such as translation and rotation etc.
3. When getting features, such as endpoints etc., of an elementary shape, all of the coordinates or values should be calculated.

Therefore, we have decided to develop an object-oriented graphic description language that blind users are easy to access.

Since the drawing functions of TikZ are rich, we introduce some of the functions from TikZ.

1. Both of the absolute and relative coordinate systems are used. In the relative coordinate system, a coordinate is defined using a coordinate system whose origin is the coordinate used immediately before.
2. Both of the Cartesian and polar coordinate systems are used.

A part of the grammar of our language is explained. At the current version of our language, the following elementary shapes can be drawn: (1) straight/arrowed lines; (2) rectangles; (3) circles; (4) ellipses, and (5) circular/ellipse arcs. Further, the following edit operations can be used: (1) translation; (2) rotation; (3) grouping, (4) ungrouping, and (5) deletion. The grammar for straight/arrowed lines is below.

Format: line(*start_point*, *end_point*) [arrow_type = *val1*, rotation = *val2*];

Description: This command generates a straight/arrowed line whose start and end points are *start_point* and *end_point*. The items in the brackets are optional and can be omitted. The arrow can be set with the option *arrow_type*, and lines can be rotated by setting a value to *rotation*.

Member value: there are three member values: start_point; end_point; and angle. start_point is the start point of this line object, end_point is the end point of this line object, and angle is the angle between the x-axis and this line object.

Member function: there is only one member function, get_point(*val*). The parameter *val* is a real number between 0 and 1. This member function returns the point on this line located at $val \times length$ from start_point. Here, *length* is the length of this line.

A sample code of our language is shown below. The numbers and colons indicate the line numbers, they are not part of the code.

1: line1 = line((-15,0), (15,0))[arrow_type = ->];

2: rect1 = rectangle((0,0), (5,3));

3: shift(rect1, -8);

4: line2 = line(rect1.east, +(60:12))[arrow_type = ->];

5: line3 = line(rect1.east, (line2.start_ponit -— line2.end_point));

6: arc1 = arc(rect1.east, 0, 60, 3);

7: line4 = line(line1.get_point(0.1), +(-90:5));

8: line5 = line(line1.get_point(0.9), +(-90:5));

9: line6 = line(line4.get_point(0.5), line5.get_point(0.5))[arrow_type = ->];

We explain the meaning of the sample code.

1: Generate a right arrow line whose start and end points are (-15,0) and (15,0), and save it as line1.

2: Generate a rectangle with points (0,0) and (5,3) as diagonal vertices, and save it as rect1.

3: Translate rect1 in the x-axis direction by -8.

4: Generate a right arrow line whose start and end points are rect1.east and +(60:12), and save it as line2. Here, rect1.east is the point on the center of the right-side edge of rect1, and +(60:12) is a point located at azimuth 60 degrees and length 12 with rect1.east as the origin.

5: Generate a line whose start and end points are rect1.east and (line2.start_point -— line2.end_point), and save it as line3. Here, the point (line2.start_point -— line2.end_point) represents the intersection when point line2.start_point is extended in the x-axis direction and point line2.end_point is extended in the y-axis direction.

6: Generate a circular arc whose center is rect1.east, the start azimuth is 0 degree, the end azimuth is 60 degrees, and radius is 3. Then, save it as arc1.

7: Generate a line whose start and end points are line1.get_point(0.1) and +(90:5), and save it as line4. Here, line1.get_point(0.1) represents a point that internally divides line1 into 1 : 9 from its start point; and +(90:5) represents a point located at azimuth 90 degrees and length 5 with line1.get_point(0.1) as the origin.

The explanations for the lines 8 and 9 are omitted.

We have developed an experimental lexical analyzer and parser for translating codes written by our language. We used Visual C++ to develop them. A bitmap image is generated using OpenCV after parsing the input code. Figure 1 show the output image of the above sample code.

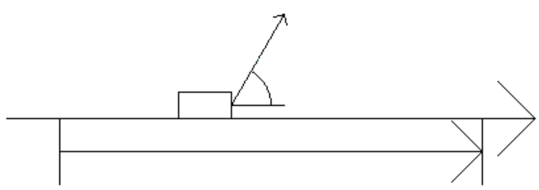


Figure 1: A Bitmap Image for the Sample Code

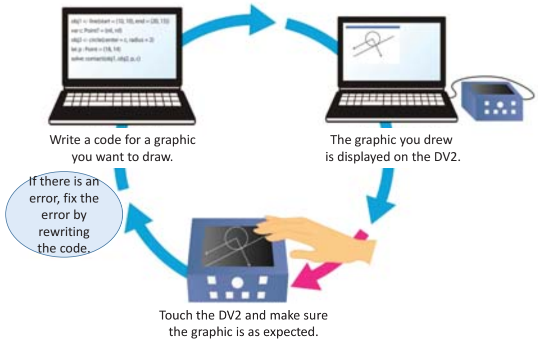


Figure 2: Outline of the Drawing Assistant System

3.1 Drawing Assistant System Available for Blind Users

Figure 2 illustrates the outline of our drawing assistant systems.

The system configuration is as follows: (1) Personal Computer; (2) Refreshable matrix tactile display DV2, product of KGS Co.; (3) Graphics tablet, intuos pen and touch, product of Wacom Co.; (4) Slide table; and (5) Display.

There are 48 pins horizontally and 32 pins vertically on the surface of DV2 which are arranged in a matrix. The dimeter of each pin is 1.2 mm. Each pin is driven up and down by a piezoelectric element. The space between each pair of adjacent pins is 2.4 mm, and so the surface of DV2 is not large enough to display a precise graphic. For this reason, the DV2 was set up on a slide table so that it could be slided manually up, down, left, and right. The slide table can move twice the vertical length of the surface in the vertical direction, and also move twice as long in the horizontal direction. As a result, the area of the surface is virtually quadrupled. The position of the DV2 is measured using a pen tablet.

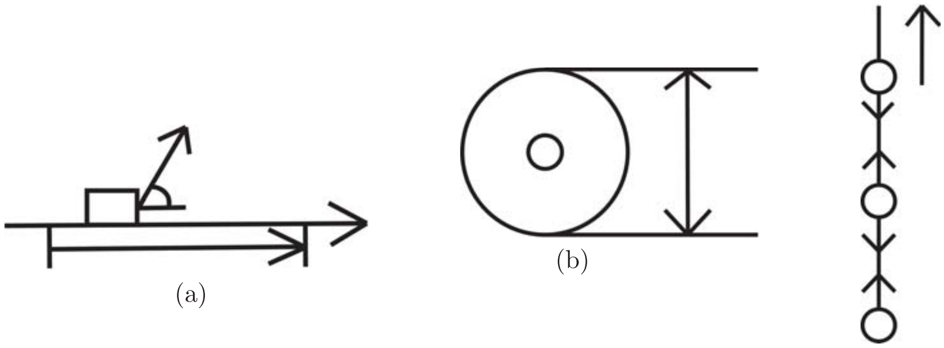
The PC is equipped with the lexical analyzer, parser, and bitmap image generator in addition to a GUI and DV2 control system, and then controls the drawing assistant system. The GUI and DV2 control system were designed using Visual C#. For debugging, the bitmap image generated is displayed on the display, this function is realized using OpenCvSharp library.

3.2 Experiments and Discussions

Two experiments were conducted to examine the effectiveness of our drawing assistant system. This section describes the outline and results of the experiments, and gives some discussion. In both the experiments, the subject was a single blind physics teacher. This blind teacher has been using TikZ for about 18 months to create documents for his lecture. Therefore, in the first experiment, we examine how much our graphic description language can easily generate figures compared to TikZ. In the second experiment, we examine whether our drawing assistant system makes the blind subject possible to draw figures independently.

3.3 Experiment 1: Comparing with TikZ

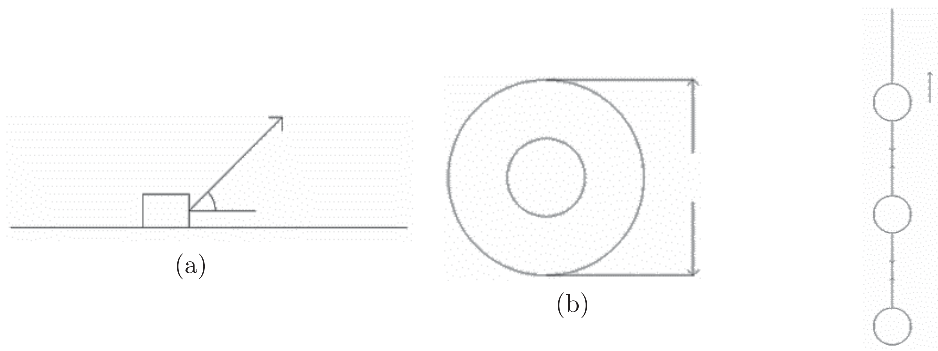
The blind subject was presented the grammar text of our graphic description language, five sample codes and their bitmap images in advance. According to the subject's self-report, he learned our graphic description language for about five hours in total. As mentioned before, this subject has been using TikZ for about 18 months to create figures of documents to be shared with his students. We asked the subject to create the three figures shown in Figure 3 using TikZ and our language. The completion time for the drawing and the number of compilations for the codes were measured as objective evaluation. An interview was conducted and used as subjective evaluation.



(a) (b) (c)

Figure 3: Figures in Experiment 1

For each of the figures in Figure 3, before writting its code, the subject understood the content of the figure by touching its tactile graphic, during touching we explained verbally him the content of the figure. After that, he wrote a code for the figure using a text editor, compiled the code, and generated a bitmap image. The bitmap image was displayed on the screen. If there were errors in the drawing, we explained the errors him verbally and asked to correct the errors. This process was continued until the subject completes the figure. Figure 4 shows the images generated using our language, and Figure 5 are the images using TikZ. Table 1 shows the times for completing the figures and the number of compilations.



(a) (b) (c)

Figure 4: Images Edited Using Our Language

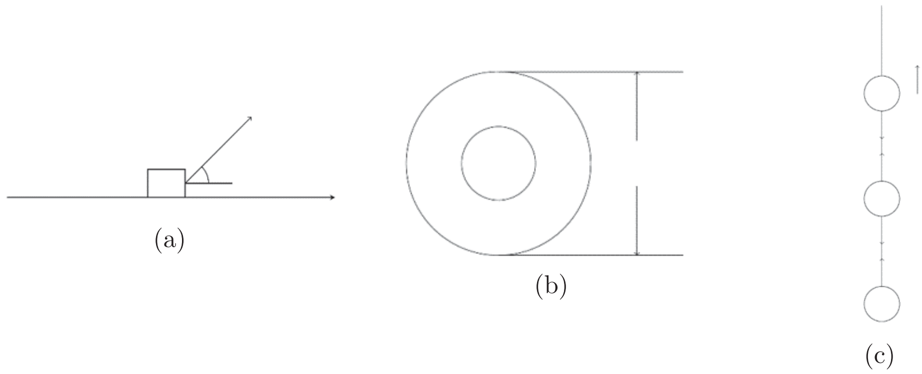


Figure 5: Images Edited Using TikZ

Table 1: Comparisons

Fig.	Time (sec) TikZ	Time (sec) Our	Compilations TikZ	Compilations Our
(a)	260	290	1	2
(b)	253	379	1	1
(c)	641	722	2	1

3.4 Experiment 2: Evaluation for Drawing Support System

The tactile graphic of the figure shown in Figure 6 was presented to the subject and we explained the contents of the figure verbally to the subject. After the subject confirmed the contents of this figure, we asked him to generate the figure using our drawing support system described in Section **??**. In this experiment, the drawing result were confirmed by the subject alone using the DV2 without any advice. Figure 7 shows the result created by the subject. Ths subject took about 21 minutes to complete the figure.

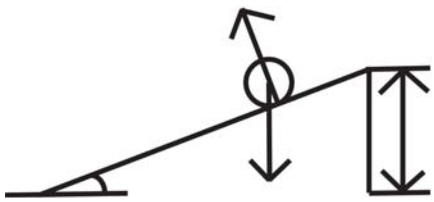


Figure 6: Figure for Experiment 2

3.5 Experimental Results and Discussions

First, the results for Experiment 1 are described. Table 1 shows the time to complete each figure and the number of compilations. From Table 1, almost no difference was observed in the number of compilations, but TikZ needed less time to complete figures. Although the learning time for our language was much less than that for TikZ, we did not observe large difference between our

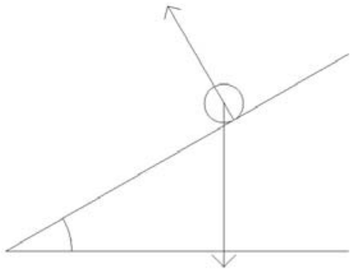


Figure 7: Result for Experiment 2

language and TikZ. Hence, this suggests that drawing using our language is considered to be easy for visually impaired users.

Next, we explain the results for Experiment 2. Figure 7 shows the result generated by the subject. Although the downward arrow from the center of the circle is longer than the original arrow, the subject was almost able to draw the figure similar to the original figure. The time to complete the drawing was about 21 minutes. The main reason why it took much longer time to complete the drawing compared to Experiment 1, is due to the operation for DV2; this subject did not often use tactile graphics and DV2, so he needed to spend a lot of time to feel tactile graphics displayed on DV2. We had the following comment from the subject. Although he was not able to generate perfect figures alone, it would be possible to create figures independently using this drawing support system if the further improvements have been done.

3.6 Conclusions

We have developed an experimental graphic description language that could be used by blind users. One of the characteristic features of our language is that, after defining a basic shape such as a straight line, by saving it as an object and attaching its features to the object, it enables drawing easily even for blind users. We have developed an experimental drawing assistant system using our language. The two experiments were conducted to verify the effectiveness of our language and the drawing assistant system. The results suggest that although our language is not grammatically and functionally rich yet, it can be easily drawn figures by the blind subject. Further, although there are still some drawbacks for the drawing assistant system, it is suggested that even blind users can draw figures independently.

4 Acknowledgments

This work was supported by JSPS Grant-in-Aid for Scientific Research (B) Number 15H02796.

References

[1] T. Watanabe and M. Kobayashi, "A Prototype of the Freely Rewritable Tactile Drawing System for Persons Who Are Blind", *Jounal of Visual Impairment and Blindness*, vol. 96, issue 6, pp. 460-464, 2002.

[2] N. Takagi et al., "Consideration of the Experience of a Blind User Using a Tactile Graphics Editor Available for Bline People", *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetic*, 2016.

[3] J. Bornschein et al., "Comparing Computer-Based Drawing Methods for Blind People with Real-Time Tactile Feedback", *Proceedings of the ACM CHI Conference on Human Factors in Computer Systems*, 2018.

[4] M. Fujiyoshi et al., "A Tactile Graphics Production System Bplot with a New Command-line Version to Enable Teachers and Staff with Severely Visually Disabilities on Their Own to Produce Braille Teaching Materials", *Jounal of Japan Society for Education Technology*, vol. 41, no. 2, pp. 149-156, 2017. (in Japanese)

[5] Edel, http://www7a.biglobe.ne.jp/ EDEL-puls/ (accessed on February 21, 2020) (in Japanese)

[6] KGS Corporation, https://www.kgs-jpn.co.jp/

[7] Braille Editing System, http://www.ttools.co.jp/product/eyes/BES/ (accessed on February 21, 2020) (in Japanese)

[8] Tiger Software Suite, https://viewplus.com/product/tiger-software-suite/ (accessed on February 21, 2020)

[9] Metec AG, https://www.metec-ag.de/produkte-graphik-display.php/ (accessed on February 21, 2020)

[10] Graphiti, https://www.orbitresearch.com/product/graphiti/ (accessed on February 21, 2020)

A remark on the human readable TeX

Hitoshi Tanaka*

Research and Support Center on Higher Education for the hearing and Visually Impaired, National University Corporation Tsukuba University of Technology, Kasuga 4-12-7, Tsukuba 305-8521, Japan
htanaka@k.tsukuba-tech.ac.jp

1 Introduction and discussion

In this paper we will give a remark on the human readable TeX, which is produced by InftyReader, with respect to the use of the screen reader NVDA.

1.1

InftyReader is OCR software to recognize scientific documents including mathematical formulae, and to output the recognition results into various file formats: LaTeX, MathML, XHTML, HRTeX, IML and Microsoft Word document. It is developed in the laboratory of M. Suzuki, Faculty of Mathematics, Kyushu University, in collaboration with several cooperation partners.

HRTeX is also called “human readable TeX” and it is a laTeX source file which deleteed all dollar marks and some notnecessary layout infomations. we always read the mathematical documentations by converting their to the human readable TeX.

1.2

NVDA is a free, open source, globally accessible screen reader for the blind and vision impaired. Blind inventors make computers accessible for the visually challenged. NVDA has been translated into 27 languages, thanks to volunteer translators. Especially, NVDA can output UEB braille codes.

1.3

Unified English Braille (UEB) is a complete code. It is a code that is used for both literary (non-technical) and STEM (technical) materials. A general principle of UEB is that each print symbol has one and only one braille equivalent. UEB follows this principle by using the same physical symbol regardless of context (literary or technical). Technical materials in UEB follow the same rules and use the same symbols as non-technical materials. For the most part, UEB follows print for spacing of symbols. Consistency in spacing should be maintained, particularly when spacing in print is inconsistent. When spacing is used merely for style with technical materials in print, spacing in UEB should reflect the structure of mathematics. A braille symbol may have a grade 1 meaning and a contracted (grade 2) meaning. Some symbols also have a numeric meaning. The symbol position in relation to other symbols and the mode in effect will determine whether the symbol is read as a grade 1, contracted, or numeric meaning.

*The author is supported by Grant-in-Aid for Scientific Research (C) (19K03510), the Japan Society for the Promotion of Science.

1.4

Discussion. Of course, the set of the mathematical symbols is a subset of the symbols which are enumerated by the unicodes. Thanks to the action of the screen reader NVDA, some usual used mathematical symbols can be expressed by the UEB braille cords. I and some blind mathematicians have read the mathematical documentations by converting their to the human readable TeX. Thus, if some LaTeX commands in human readable TeX are transcribed by unicodes, then we are happy to read the mathematical symbols as UEB braille codes.

The following is a samples list of mathematical symbol, LaTeX command, abbreviation, which I always use, and UEB braille symbol.

1.5

List.

α, `alpha`, `al`, ⠨⠁

β, `beta`, `bt`, ⠨⠃

γ, `gamma`, `gm`, ⠨⠛

δ, `delta`, `dl`, ⠨⠙

λ, `lambda`, `lm`, ⠨⠇

σ, `sigma`, `sg`, ⠨⠎

ω, `omega`, `om`, ⠨⠺

Δ, `Delta`, `Dl`, ⠠⠨⠙

Φ, `Phi`, `Phi`, ⠠⠨⠋

Σ, `Sigma`, `Sg`, ⠠⠨⠎

Ω, `Omega`, `Om`, ⠠⠨⠽

$\pm$, `pm`, `pm`, ⠸⠖

$\mp$, `mp`, `mp`, ⠸⠤

$\cap$, `cap`, `cap`, ⠨⠦

$\cup$, `cup`, `cup`, ⠨⠖

$\leq$, `leq`, `leq`, ⠸⠈⠣

$\geq$, `geq`, `geq`, ⠸⠈⠜

$\perp$, `perp`, `perp`, ⠼⠤

$\|$, `parallel`, `parallel`, ⠼⠇

$\subset$, `subset`, `subset`, ⠘⠣

$\supset$, `supset`, `supset`, ⠘⠜

$\subseteq$, `subseteq`, `subseteq`, ⠸⠘⠣

$\supseteq$, `supseteq`, `supseteq`, ⠸⠘⠜

$\in$, `in`, `in`, ⠘⠑

$\ni$, `ni`, `ni`, ⠈⠘⠑

$\emptyset$, `emptyset`, `0`, ⠈⠚

∂, `partial`, `6`, ⠈⠙

∞, `infty`, `8`, ⠼⠿

$\mathbb{C}$, `mathbb{C}`, `C`, –

$\mathbb{N}$, `mathbb{N}`, `N`, –

$\mathbb{R}$, `mathbb{R}`, `R`, –

$\mathbb{Z}$, `mathbb{Z}`, `Z`, –

$\mathcal{B}$, `{mathcal B}`, `cB`, –

$\mathcal{D}$, `{mathcal D}`, `cD`, –

2 Appendix

As an appendix we give a discussion concerning to Braille's table and braille table of the Japanese syllabary.

Braille is a tactile method of reading and writing for blind people developed by Louis Braille (1809–1852), a blind Frenchman. The braille system uses six raised dots in a systematic arrangement with two columns of three dots, known as a braille cell. By convention, the dots in the left column are numbered 1, 2 and 3 from top to bottom and the dots in the right column are numbered 4, 5 and 6 from top to bottom.

The six dots of the braille cell are configured in 64 possible combinations (including the space which has no dots present). The 63 braille characters with dots are grouped in a table of seven lines.

(Famous Braille's table)

⠁	⠃	⠉	⠙	⠑	⠋	⠛	⠓	⠊	⠚
⠅	⠇	⠍	⠝	⠕	⠏	⠟	⠗	⠎	⠞
⠥	⠧	⠭	⠽	⠵	⠯	⠿	⠷	⠮	⠾
⠡	⠣	⠩	⠹	⠱	⠫	⠻	⠳	⠪	⠺
⠂	⠆	⠒	⠲	⠢	⠖	⠶	⠦	⠔	⠴
⠌	⠬	⠼	⠜	⠄	⠤	-	-	-	-
⠈	⠘	⠸	⠐	⠨	⠰	⠠	-	-	-

The Japanese braille system is based on a tableof theJapanese syllabary (the kana syllabary) which developed by Kuraji Ishikawa in 1890.

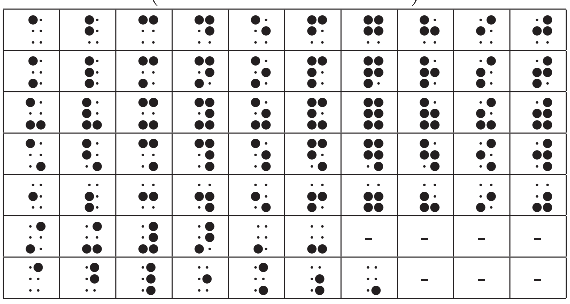
3 Appendix

As an appendix we give a discussion concerning to Braille's table and braille table of the Japanese syllabary.

Braille is a tactile method of reading and writing for blind people developed by Louis Braille (1809–1852), a blind Frenchman. The braille system uses six raised dots in a systematic arrangement with two columns of three dots, known as a braille cell. By convention, the dots in the left column are numbered 1, 2 and 3 from top to bottom and the dots in the right column are numbered 4, 5 and 6 from top to bottom.

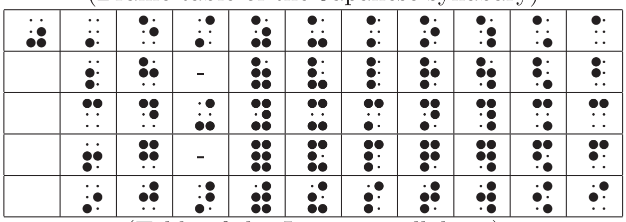
The six dots of the braille cell are configured in 64 possible combinations (including the space which has no dots present). The 63 braille characters with dots are grouped in a table of seven lines.

(Famous Braille's table)



The Japanese braille system is based on a tableof theJapanese syllabary (the kana syllabary) which developed by Kuraji Ishikawa in 1890.

(Braille table of the Japanese syllabary)



(Table of the Japanese syllabary)

nn	wa	ra	ya	ma	ha	na	ta	sa	ka	a
	wi	ri	-	mi	hi	ni	ti	si	ki	i
	wu	ru	yu	mu	hu	nu	tu	su	ku	u
	we	re	-	me	he	ne	te	se	ke	e
	wo	ro	yo	mo	ho	no	to	so	ko	o

We now rearrange the Japanese syllabary as follows:

(A rearrangement of the Japanese syllabary)

a	i	u	e	o	ra	ri	ru	re	ro
ka	ki	ku	ke	ko	sa	si	su	se	so
na	ni	nu	ne	no	ta	ti	tu	te	to
ha	hi	hu	he	ho	ma	mi	mu	me	mo

Compare the following two tables.

(A rearrangement of the Japanese syllabary)

⠁	⠃	⠉	⠋	⠊	⠑	⠓	⠙	⠛	⠚
⠡	⠣	⠩	⠫	⠪	⠱	⠳	⠹	⠻	⠺
⠅	⠇	⠍	⠏	⠎	⠕	⠗	⠝	⠟	⠞
⠥	⠧	⠭	⠯	⠮	⠵	⠷	⠽	⠿	⠾

(Braille's table)

⠁	⠃	⠉	⠙	⠑	⠋	⠛	⠓	⠊	⠚
⠅	⠇	⠍	⠝	⠕	⠏	⠟	⠗	⠎	⠞
⠥	⠧	⠭	⠽	⠵	⠯	⠿	⠷	⠮	⠾
⠡	⠣	⠩	⠹	⠱	⠫	⠻	⠳	⠪	⠺

That is, the braille table of the Japanese syllabary was produced by commuting rows and columns of Braille's table.

Rule based Mathematics and Layout Analysis of PDF Documents

Volker Sorge[1], M Balakrishnan[2], Neha Jadhav[2], Akashdeep Bansal[2], and Himanshu Garg[2]

[1] University of Birmigham, UK `v.sorge@cs.bham.ac.uk`
[2] IIT Delhi, India `mbala@cse.iitd.ac.in` `nehamjadhav@gmail.com` `akashdeep@cse.iitd.ac.in` `himanshu.garg@cse.iitd.ac.in`

1 Introduction

PDF is still one of the most widely used formats for production and distribution of digital content. Although standards like PDF/UA exist, inaccessible PDF documents continue to be the norm. In particular complex PDF documents containing tables, equations, diagrams, etc. are rarely accessibile through screen readers. Since these documents are prevalent in STEM subjects, their recognition and analysis, in particular with a view to making them fully accessible to visually impaired readers has drawn some attention over the years.

Initial attempts included the application of powerful OCR techniques similar to those for scanned documents in order to perform formula recognition [4]. However, since PDF documents effectively contain most of the information on characters and fonts that are important for successful document analysis, a way to avoid OCR errors is to extract character information directly. The MaxTract [1] system exploited this information and analysed digitally born PDF files directly. One obstacle for this approach is the lack of exact bounding box information for characters in PDF documents. While this information is not particularly important for textual content that is linear, it is crucial for formula recognition due to the two-dimensional layout of mathematics. Maxtract overcame that obstacle by introducing an OCR step exclusively to match extracted character information against recognised bounding boxes of glyphs. While MaxTract had reliable formula recognition capabilities, it suffered from the use of a bespoke PDF parser making it unsuitable for PDF's from the wild, and was therefore not production ready. A similar technique as in MaxTract has recently been also implemented in the Infty system [6] for digitally born PDF files to complement their math recognition on scanned documents, thus reducing the need for full blown OCR on amenable PDFs.

In recent years it has become more and more common to open and read PDF documents directly in browsers rather than in third party plug-ins or downloading them first. This is generally achieved by employing transformation systems that translate PDF documents on the server-side into an HTML representation that can be browsed at client-side by a user. And while they mainly aim at preserving the visual layout, they often produce reasonably accessible results for pure text documents. That is, screen readers can voice the content of regular text passages but generally fail on more complex content like tables or formulas. Moreover, there are deficits in marking up content meaningfully to distinguish headings at different levels, paragraphs, headers and footers. In our work we build a primarily client side system that allows the transformation of digitally born PDF documents into HTML or ePub format that contains semantically important markup not only for textual elements but for complex structures like tables and formulas as well. We thereby combine a powerful open-source PDF parsing library — Pdf2htmlEX [5] — with higher level semantic analysis and table recognition as well as the useful parts of MaxTract for formula analysis. This alleviates the limitations of a bespoke parser by using a standards compliant PDF parser. Our system generates HTML output with its mathematical content in LaTeX and operates on the client side within the browser. Working within the web browser also allows us to leverage the capabilities of MathJax, a near standard library for accessible rendering of LaTeX math in the web browser. Figure 1 shows the architectural components of the system.

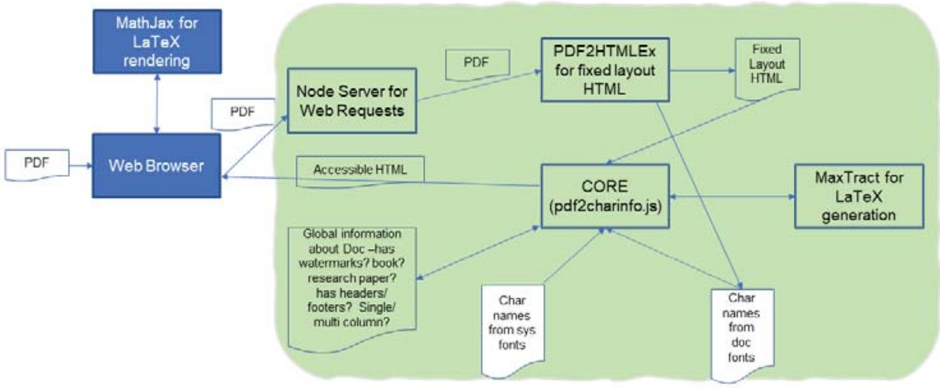


Figure 1: Components of the web-based system for conversion of digitally generated PDF's. Fixed layout HTML from pdf2htmlEX is passed to the core for conversion to accessible and reflowable HTML. Mathematical content is converted to LaTeX using MaxTract and the LaTeX is rendered inside HTML using the MathJax library.

In this paper we present our work which forms part of the research project RAVI (Reading Assistance for Visually Impaired) for making digitally born PDF's accessible.

2 Extracting and Understanding Content

The initial step in our workflow is the extraction and interpretation of the content provided in the PDF file. File data is extracted with the help of Pdf2htmlEX [5] library which renders the content visually similar to PDF to prevent any loss of information. Consequently the original layout-based format with font and position information of the characters is available in HTML format. However, since the rendering mimics the PDF rendering, graphical information is rendered in a single background bitmap image, as an underlay to the textual content which is in the foreground. Thereby graphical information does not only comprise diagrams and images, but also vertical or horizontal lines appearing in tables, formulas, headers or footers, as well as images of non-Unicode characters inserted by Publishers to make content visually comprehensive. Figure 2 represents the page of a mathematical article broken up into the textual elements in the foreground and the graphical elements in the background image.

Rendered textual content present in the HTML file is given in a series of `<div>` and `<span>` tags with positional and styling information in CSS. However, these have no semantic meaning, e.g., single spans can comprise multiple words or one word can be split into single spans, depending on font or layout considerations like spacing or kerning. Consequently there is even less markup indicating semantically meaningful grouping like paragraphs, headings or mathematics.

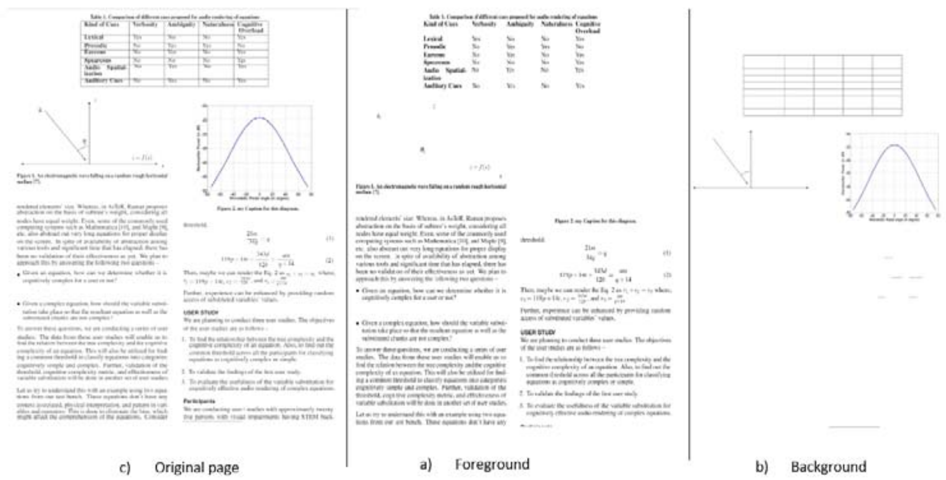


Figure 2: Example page and its rendering in separate foreground and background.

Nevertheless if that information is in a font where characters correspond to their ASCII positions, they can generally be voiced reasonably well by screen readers. However, common characters are transformed via CSS and will not resemble the displayed symbol. For example, in the computer modern extended (CMEX) font, which is commonly used for mathematical papers, symbols like integral or summation are associated with code points of characters 'Z' and 'X', respectively. Consequently the actual ASCII characters in the foreground page are 'Z' and 'X', which are only rendered correctly when the font information is available. Both font names and characters ASCII positions are stored as a binary sources in CSS, which we can decompress to obtain exact symbol and font information.

For the spatial representation of the mathematical content or table content the exact spatial information of the characters is important. Like in PDF, bounding box information is not available, as CSS character transformations are performed by the browser's layout engine and can not be measured in the client. To get the exact bounding box, each span from HTML is drawn separately on the canvas and the exact bounding boxes of its component glyphs are computed using a connected component analysis of the non-white pixels.

3 Component Analysis and Tagging

After the initial content extraction, the obtained information of position, bounding box, and font of characters is available in JSON format for further analysis. This information, together with the HTML content and the background image is now used for further analysis where we separate and tag content into different components, parts like (1) Text (2) Tables (3) Mathematics .

3.1 Text

To extract layout information of the textual components in the foreground, we use both font and bounding box data. We step-wise combine components into words, lines, paragraphs, etc. To combine spans via bounding boxes, a clustering approach based on [2] is used. This is fine tuned based on the characteristics of the document like single column or two column document, STEM related document etc. In future work, we plan to learn these characteristics from the document itself.

In the initial step, text lines are extracted from the page with a line algorithm which combines bounding boxes that are close to each other and overlap vertically. These text lines are next analysed using font size, font type, distance to identify headings. Remaining text lines that appear close to each other are grouped into paragraphs. Similarly, by analyzing the positions of the text lines across pages, header and footer components are identified and removed. List items are also tagged with appropriate HTML markup. Figure 3 shows the application flow.

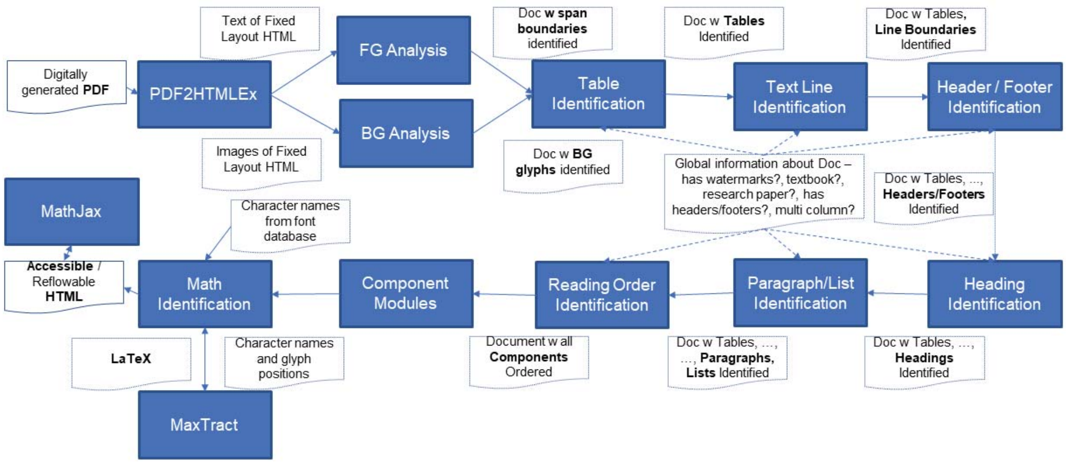


Figure 3: The Application Flow: output from pdf2htmlEX is analyzed by different specialized units for identifying Tables, Text Lines, Headers/Footers, Headings, Paragraphs and List Items. Once identified, these components are potentially reordered to match the reading order and then output with appropriate tags. Math if any is converted to LaTeX using MaxTract and rendered using MathJax

3.2 Tables

Our table recognition procedure works by analysing and cross referencing foreground with background information to gain insights about the table structure. In particular, we group tables into three categories (See Figure 4):

i) **Tables with a complete grid structure**, i.e., both horizontal and vertical lines are used to separate rows and columns. Our algorithm relies on the lines present in the table grid and identifies spans enclosed inside the cell boundaries to recreate the table.

ii) **Tables enclosed in a box that may or may not contain lines.** We rely on candidate outer boundaries from the background to find enclosed spans treating each bounding box as a potential table and identifying its structure. See Algorithm 1 for details.

iii) **Tables that are exclusively in the foreground**. For such tables we rely on text clusters to identify cells and then combine the cells into tables. Due to little other structure information this is not always successful.

Income level of individual or geography	% of the area median income
Low-income	Less than 50
Moderate-income	At least 50 and less than 80
Middle-income	At least 80 and less than 120
Upper-income	120 or more

Age groups	U.S. population		
	Proportion (total)	Proportion (20+ years)	Total
Under 1 year	0.0156		3,533,692
1 - 2 years	0.0287		6,493,373
3 - 5 years	0.0419		9,483,880
6 - 11 years	0.0920		20,834,439
12 - 19 years	0.1418		32,113,079
20 - 29 years	0.1803	0.2650	40,839,623
30 - 39 years	0.1392	0.2046	31,526,222
40 - 49 years	0.1005	0.1477	22,759,163
50 - 59 years	0.1030	0.1514	23,325,286
60 - 69 years	0.0833	0.1225	18,870,102
70 - 79 years	0.0512	0.0752	11,591,846
80 years plus	0.0228	0.0336	5,175,100
Total			226,545,805

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Figure 4: Tables with all boundaries, with only outer boundary and without any boundary

Algorithm 1 Analysis of Tables with only outer boundaries

```
for Span in spanList: do
  if New Row Condition then
    CurrRow.push(CurrCell) Table.push(CurrRow) CurrRow and CurrCell initialised with
    CurrSpan
  else
    if New Column Condition then
      if Span is inline with a previously made column then
        Add in that column
      else
        Row.push(CurrCell) Initialise CurrCell with CurrSpan
      end if
    else
      if Span is inline with a previously made column then
        Add in that column
      else
        CurrCell.push(CurrSpan)
      end if
    end if
  end if
end for
CurrRow.push(CurrCell)
Table.push(CurrRow)
Table.SubRowAnalysis()
Table.SubCellAnalysis()
```

3.3 Mathematics

Mathematical formulas can be composed of both symbols in the foreground and graphical elements in the background. For example, Figure 5 shows an equation split up into its foreground elements at the top and background elements at the bottom. Note that the horizontal lines for fractions and the root are in the background. Such mathematical regions are isolated so

$$\hat{n} = \frac{1}{\sqrt{1 + (\frac{df}{dx})^2}} \left(-\frac{df}{dx}\hat{x} + \hat{z} \right)$$

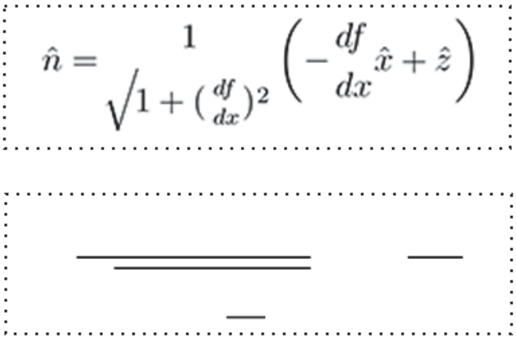


Figure 5: Mathematical equation split into foreground and background content.

that we can send them separately to MaxTract for conversion to LaTeX. While the original pattern matching engine of MaxTract [1] system was capable of converting full pages to LaTeX, its unique strengths lay in Math formula analysis. Sending only mathematical regions to Maxtract allows us to fine-tune the overall layout analysis outside of it and delegate to it the Math analysis. To isolate these math regions from the rest of the text for processing by MaxTract, we use heuristics based on the following

i) Use of Math fonts such as Symbol, CMR and use of Math symbols such as $+, =$ or functions such as log, exp, sin. The font and character information propagated from the pdf by pdf2htmlEx comes handy here especially for LaTeX generated pdf's.

ii) It is not uncommon to find cases where a non-mathematical or unknown font is used to display Mathematical content and where character information alone cannot identify a maths element. For such cases we make use of the presence of overlapping horizontal lines to identify mathematical content. Horizontal lines of thickness and aspect ratio within specific thresholds are considered to be belonging to Mathematical regions.

When converting different regions to HTML each mathematical region identified above is passed through MaxTract. The LaTeX output from MaxTract is embedded in its original position relative to its surrounding text with markup - \(and \) - suitable for rendering by MathJax [3]. As we include the MathJax library in the output HTML, the library takes care of suitably rendering the mathematical content.

While the collection of nearly 400 TeX fonts provides a sizeable collection of character to name mappings, characters from unknown fonts are propagated as is. This potentially means that some mathematical text will remain inaccessible in the output. This limitation can be mitigated to some extent by manually providing the character to name mappings for frequently occurring unknown fonts of the document collection. The system provides for integration of such handwritten font tables along with the existing system font tables.

4 Conclusion

In our project we are working towards the development of a web-based automatic conversion tool that can convert PDF to accessible file formats. Our main goal is to provide meaningful semantic markup for document elements like headings, paragraphs, tables, and formulas to make the content more easily accessible with screen readers for visually impaired readers. Textual layout analysis is based on pdf2htmlEX output combined with MaxTract's strengths on formula analysis to reconstruct files in an accessible HTML format. This ensures that there are no restrictions in PDF version that can be handled while allowing formulas to be translated into LaTeX to exploit the strength of online processing system such as MathJax. We have started integrating the different components of the system and are testing the results both in terms of their accessibility and by comparison with existing tools.

References

[1] Josef B. Baker, Alan P. Sexton, and Volker Sorge. Maxtract: Converting pdf to latex, mathml and text. In *Proceedings of the 11th International Conference on Intelligent Computer Mathematics*, CICM'12, pages 422–426, Berlin, Heidelberg, 2012. Springer-Verlag.

[2] Anil K. Jain and Bin Yu. Document representation and its application to page decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):294–308, March 1998.

[3] MathJax — beautiful math in all browsers. https://www.mathjax.org/.

[4] M.Suzuki, F.Tamari, R.Fukuda, S.Uchida, and T.Kanahori. Infty- an integrated ocr system for mathematical documents. In C.Vanoirbeek, C.Roisin, and E. Munson, editors, *Proceedings of ACM Symposium on Document Engineering*, pages 95–104, Grenoble, France, 2003.

[5] Lu Wang. pdf2htmlex: Publishing PDF online.

[6] Katsuhito Yamaguchi, Toshihiko Komada, Fukashi Kawane, and Masakazu Suzuki. New features in math accessibility with infty software. In Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, editors, *Computers Helping People with Special Needs*, pages 892–899, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

Locating Mathematical Definitions in a Document

Akashdeep Bansal*[1], Pawan Kumar†[1], Volker Sorge‡[2], and M Balakrishnan§[1]

[1]Indian Institute of Technology Delhi, India
[2]University of Birmingham, United Kingdom

Abstract

At present STEM education beyond middle school is inaccessible to visually impaired students in countries like India. A few exceptions can be traced to enormous efforts of the parents in working with these students to make STEM text material accessible. Access to equations, tables, charts and figures are the key bottleneck. With the increased penetration of screen reading software, effective audio rendering of equations can significantly help in making many of the e-texts accessible. Unfortunately, linear syntactic rendering of equations can create considerable cognitive load even for relatively simple equations. In this work we plan to address an important aspect for making mathematical subjects accessible. It is related to contextual rendering of equations based on the subject of the document as well as local definitions. We are working on a classifier-cum-locator which can identify the mathematical definitions in a document. This will help in adapting audio rendering of equations based on their contextual semantics.

1 Introduction and Motivation

Despite significant advances in assistive technologies, a large section of visually impaired students, especially in developing countries like India, are not able to pursue STEM subjects in senior school years and college. Inaccessible STEM content including textbooks contribute in a major way to this challenge. Clearly when significant employment opportunities depend upon STEM education, this has a huge impact on their employability and integration into society.

Access to equations is a critical requirement for STEM. Audio rendering and tactile Braille are the two main modalities used by persons with visual impairment for accessing equations. Audio is the preferred modality due to ease and cost of production, possibility of digital dissemination as well as having no need for specially trained instructors to teach Braille/Tactile Graphics. MathJax [5, 6], MathPlayer [12, 8], ChattyInfty [9, 18], etc. are some of the common tools used for accessing equations using audio. All these solutions provide syntactical audio rendering with optimized use of lexical and prosodic cues. None of these solutions have the capability of adapting the rendering on the basis of the contextual semantics. For example, A^T can be "the transpose of the matrix A" or "A to the power T" depending on the context whether A is a matrix or a variable, respectively. Current systems will always read it as "A superscript T", which is a syntactical rendering.

Lack of contextual rendering adds cognitive load and leads to the steepness in the learning curve for users using audio rendering. Another key contributor is linearization of the long and complex equations to render it through a linear interface such as audio [3]. We are already working on a solution for effective delivery of long and complex equations [4]. To achieve this we first convert eBorn PDFs into ePub [14, 13] and based on a complexity metric decompose equations dynamically when rendering.

In this work, we are focusing on contextually improved voicing of equations by locating and interpreting mathematical definitions in documents. Mathematical definitions are those phrases that assign mathematical

*akashdeep@cse.iitd.ac.in
†pawan.kumar@cse.iitd.ac.in
‡v.sorge@cs.bham.ac.uk
§mbala@cse.iitd.ac.in

properties to symbols used in subsequent text. For example, "Let $F : R^n \to R^n$ be a C^1-vector field" defines F to be a C^1-vector field. Unfortunately, definitions are only rarely given in such obvious manner and meaning is often assigned to symbols in a much more subtle way. Consequently, previous research [19, 15, 2] has demonstrated that machine learning techniques are more helpful than simple pattern matching techniques. In our work we aim to train a classifier for definitions based on a large ground truth set. In order to construct the latter we have assembled an annotation tool that presents the concordance analysis of a corpus of pre-processed documents which can be manually annotated to extract valid mathematical definitions. In this abstract we present the basic principles of the analysis and the functioning of the annotation tool.

2 Semantic Context Analysis

The true meaning of mathematical symbols can only rarely be deduced by their occurrence in a single formula alone. And although many commonly used mathematical symbols often have widely understood semantics, their meaning in context of a particular mathematical text might differ considerably. For example, while in non-associative algebra structures like rings use addition and multiplication symbols, their meaning must not be confused with their counter parts in traditional arithmetic.

Lack of contextual semantic leads to two types of ambiguities —

1. **Lexical ambiguity** — A symbol can have different meaning on the basis of the context. e.g., δ can be mapped to KroneckerDelta, DiracDelta, DiscreteDelta, or δ.

2. **Characteristical ambiguity** -– An expression can have different interpretation based on the characteristics of a symbol, even after having the same lexical meaning. e.g., w^{-1} can be an inverse function or $1/w$ depending on whether the symbol w is a function or a variable, respectively. Here, even though the lexical meaning of w and -1 is same in both the scenarios. Still, the expression w^{-1} has a completely different interpretation based on whether w is a function or a variable. In the former case,w^{-1} means the inverse function, in the latter w^{-1} means "1 divided by w

We can roughly divide the sources from which a given symbol or expression can get its definition in a document into three scenarios:

- **Scenario 1**: The symbol is never defined within the document, but is well understood in general or in the subject domain the document belongs (e.g., the equality sign or the normal subgroup sign in Group theory). Note here there is some implicit assumption about the audio rendering system already having default meaning of symbols for every domain.

- **Scenario 2**: The symbol is defined once within the document and is expected to carry that meaning throughout the document.

- **Scenario 3**: The same symbol is defined multiple times within the document and is expected to carry that meaning only within the scope of those definitions. In this case, it is also necessary to determine the scope of each definition and thus tailor the rendering accordingly.

Scenario (1) requires the identification of subjects domains having different semantics and creation of default symbol mappings for each domain. To some extent, this exists for certain topics, e.g., Wikipage [1] provides a detailed list of definitions of mathematical symbols in a small number of different subjects. While reading a document dealing with probabilities, reading $P(A)$ as "Probability of event A" is critical for understanding the equation instead of "P OpenParen A CloseParen". One could naively enforce an interpretation of P as probability or allow for explicit selection of the domain. But even if the subject domain is explicitly known to be probability theory, a default interpretation for P as probability might not necessarily be correct, as P could for example also denote a polynomial or a partition. Therefore, a dynamic approach should be more robust.

In our work, we are therefore focusing on the semantic information, which can be learned from the document itself; no prior domain knowledge is required. Hence, this work covers the scenarios (2) and (3) in the above list.

In the past, there have been few attempts in this area: it has been shown previously [17, 19, 10, 15, 2] that the consideration of surrounding text can relatively improve the performance of semantic disambiguation in comparison with a mere expression analysis [16, 7, 11]. But it has also been demonstrated [19] that machine learning based approaches can provide better results in comparison to simple pattern matching based methods. We therefore follow a machine learning approach for inferring mathematical definitions automatically from the document using a classifier-cum-locator. In particular we aim to extend work from [2] to train a classifier on the basis of a large ground truth set of definitions obtained from a concordance analysis, following a methodology that is outlined by these steps:

- **Identifying mathematical symbols definitions**: This requires the identification of the concordances which define the definitions of the mathematical symbols in the document. We achieved it by creating an efficient annotation tool (described in section 4). This is helping us in effectively annotating a large corpus to create the ground truth.
- **Classifier**: Developing a machine learning based classification algorithm. This will be trained on the ground truth created using the above annotation tool. This will help us in identifying whether the given concordance contains the mathematical definition or not. This is described in section 6.
- **Identifying the scope of these definitions**: To distinguish between the above mentioned scenarios (2) and (3), it needs to be inferred whether the given definition's scope is local or global. We plan to achieve it by checking whether a particular expression/symbol has multiple different definitions in the given document and associating the occurrence of that particular expression/symbol to an immediately preceding definition. By default, the expression/symbol would be mapped to the lexical meaning. After working on the scenario (1), it can be mapped to the default meaning associated with the subject.

3 Concordance Analysis

Our machine learning approach is based on a concordance analysis, that is on a enumeration of all expressions in question together with the context in which they occur. In our case, we are working with mathematical expressions and as context we take up to five words or expressions before and after the principal expression.

Table 1 contains a number of examples of concordances, where the principal mathematical expression is marked in red. Note, that single mathematical expressions are counted as a single element of the concordance, regardless of the number of symbols they contain. Moreover, while the principal is usually in the center, this can be broken up by paragraph making concordance pre- or postfixes shorter than five expressions/words.

Obviously not every concordance constitutes a definition of a mathematical expression. Table 1 contains examples of both together with an explanation why or why not a concordance can be seen as a definition. As a consequence it is necessary to build a ground truth by manually tagging of concordances as to whether they contain a definition or not.

4 Annotation Tool and Ground Truth

To generate the ground truth we need to annotate the concordances manually by several independent annotators. Although there are a number of tools for concordance analysis and corpus annotations available, none can adequately deal with documents containing considerable amount of mathematics. We have consequently developed an annotation tool with the objective to provide an efficient way for annotating the concordances having and not having a valid mathematical definition.

For this, we took around 100 XML documents from various domains of STEM. In these documents, the tool first identifies the mathematical entities (maybe a symbol or expression). As in general on the web, we found documents which are not appropriately tagged. Hence, the tool searches for various tags such as MathML, bold, and italic, and also look for the LaTeX expressions. The tool searches for bold and italic because many of the times authors tagged the mathematical entities by bold/italic to give the same visual appearance. The average number of entities identified in the documents are 1052.644, having max of 3046 and min of 252 and standard deviation of 586.2051.

Concordance	**Definition (Yes/No)**	**Explanation**
We present, in dimension $n \geq 2$, a survey of samples to:	Yes	n is a dimension.
ensuring that an equilibrium point x^* is a local attractor is	Yes	x^* is a point.
$F : R^n \to R^n$ satisfying that for any $x \in R^n$, all the eigenvalues of $JF(x)$	Yes	x represents eigenvalues.
$n < 2$ and non-injective polynomial maps $f : R^n \to R^n$ with $[0, \infty) \cap Spec(Jf(x)) = 0$, for all $x \in R^n$".	Yes	f is a non-injective polynomial map.
that the C^1-vector fields $F : R^n \to R^n$ satisfying that for any $x \in R^n$	Yes	F is a C^1-vector field.
Let us recall that the C^1-vector fields $F : R^n \to R^n$ satisfying that for any $x \in R^n$	No	This is not a definition of C^1.
$JF(x)$, the Jacobian matrix of F at x, has negative real	No	The definition of F is not given; even though you can infer that F is a function, this is not explicit in this concordance.
restricted to the invariant plane $z = 0$ is a center. Moreover, perturbing	No	Even though $z = 0$ is a plane is defined in this concordance. Still, to infer that z is a dimension, you requires the prior domain knowledge. That's why it is not a valid mathematical definition. Here, we are only looking for the concordances which has the proper definition of the symbol within the concordance.
construct polynomial maps $F = \lambda I + H : R^3 \to R^3$ with JH nilpotent, such that the WMYC	No	This is not a definition of JH.
the MYC is true when $n \leq 2$ and false when $n \geq 3$ (see	No	This is not a definition of n.

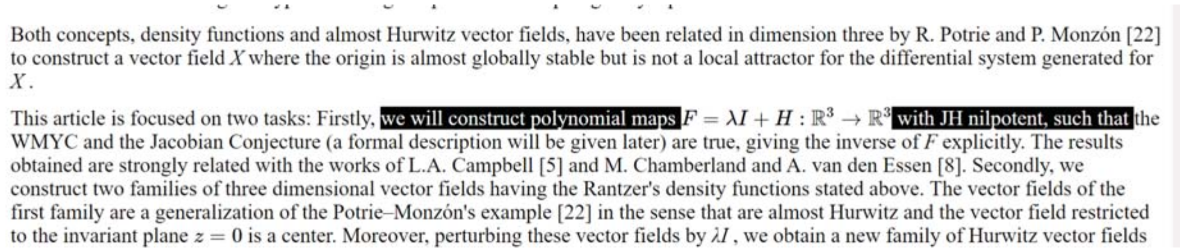
Table 1: Some sample concordances containing and not containing valid mathematical definitions. The principal mathematical expression is marked with red color in above concordances.

Corresponding to each identified entity, five previous and five next words are highlighted (as shown in Figure 1). This whole highlighted part, including the central mathematical entity is referred to as concordance in this paper. The person who is annotating the concordances needs to press $\rightarrow$ (right arrow) or $\leftarrow$ (left arrow) to mark the highlighted concordance as a valid or not a valid mathematical definition, respectively. To provide the assurance, on pressing the right-arrow/left-arrow the background colour gets changed to blue or red, respectively (as shown in Figure 2 and 3, respectively.). The tool also provides an option to mark error in concordance identification/highlighting by pressing the key "e" (Figure 4). The navigation functionality was provided through up/down arrow keys to go to the previous/next concordance, respectively. The tool also automatically goes to the next concordance after 1 second of marking the currently highlighted concordance.

This tool allows us to create a large amount of ground truth efficiently. The average annotation speed observed is that a person can annotate 400 concordances within 30 minutes. Also, the same file is annotated by multiple people having a strong background in STEM, to remove any personal bias.

5 Data Collection

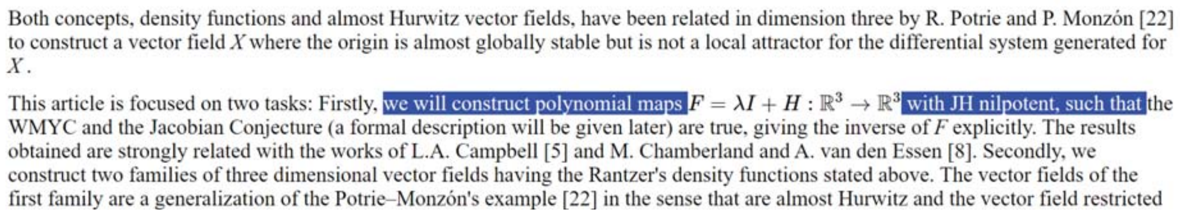
The present analysis is based on nine files, all of them have been annotated by three different annotators. The same file was annotated by multiple annotators to avoid the individual biases. Final label is decided based on the max approach, detailed rules are mentioned in table 2. e.g., If two annotators have given



Both concepts, density functions and almost Hurwitz vector fields, have been related in dimension three by R. Potrie and P. Monzón [22] to construct a vector field X where the origin is almost globally stable but is not a local attractor for the differential system generated for X.

This article is focused on two tasks: Firstly, we will construct polynomial maps $F = \lambda I + H : \mathbb{R}^3 \to \mathbb{R}^3$ with JH nilpotent, such that the WMYC and the Jacobian Conjecture (a formal description will be given later) are true, giving the inverse of F explicitly. The results obtained are strongly related with the works of L.A. Campbell [5] and M. Chamberland and A. van den Essen [8]. Secondly, we construct two families of three dimensional vector fields having the Rantzer's density functions stated above. The vector fields of the first family are a generalization of the Potrie–Monzón's example [22] in the sense that are almost Hurwitz and the vector field restricted to the invariant plane $z = 0$ is a center. Moreover, perturbing these vector fields by λI, we obtain a new family of Hurwitz vector fields

Figure 1: A Screenshot showing a sample concordance highlighted by the annotator tool



Both concepts, density functions and almost Hurwitz vector fields, have been related in dimension three by R. Potrie and P. Monzón [22] to construct a vector field X where the origin is almost globally stable but is not a local attractor for the differential system generated for X.

This article is focused on two tasks: Firstly, we will construct polynomial maps $F = \lambda I + H : \mathbb{R}^3 \to \mathbb{R}^3$ with JH nilpotent, such that the WMYC and the Jacobian Conjecture (a formal description will be given later) are true, giving the inverse of F explicitly. The results obtained are strongly related with the works of L.A. Campbell [5] and M. Chamberland and A. van den Essen [8]. Secondly, we construct two families of three dimensional vector fields having the Rantzer's density functions stated above. The vector fields of the first family are a generalization of the Potrie–Monzón's example [22] in the sense that are almost Hurwitz and the vector field restricted

Figure 2: A Screenshot showing a sample concordance highlighted by the annotator tool after pressing right arrow (key for marking as having valid mathematical definition.)

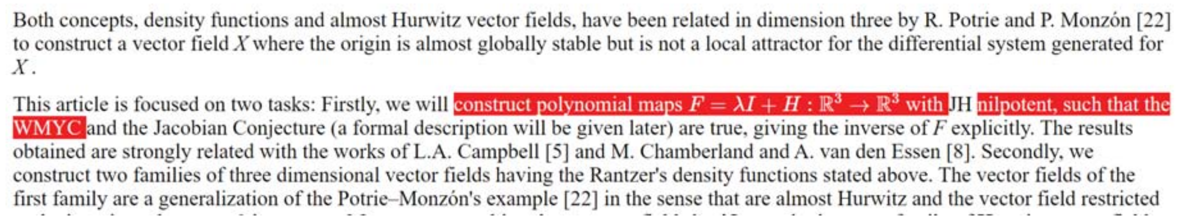
label 1, whereas, one annotator has given the label 0, then the final label will be 1. Here, 0 represents ”doesn’t contain a valid mathematical definition”, 1 represents ”contains a valid mathematical definition”, and −1 represents ”an error”. The average number of concordances in these files are 619. The total number of concordances having final label as 0 and 1 are mentioned in table 3. We have split this collection into training and test set in 80:20 ratio.

Label 1	Label 2	Label 3	Final Label
0	0	0	0
0	0	1	0
0	1	1	1
1	1	1	1
-1	0	0	0
-1	0	1	-1
-1	1	1	1
-1	-1	0	-1
-1	-1	1	-1

Table 2: Rules for final label based on the labels from different annotators. Here, 0 represents ”doesn’t contain a valid mathematical definition”, 1 represents ”contains a valid mathematical definition”, and −1 represents ”an error”.

Label	Total Number
1	892
0	4199

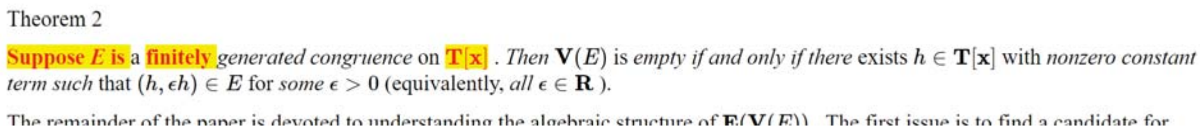
Table 3: The total number of concordances having final label as 0 and 1.



Both concepts, density functions and almost Hurwitz vector fields, have been related in dimension three by R. Potrie and P. Monzón [22] to construct a vector field X where the origin is almost globally stable but is not a local attractor for the differential system generated for X.

This article is focused on two tasks: Firstly, we will construct polynomial maps $F = \lambda I + H : \mathbb{R}^3 \to \mathbb{R}^3$ with JH nilpotent, such that the WMYC and the Jacobian Conjecture (a formal description will be given later) are true, giving the inverse of F explicitly. The results obtained are strongly related with the works of L.A. Campbell [5] and M. Chamberland and A. van den Essen [8]. Secondly, we construct two families of three dimensional vector fields having the Rantzer's density functions stated above. The vector fields of the first family are a generalization of the Potrie–Monzón's example [22] in the sense that are almost Hurwitz and the vector field restricted

Figure 3: A Screenshot showing a sample concordance highlighted by the annotator tool after pressing left arrow (key for marking as not having a valid mathematical definition.)



Theorem 2

Suppose E is a finitely *generated congruence* on $\mathbf{T}[\mathbf{x}]$. *Then* $\mathbf{V}(E)$ is *empty if and only if there* exists $h \in \mathbf{T}[\mathbf{x}]$ with *nonzero constant term such* that $(h, \epsilon h) \in E$ for *some* $\epsilon > 0$ (equivalently, *all* $\epsilon \in \mathbf{R}$).

;

Figure 4: A Screenshot showing a concordance for an erroneously identified element.

6 Classifier

As a baseline classifier, we have considered Naive Bayes and SVM. The concordance pre-processing requires removing punctuations and stopwords, using NLTK library, and replacing mathematical symbols/expressions with keyword "MATH". Table 4 shows some sample concordances after replacing mathematical symbols/expressions by the keyword "MATH". Each concordance is vectorized using term frequency and inverse document frequency, abbreviated as tf-dif. We have used uni-grams as well as bi-grams and taken logarithm of their term-frequency. The SVM, from library sickit-learn, is trained with early-stopping and uses 20% of the training data as validation set. We have used the same tf-idf based feature vector for SVM as well. As evident from the table 5 and 6, SVM provides better f1-score and makes for a better choice for a baseline classifier. In Future experiment on classifier, we will try to bring in semantics of the context window using deep-learning based models to learn embeddings for the concordances that might help improve the accuracy and f1-score.

7 Conclusion and Future Work

The developed annotation tool is found to be very useful in quickly annotating the files for generating ground-truth. Even with limited annotations, using a classifier we are able to achieve an accuracy of 85% and f1-score of 0.70. SVM with early-stopping is found to be a better choice for baseline classifier.

Further, We are exploring the possibility of more meaningful replacement of mathematical symbols/expressions for tokenization. We also want to explore how we can use these definitions for semantic disambiguation of the mathematical symbols. We are also analyzing whether the current choice of five previous and next words is "appropriate and effective" or it needs to be changed. Finally, we want to identify each mathematical definition's scope to determine whether this is a local or global definition.

Acknowledgement

We would like to thank Mark Lee for his valuable inputs. This project is funded under the SPARC (Scheme for Promotion of Academic Research Collaboration), Ministry of Education, Govt. of India. MathJax work was supported in part by Simons Foundation Grant, No.514521.

Concordance	Tokens
We present, in dimension $n \geq 2$, a survey of samples to:	We present, in dimension MATH , a survey of samples to:
ensuring that an equilibrium point x^* is a local attractor is	ensuring that an equilibrium point MATH is a local attractor is
$F : R^n \to R^n$ satisfying that for any $x \in R^n$, all the eigenvalues of $JF(x)$	MATH satisfying that for any MATH , all the eigenvalues of MATH
$n < 2$ and non-injective polynomial maps $f : R^n \to R^n$ with $[0, \infty) \cap Spec(Jf(x)) = 0$, for all $x \in R^n$".	MATH and non-injective polynomial maps MATH with MATH, for all MATH".
that the C^1-vector fields $F : R^n \to R^n$ satisfying that for any $x \in R^n$	that the MATH -vector fields MATH satisfying that for any MATH
Let us recall that the C^1-vector fields $F : R^n \to R^n$ satisfying that for any $x \in R^n$	Let us recall that the MATH -vector fields MATH satisfying that for any MATH
$JF(x)$, the Jacobian matrix of F at x, has negative real	MATH, the Jacobian matrix of MATH at MATH, has negative real
restricted to the invariant plane $z = 0$ is a center. Moreover, perturbing	restricted to the invariant plane MATH is a center. Moreover, perturbing
construct polynomial maps $F = \lambda I + H : R^3 \to R^3$ with JH nilpotent, such that the WMYC	construct polynomial maps MATH with MATH nilpotent, such that the WMYC
the MYC is true when $n \leq 2$ and false when $n \geq 3$ (see	the MYC is true when MATH and false when MATH (see

Table 4: Tokenized form of sample concordances after replacing mathematical symbols/expressions with the keyword "MATH".

	Naive Bayes	SVM
Accuracy	0.841	0.852
f1-score	0.652	**0.702**

Table 5: Accuracy and f1-score of base classifier based on the Naive Bayes and SVM.

	Naive Bayes		SVM	
	1	**0**	**1**	**0**
1	53	126	72	107
0	36	804	43	797

Table 6: Confusion Matrices

References

[1] List of mathematical symbols by subject. `https://en.wikipedia.org/wiki/List_of_mathematical_symbols_by_subject`, 2020.

[2] R. Almomen. *Context classification for improved semantic understanding of mathematical formulae*. PhD thesis, University of Birmingham, 2018.

[3] A. Bansal, M. Balakrishnan, and V. Sorge. Comprehensive accessibility of equations by visually impaired. *ACM SIGACCESS Accessibility and Computing*, (126):1–1, 2020.

[4] A. Bansal, M. Balakrishnan, and V. Sorge. Evaluating cognitive complexity of algebraic equations. In *Journal on Technology & Persons with Disabilities*, page (to appear). CSUN, 2021.

[5] D. Cervone, P. Krautzberger, and V. Sorge. New accessibility features in mathjax. In *31th Annual International Technology and Persons with Disabilities Conference Scientific/Research Proceedings*. California State University, Northridge., 2016.

[6] D. Cervone, P. Krautzberger, and V. Sorge. Towards universal rendering in mathjax. In *Proceedings of the 13th Web for All Conference*, page 4. ACM, 2016.

[7] I. A. Doush, F. Alkhateeb, and E. Al Maghayreh. Towards meaningful mathematical expressions in e-learning. In *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, pages 1–5, 2010.

[8] L. Frankel, B. Brownstein, and N. Soiffer. Navigable, customizable tts for algebra. In *28th Annual International Technology and Persons with Disabilities Conference Scientific/Research Proceedings*. California State University, Northridge, 2014.

[9] T. Kanahori and M. Suzuki. Scientific pdf document reader with simple interface for visually impaired people. In *International Conference on Computers for Handicapped Persons*, pages 48–52. Springer, 2006.

[10] M.-Q. Nghiem, G. Y. Kristianto, G. Topić, and A. Aizawa. A hybrid approach for semantic enrichment of mathml mathematical expressions. In *International Conference on Intelligent Computer Mathematics*, pages 278–287. Springer, 2013.

[11] M.-Q. Nghiem, G. Yoko, Y. Matsubayashi, and A. Aizawa. Towards mathematical expression understanding.

[12] N. Soiffer. Mathplayer v2.1: web-based math accessibility. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 257–258. ACM, 2007.

[13] V. Sorge, M. Balakrishnan, N. Jadhav, A. Bansal, and H. Garg. Rule based mathematics extraction and layout analysis of pdf documents. In *Proceedings of the 4th International Workshop on Digitization and E-Inclusion in Mathematics and Science (DEIMS)*, 2021.

[14] V. Sorge, A. Bansal, N. M. Jadhav, H. Garg, A. Verma, and M. Balakrishnan. Towards generating web-accessible stem documents from pdf. In *Proceedings of the 17th International Web for All Conference*, pages 1–5, 2020.

[15] Y. Stathopoulos and S. Teufel. Mathematical information retrieval based on type embeddings and query expansion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2344–2355, 2016.

[16] J. Stuber and M. Van den Brand. Extracting mathematical semantics from LaTeX documents. In *International Workshop on Principles and Practice of Semantic Web Reasoning*, pages 160–173. Springer, 2003.

[17] M. Wolska and M. Grigore. Symbol declarations in mathematical writing. 2010.

[18] K. Yamaguchi, T. Komada, F. Kawane, and M. Suzuki. New features in math accessibility with infty software. In *International Conference on Computers for Handicapped Persons*, pages 892–899. Springer, 2008.

[19] K. Yokoi, M.-Q. Nghiem, Y. Matsubayashi, and A. Aizawa. Contextual analysis of mathematical expressions for advanced mathematical search. *Polibits*, (43):81–86, 2011.

The BrailleMathCodes Repository

Paraskevi Riga[1], Theodora Antonakopoulou[1], David Kouvaras[1], Serafim Lentas[1], and Georgios Kouroupetroglou[1]

[1]National and Kapodistrian University of Athens, Greece
Speech and Accessibility Laboratory,
Department of Informatics and Telecommunications
p.riga@di.uoa.gr, ic1190004@di.uoa.gr, sdi1000196@di.uoa.gr, sdi170007@di.uoa.gr, koupe@di.uoa.gr

Abstract

Math notation for the sighted is a global language, but this is not the case with braille math, as different codes are in use worldwide. In this work, we present the design and development of a math braille-codes' repository named BrailleMathCodes. It aims to constitute a knowledge base as well as a search engine for both students who need to find a specific symbol code and the editors who produce accessible STEM educational content or, in general, the learner of math braille notation. After compiling a set of mathematical braille codes used worldwide in a database, we assigned the corresponding Unicode representation, when applicable, matched each math braille code with its LaTeX equivalent, and forwarded with Presentation MathML. Every math symbol is accompanied with a characteristic example in MathML and Nemeth. The BrailleMathCodes repository was designed following the Web Content Accessibility Guidelines. Users or learners of any code, both sighted and blind, can search for a term and read how it is rendered in various codes. The repository was implemented as a dynamic e-commerce website using Joomla! and VirtueMart.

1 Introduction

Braille constitutes a tactile writing system used by people who are visually impaired. It employs embossed dots evenly arranged in quadrangular cells, with each cell being three dots high and two dots wide. Braille codes are systems for transcribing printed material using a braille alphabet. To increase the number of characters that can be represented with the six dots, a code either implements a special braille character called prefix or a composition sign or makes the meaning of a braille character dependent on the context. Braille can be extended to 8-dot cells to support STEM notification in assistive technologies, such as braille displays (Kacorri & Kouroupetroglou, 2013).

Early braille education of the blind is crucial to literacy, education, and employment. While in the 60s, braille literacy of legally blind school-age children in the USA was 50 percent in the United States, only 10 percent use braille as their primary reading medium (American Printing House for the Blind, 1996). Braille education provides visually impaired students with the opportunity to excel in several school areas such as vocabulary, comprehension, and even STEM, but also enhances their employment rates drastically (Riles, 2004).

One of the causes for the decline in braille usage was the move of children with blindenss from specialized schools into mainstream public schools, where only a small percentage could afford to train and hire Braille-qualified teachers. To surpass this problem, all involved in the learning process, teaching-staff, visually impaired students, and possibly parents/caretakers, should be trained in the braille codes according to the national standards.

Braille readers use different codes to read and write different types of documents. Visually impaired students are generally taught three main codes: a) literary, b) math, and c) music braille, with math being a lot more difficult as it is not linear, it uses a lot of special characters and must be correctly interpreted at all times (unambiguous).

Some codes integrate literary and math support, while others are math specific. Each country, and even each educational institution within the country, follows its teaching system both for the sighted people seeking qualification and the visually impaired students. The educational material used (user guides with examples and exercises) is divided into courses related to different aspects of math. For blind students, code symbols are introduced as they occur in the print versions of the material being used in the curriculum. For sighed people, on the other hand, code symbols are thematically organized. For example, in Greece, although the Nemeth braille code was officially adopted in 2004 (Kouroupetroglou & Florias, 2003), there is still no preparation course to offer instructions for the Nemeth code to the sighted, and the braille competency certification only refers to literary braille. In developing the educational material to teach such a course, we found a gap in web content. There was no index or repository to search for braille math symbols, Unicode, and LaTeX encodings.

2 Printed math for the blind

2.1 Braille Math Codes

Math notation for the sighted is a global language, but this is not the case with braille math, as different codes are in use worldwide (Library of Congress, 1990). Since rules for the encoding of mathematics are not compatible from one system to the next, braille users who wish to read or write a range of material coming from different countries need to learn different sets of rules. The rules of each code include: a) the representation of math symbols, b) the use of "formatting" information, and c) other indicators.

The notations that are currently in use include the Antoine Notation (French Braille Code), Nemeth Code, Unified English Braille Code (UEB), British Mathematics Notation (BAUK), Spanish Unified Mathematics Code, Marburg Mathematics (German Code), Woluwe Code (Notaert Code), Italian Braille Code, Swedish Braille Code, Finnish Braille Code, Russian Code, and Arabic Code. Some of them are solely and others partially mathematics-related.

French Braille, created by Louis Braille in 1837, is the original braille alphabet and the basis of all others. It was meant for literary braille. In 1922, the braille code was first adapted to Mathematics by Louis-Auguste Antoine, a mathematician who became blind at war. The last revision of Antoine Notation was in 2007 (Croisetière, D’Amour, Ferland, & Rainville, 2008).

Abraham Nemeth first wrote The Nemeth Braille Code for Mathematics (Nemeth, 1972) in 1952. “The code is intended to convey as accurate an impression as is possible to the braille reader of the corresponding printed text" and the result is a compact human-readable markup language where an ink-print math symbol with two different meanings is brailled the same in both instances. Nemeth Code is the standard for teaching and doing mathematics in braille in the US. 8-dot braille code has been introduced for complex Nementh symbols (Martos, Kouroupetroglou, & Argyropoulos, 2015)

UEB was initially specified in 1991 (Cranmer & Nemeth, 1991). It is intended to form one set of rules, the same worldwide, which could be applied across various types of English-language material, e.g., for literary and technical material. Even though at the beginning of 1992, UEB integrated the Nemeth Code, it fails to handle mathematics as compactly. Besides requiring more space to represent and more time to read and write, the verbosity of UEB can make learning mathematics more difficult.

British Mathematics Notation was first designed in 1970, and its last revision was in 2005 (Kingdom, 2005). It was used in the UK until UEB replaced it, but it is still used in many places in Africa.

Spanish Unified Mathematics Code, known in Spanish as "Codigo Matematico Unificado" (CMU), is widely used in Spanish and Portuguese speaking countries. The Spanish National Organization of the Blind (ONCE) proposed CMU in 1970 to unify the math symbolography but, it was only accepted in 1987 by all parties.

In 1955, the "Internationale Mathematikschrift für Blinde" notation was designed at the Marburg school for the Blind in Germany. Marburg notation, as it is known, was last updated in 1992 (Britz, Epheser, & Pograniczna., 1992) and is used by German-speaking countries.

In 1975 Flanders agreed on a code to represent mathematical formulae in braille, the Woluwe Code (Notaert, Suij, & Vandekerkhove, 1975) (after the Woluwe School for the blind), also known as Notaert Code (after Gilbert Notaert). It was created based on the Marburg Code and is used in Dutch-speaking parts of Belgium.

The last specification of the Italian Braille code, which incorporates math, was published in 2003 (Biblioteca Italiana, 2003).

In 1992, a report was published in Sweden to present the Swedish braille system with a set of symbols and writing principles for modem mathematics (Becker, Stenberg, Lindqvist , & Trowald, 1992). It was adopted by the Swedish Braille Council in June 1972, was also used in Iceland till 1990, and was last revised in 1912.

The Finnish mathematical notation was published along with physics and chemistry notation in 1979 (Central Union of the Blind, 1979). In the latest edition of 2014 ASCIIMath is used instead of braille, as students have moved to electronic books.

The formal braille code for math, physics, chemistry, and astronomy was published in a second edition in 1982 in Russia (Bykov, Egorov, Morozova, & Proskurjakov, 1982). It is also used in Belarus.

Since 1951, with the introduction of braille machines, braille in Arabic is read from left to right to be in line with the braille reading format of all languages. The braille codes in the Arabic nations include Arabic pre-2002, Arabic, Dari, Dhivehi, Farsi, Somali, and Urdu Pakistani. The Unified Arabic Braille Project started in 2015, is an effort to reduce the limitations of the current systems (Mada, 2021). The math code of the Unified Arabic Braille is supported by the Liblouis translator.

Information about each code is found in the original publications, in dedicated teaching books, and online web pages made for learning. Until today, there was no available electronic repository to use as an electronic index or search tool for these codes.

2.2 Math pseudo-codes

Pseudo-codes are written using a qwerty-keyboard or WYIWYG applications and have a visual nature. They are not considered braille codes because they still require translation to braille (Whapples, 2007). Examples include LaTeX (Knuth, 1984) as well as variations of LaTeX such as Human Readable Tex (HrTex) (Suzuki, Kanahori, Ohtake, & Yamaguchi, 2004). Contrary to dedicated braille codes, they are not developed purely for the blind, but they are used when braille math notation is not an option. The LaTeX notation, which provides a more uniform format than braille, is commonly used to allow for easier communication with sighted users.

When it comes to higher education, LaTeX seems the most flexible notation; the user can expand it, and expansions are provided in packages by a supporting community. Although all braille math codes can express the most common mathematical structures in secondary education, most of them require enhancements to cover higher education mathematics.

2.3 Math XML

Presentation MathML (W3C Math Working Group, 1998), is an XML application for encoding mathematics on the Web based on their visual structure, an integral part of HTML5. MathML is used in braille translators to produce braille math codes and in AT applications to produce an acoustic rendering of math expressions, sometimes along with the haptic rendering.

LaTeX and MathML were found to be too complicated for use in primary and lower secondary education.

3 Assistive technologies and math

When math is in an electronic form, not graphically presented, then it can be rendered both haptically and acoustically. Screen readers, e.g., NVDA (Access, n.d.) and JAWS (Freedom Scientific, n.d.) are software applications that convert text displayed on screen into synthesized speech or braille. In the case of mathematics, the synthetic speech is valuable when math is in a format that can be rendered acoustically either straight from the screen reader or by another software application (e.g., MathPlayer). When math is expressed using a braille notation or some pseudo-code, then a refreshable braille display offers direct access to information, allows the user to check the format, spacing, and spelling in the text.

Students with visual impairments face challenges in learning from digital libraries and the web as the content is not expressed in an accessible way. The least the editors of such content can do is avoid graphic rendering of math expressions and use MathML or Unicode instead. This allows users to listen to the equations and even translate them to their braille code.

4 Math Braille Translators

At the moment, the best way for a blind person to read and write mathematical expressions is by using a mathematical braille notation. Communication of math between sighted people and braille users is done by math translation. Sighted people translate braille math to printed math, either by themselves or by using electronic braille translators. 100% correct braille translation can only be done by a human, as it usually depends on the context and requires text understanding.

Printed-to-braille math translators translate sighted users' code to braille code(s). UMCL provides a transcription of mathematical expressions from several mainstream formats to several mathematical braille codes (Archambault & Guyon, 2011), and math2braille translates MathML to the braille code used in the Netherlands (Crombie, Lenoir, McKenzie, & Barker, 2004). Duxbury DBT is a commercial program that handles LaTeX files and translates them to 9 math braille codes (Duxbury Systems, 2020).

Tools that backtranslate the braille math back to a format that can be read by a sighted individual. SBT is a tool that translates Spanish math braille to MathML (Alonso, Fuertes, González, & Martínez, 2006).

Translation and backtranslation programs translate some code used for the authoring of math by sighted users to some braille code(s) and vice versa. MAVIS developed the first Nemeth braille code to LaTeX backtranslator (Karshmer, Gupta, & Geiiger, 1998). The Lambda Mathematical Code (Schweikhardt, Bernareggi, Jessel, Encelle, & Gut, 2006) was directly derived from MathML. It was designed to be used with braille peripherals and speech synthesis. It is automatically (back)convertible, in real-time and error-free, into an equivalent MathML version and, through it, into the most popular mathematical formats. LaBraDoor started with translating LaTeX to Marburg notation and subsequently backtranslated Marburg to LaTeX (Murillo-Morales, Miesenberger, & Ruemer, 2016). Liblouis is an open-source literary and math braille translator and back-translator of MathML and supports Nemeth, Marburg, and Arabic Braille (Egli, 2009).

5 BrailleMathCodes Repository

5.1 Content

The first step to the creation of the repository was the gathering of the information. Materials and content were gathered from the web, and symbol tables were created for each mathematical braille code. For codes with an official book reference, the online book was used as a source. For codes with no book reference, we relied on the content provided by national blind associations and libraries for the blind. No changes were made to the written forms of the symbols other than their transcription from braille patterns or graphics to text. This was needed to make the content accessible to those using adaptive technologies (refreshable braille displays, speech synthesizers, etc.) and allow error fixes.

Each symbol or indicator was matched with an image, a Unicode, and a name. Unicode is the most standard multilingual character encoding and is descriptive of the symbol form (The Unicode Consortium, 2000). To match Unicode with a braille formed symbol, we did both textual and visual search of Unicode tables when there was no reference of it in the original source. The name was descriptive, as it is in the Nemeth book, and it was also accompanied by its mathematical use(s) in parentheses, when different, as the verbal rendering in math changes based on the context. For example, the Unicode character "circled plus", U+2295 is referred to as: a) direct sum (an operation from abstract algebra), b) dilation (mathematical morphology), and c) exclusive or (a logical operation). The name of the symbol changes with the spoken language of each country; currently we provide the verbal rendering of symbols in English and Greek. However, we can extend the verbal rendering translations to cover more languages

The code tables were then joined to one table, where we have all the existing forms for each symbol. The join process was semi-automatic. As symbol names were not expressed in the same way for each code, unicodes were not always present, and some notations presented their own braille indicators with no equivalency.

To organize the whole table in sections, we followed the classification of symbols and indicators made by Nemeth, where symbols are grouped based on their use or the rule that governs them. However, we changed the order and made a distinction between symbols and indicators (Table 1). We did not classify the symbols as they occur in the print versions of the mathematic texts, as the repository's scope is not to teach mathematical concepts.

Then, we matched each math braille code with its LaTeX analogous and forwarded it with Presentation MathML. For each math symbol, we selected a characteristic example in Nemeth code, translated in MathML, when applicable (Table 2).

Most math braille codes cover the symbols used in pre-university curricula (secondary education). To study maths in high-level education, a braille user has to extend the given notations. There are works, some continuing, reviewed and published in conferences and journals of special education, that extend some math braille codes, e.g., Nemeth (Martos, Kouroupetroglou, Argyropoulos, & Deligiorgi, Towards the 8-dot Nemeth braille code, 2014). When these extensions are adopted by official codes, we will include them in our repository, so that braille learners will not have to reinvent them.

Math Symbols	**Math Braille Indicators** 7
. |_ Signs and symbols . . |_ Punctuation signs and symbols 17 . . |_ Numeric signs and symbols 13 . . |_ Signs and symbols of operation 25 . . |_ Radicals 1 . . |_ Modifiers . . . |_ Arc 2 . . . |_ Bar 2 . . . |_ Caret 4 . . . |_ Arrow 48 . . . |_ Other modifiers 6 . . |_ Signs and symbols of comparison 116 . . |_ Signs and symbols of grouping 42 . . |_ Reference signs and symbols 7 . . |_ Miscellaneous signs and symbols 37 . |_ Shapes . . |_ Basic shapes 25 . . |_ Shapes with interior modification . . . |_ Angle 4 . . . |_ Circle 12 . . . |_ Square 6 . . |_ Shapes with structural modification . . . |_ Angle 11 . . . |_ Triangle 5 . |_ Function names and their abbreviations 38	. |_ Alphabetic indicators 39 . |_ Capitalization indicators 2 . |_ Fraction indicators 14 . |_ Level indicators (superscripts and subscripts) 15 . |_ Radical indicators 2 . . |_ Order of radical 3 . |_ Shape indicators 6 . |_ Arrow direction indicators 4 . |_ Cancellation indicators 2 . |_ Type-form indicators for words, phrases, and mathematical statements 4 . |_ Type-form indicators for letters, numerals, and compound expressions 4 . |_ Spatial arrangements 16 . |_ Modification indicators 7

Table 1: Classification of math symbols and braille indicators. Each category is followed by the number of its entries.

5.2 Design and implementation

The user interface of (BrailleMathCodes repository, 2021) was designed following the Web Content Accessibility Guidelines (WCAG) (World Wide Web Consortium), so the content is perceivable (all three modalities are present in the content) and understandable. The repository was checked to be both operable and robust. The user can either A) search for a symbol by a) name (text input), b) Unicode (dropdown), or c) LaTeX (dropdown) or B) browse through the website's math symbol/braille indicator categories to find what he is looking for (Figure 1).

The repository was implemented as a dynamic e-commerce website using Joomla! (Cao & Yu, 2010), an open-source content management system (CMS), and VirtueMart (VirtueMart, n.d.), one of the oldest and more established eCommerce extensions for Joomla, where each symbol is “a product” and the corresponding codes are its fields. The source code of the CMS and the extension had to be altered to support braille codes and make the website fully accessible. The free template Horme 3, which does mobile detection, was used in full-width layout, with some customizations in the CSS and PHP codes.

Symbol field title	Value
Image	∃
Verbal Wording in Greek	υπαρξιακός ποσοδείκτης (υπάρχει τουλάχιστον ένα)
Verbal Wording in English	existential quantifier (there exists, for some)
Category	Miscellaneous signs and symbols
Unicode	u+2203
LaTeX	\exists
MathML Code	<math xmlns="http://www.w3.org/1998/Math/MathML"> <mo mathvariant="normal">∃</mo> </math>
MathML Example	<math xmlns="http://www.w3.org/1998/Math/MathML"> <mo mathvariant="normal">∃</mo> <mi mathvariant="italic">x</mi> <mo mathvariant="normal">,</mo> <mi mathvariant="normal"> </mi> <mi mathvariant="italic">x</mi> <mo mathvariant="normal"><</mo> <mfrac> <mrow><mn mathvariant="normal">1</mn></mrow> <mrow><mi mathvariant="italic">n</mi></mrow> </mfrac> </math>
Nemeth Code	⠈⠿
Nemeth Example	⠈⠿⠭⠠ ⠭ ⠐⠅ ⠹⠂⠌⠝⠼
Marburg Code	⠯⠢
UEB Code	⠘⠢
BAUK Code	⠳⠢
Unified Spanish Code	⠨⠢
Woluwe Code	⠯⠢
Antoine Code	⠸⠡
Swedish Code	⠻⠠⠑
Finnish Code	⠳⠢
Italian Code	⠨⠢
Russian Code	⠫⠢
Unified Arabic Code	⠈⠿

Table 2: Example information of existential quantifier.

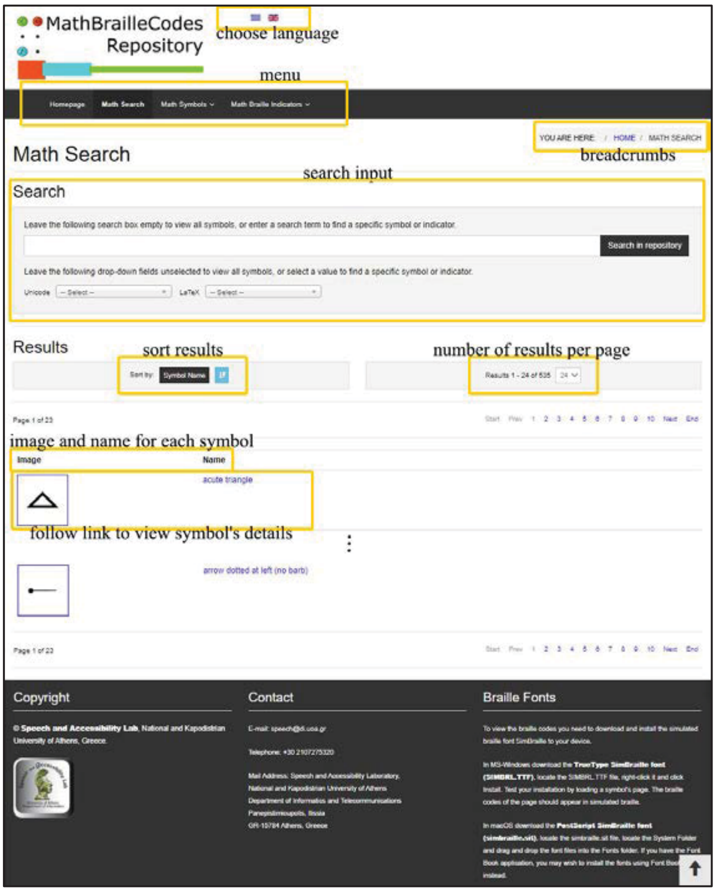


Figure 1: Screenshot of the BrailleMathCodes search page

5.3 Use

As stated previously, the repository is accessible, so visually impaired users can use their computer or smartphone along with a screen-reader and a refreshable braille display to access the repository. No login is required.

Users can either search for a specific code or browse through the math braille symbols and indicators. No conversion tool is used in the background, but learners of any code, both sighted and blind, can search for a term by its name, Unicode or LaTeX form, and then read how it is rendered in all the existing codes.

Search terms in BrailleMathCodes include math symbols, Unicodes, and braille math codes, but they cannot be expressed as mathematical formulae, as the examples in our collection are not large scale and the search algorithms used are string-based.

In the future, we plan to add more examples to make the repository a full learning environment for math braille codes, as well as to apply it in a mathematical information search and retrieval engine (Pattaniyil & Zanibbi, 2014) (Novotný, 2019) (Aizawa & Kohlhase, 2021).

6 Conclusions

The BrailleMathCodes repository meets the need of math braille users to search and find the codes for a symbol or indicator without the need to look it up in scattered sources. It contains the latest edition of each code and can be easily expanded to more codes and languages. We believe that as a dictionary, it will be a helpful tool for every braille math user, sighted, or visually impaired. Once all the codes and examples are gathered and imported, we plan to extend the repository's content to include math and code tutorials. At that point, we will need to enhance the search capabilities of the repository.

References

Access, N. (n.d.). Retrieved from NonVisual Desktop Access (NVDA) screen reader: http://www.nvda-project.org/

Aizawa, A., & Kohlhase, M. (2021). Mathematical Information Retrieval. *Information Retrieval and Access Tasks*, 169-185.

Alonso, F., Fuertes, J., González, Á., & Martínez, L. (2006). SBT: A Translator from Spanish Mathematical Braille to MathML. *ICCHP* (pp. 1207-1214). Linz: Springer.

American Printing House for the Blind. (1996). Estimated Number of Adult Braille Readers in the United States. *Journal of Visual Impairment & Blindness*, *90*(3), 287.

Archambault, D., & Guyon, F. (2011). UMCL Transcription Tools - Universal Maths Conversion Library. *AAATE Conference*, (pp. 416-423). Maastricht.

Becker, S., Stenberg, G., Lindqvist , B., & Trowald, N. (1992). *Matematiska Symboler i Punktskrift.* Uppsala: Department of Educational Research, School o£ Education.

Biblioteca Italiana. (2003). *Codice Braille Italiano.* Ciechi "Regina Margherita".

BrailleMathCodes repository. (2021). *Speech and Accessiblity Lab. University of Athens*. Retrieved from https://access.uoa.gr/mathbraillecodes

Britz, K., Epheser, H., & Pograniczna., D. (1992). *Internationalen Mathematikschrift für Blinde.* Marburg: Deutsche Blindenstudienanstalt e.V.

Bykov, A., Egorov, M., Morozova, F., & Proskurjakov, I. (1982). *System of Symbols for Math, Physics, Chemistry, and Astronomy: A Learning Aid.* Moscow: The All Russia Association for the blind.

Cao, X., & Yu, W. (2010). Using Content Management System Joomla! to Build a Website for Research Institute Needs. *International Conference on Management and Service Science* (pp. 1-3). Wuhan: IEEE.

Central Union of the Blind. (1979). *Finnish Braille Committee's Approved Mathematics, Physics and Chemistry Notation.* Helsinki: Central Union of the Blind.

Cranmer, T., & Nemeth, A. (1991). *A Uniform Braille Code.*

Croisetière, P., D'Amour, M., Ferland, P., & Rainville, G. (2008). *Code Braille Français Uniformisé Pour la Transcription des Textes Imprimés (CBFU).* Québec: Drummondville: Office des personnes handicapées du Québec.

Crombie, D., Lenoir, R., McKenzie, N., & Barker, A. (2004). math2braille: Opening Access to Mathematics. *ICCHP* (pp. 670-677). Berlin, Heidelberg: Springer.

Duxbury Systems, I. (2020, 12 31). Retrieved from Braille Translation Software from Duxbury Systems: https://www.duxburysystems.com

Egli, C. (2009). Liblouis - a uniiversal solution for Braille transcription services. *The Daisy 2009 Conference.* Leipzig.

Freedom Scientific, I. (n.d.). Retrieved from Job Access With Speech (JAWS): http://freedomscientific.com/Products/Blindness/JAWS

Kacorri, H., & Kouroupetroglou, G. (2013). Design and Developing Methodology for 8-dot Braille Code Systems. *Lecture Notes in Computer Science (LNCS), Volume 8011, pp. 331-340, 2013., 8011*, 331-340.
Karshmer, A., Gupta, G., & Geiiger, S. (1998). Reading and writing mathematics: the MAVIS project. *ASSETS* (pp. 136-143). Marina del Rey, CA: ACM Press.
Kingdom, B. A. (2005). *Braille Mathematics Notation.* Peterborough: RNIB.
Knuth, D. E. (1984). *Computers & Typesetting, Volume A: The TeXbook.* Boston, USA: Addison-Wesley.
Kouroupetroglou, G., & Florias, E. (2003). *Greek Braille Scientific Notation – Application in Information Systems for the Blind.* Athens, Greece: Education and Rehabilitation Center for the Blind. Retrieved from https://speech.di.uoa.gr/sppages/spppdf/BG1a.pdf
Library of Congress. (1990). *World Braille usage.* National Library Service for the Blind, & Physically Handicapped. Paris: UNESCO.
Mada, Q. A. (2021, 1 3). Retrieved from The Unified Arabic Braille Project: https://mip.qa/braille
Martos, A., Kouroupetroglou, G., & Argyropoulos, V. (2015). *International Conference on Enabling Access for Persons with Visual Impairment (ICEAPVI)* (pp. 115-119). Athens: University of Athens. Retrieved from https://access.uoa.gr/ICEAPVI-2015/proceedingsFiles/117-121.pdf
Martos, A., Kouroupetroglou, G., Argyropoulos, V., & Deligiorgi, D. (2014). Towards the 8-dot Nemeth braille code. *Lecture Notes in Computer Science, 8547*, 533-536. doi:10.1007/978-3-319-08596-8_83
Murillo-Morales, T., Miesenberger, K., & Ruemer, R. (2016). A LaTeX to Braille Conversion Tool for Creating Accessible Schoolbooks in Austria. *ICCHP* (pp. 397-400). Linz: Springer.
Nemeth, A. (1972). *The Nemeth Braille Code for Mathematics and Science Notation.* Louisville, Kentucky: AAWB-AEVH Braille Authority.
Notaert, G., Suij, M., & Vandekerkhove, E. (1975). *Handleiding Braillesymbolen wiskunde (Manual for braille mathematics).* Woluwe.
Novotný, V. (2019). *Document Representations for Fast and Accurate Retrieval of Mathematical Information.* Brno.
Pattaniyil, N., & Zanibbi, R. (2014). Combining TF-IDF text retrieval with an inverted index over symbol pairs in math expressions: the tangent math search engine at NTCIR. *NTCIR* (pp. 135-142). Tokyo, Japan: online.
Riles, R. (2004). *Research Study: Early Braille Education Vital.* Future Reflections.
Schweikhardt, W., Bernareggi, C., Jessel, N., Encelle, B., & Gut, M. (2006). LAMBDA: A European System to Access Mathematics with Braille and Audio Synthesis. *ICCHP* (pp. 1223-1230). Linz: Springer, Berlin, Heidelberg.
Suzuki, M., Kanahori, T., Ohtake, N., & Yamaguchi, K. (2004). An Integrated OCR Software for Mathematical Documents and Its Output with Accessibility. *ICCHP* (pp. 648-655). Linz: Springer.
The Unicode Consortium. (2000). *The Unicode Standard, Version 3.0.* Addison-Wesley Longman.
VirtueMart. (n.d.). *an Open Source E-Commerce solution*. Retrieved from http://virtuemart.net/
W3C Math Working Group, S. B. (1998). Retrieved from Mathematical Markup Language (MathML) 1.0 Specification: http://www.w3.org/TR/REC-MathML/
Whapples, M. (2007). Obtaining Braille mathematical documents. *MSOR Connections 7.3*, 18-21.
World Wide Web Consortium, W. (n.d.). *Web Content Accessibility Guidelines (WCAG) Recommendation.* Retrieved from W3C: https://www.w3.org/WAI/standards-guidelines/wcag/