Authors' preprint copy of:
Kacorri, H., Riga, P. and Kouroupetroglou, G., 2014, July. Performance Metrics and Their Extraction Methods for Audio Rendered Mathematics. In International Conference on Computers for Handicapped Persons (pp. 614-621). Springer International Publishing.

Performance Metrics and their Extraction Methods for Audio Rendered Mathematics

Hernisa Kacorri[1,2], Paraskevi Riga[1], and Georgios Kouroupetroglou[1]

[1]National and Kapodistrian University of Athens, Department of Informatics and Telecommunications, Panepistimiopolis, Ilisia, 15784 Athens, Greece
{c.katsori,p.riga,koupe}@di.uoa.gr
[2]The City University of New York (CUNY), Computer Science Program, The Graduate Center, 365 Fifth Ave, New York, NY 10016 USA
hkacorri@gc.cuny.edu

Abstract. In this paper we describe and compare three approaches to calculate structure- and content-based performance metrics for user-based evaluation of math audio rendering systems: Syntax Tree alignment, Baseline Structure Tree alignment, and MathML Tree Edit Distance. While the first two approaches require "manual" tree transformation and alignment of the mathematical expressions, the third approach obtains the metrics without human intervention using the minimum edit distance algorithm on the MathML representations of the math expressions. Our metrics and their extraction methods are demonstrated in a pilot user study evaluating the Greek audio rendering rules of MathPlayer with 7 participants and 39 stimuli. We observed that the obtained results for the metrics are significantly correlated between all three approaches.

Keywords: math audio rendering, metrics, accessibility, MathML, usability.

1 Introduction

Acoustic modality is one approach often favored by researchers to create an accessible platform for mathematics [1-6]. It is essential to evaluate and compare the accuracy of mathematical expressions provided by a rule-based system with speech output. Recently, we introduced the EAR-Math methodology [7], along with a number of associated novel metrics, to automatically calculate their performance through quantitative methods in audio rendered mathematics. However, one limitation is the requirement for "manual" steps required in computing these metrics. The focus of this paper is to explore and compare alternative ways to calculate the proposed metrics with and without human intervention. This would allow for more robust results with fewer human errors during the data processing steps. Furthermore, we present results of a pilot study in evaluating the Greek audio rendering rules with MathPlayer [8].

2 Related Work

Relatively few researchers have evaluated the quantitative performance of their math audio-access approaches. We reviewed the methods adopted for assessing the accuracy of these approaches as perceived by the users. In the user evaluation of MathTalk [9], participants' response was considered correct if the perceived formula retained over 75% of the rendered formula's content and its major structural features. However, no further details on the calculation of these accuracy metrics were available. The transcriptions of participants during the evaluation of I-Math [10] were compared linearly to the original textual expressions. The number of correct words and positions were evaluated using precision, recall, and F-score. Last, TechRead's evaluation [11] and the user study in [12] asked participants to choose among multiple un-weighted answers the one that best matched the audio stimuli. The differences in these evaluation approaches and metrics pose challenges for future researchers to compare their findings to previous work.

3 Performance Extraction for Audio Rendered Mathematics

Evaluation of Audio Rendered Math (EAR-Math), proposed in [7], is as an experimental methodology for user-based performance evaluation of mathematical expressions rendered by a rule-based system with audio output. Mathematical expressions, non-linear in nature, make it challenging to define a fine-grained error rate metric that describes a rendering system's performance. To measure the distance between the intended math expression and the one perceived by the users, EAR-Math proposes three error rate metrics. They are tailored to account for both content and structure and are derived by first aligning the two mathematical expressions and then computing the number of insertions, deletions, and substitutions in the perceived expression compared to the reference over the total number of elements in the reference expression for each of the three categories of elements:

$$SER = \frac{\#Ins + \#Del + \#Sub}{\#StructuralElem} \quad OER = \frac{\#Ins + \#Del + \#Sub}{\#Operators} \quad INER = \frac{\#Ins + \#Del + \#Sub}{\#Identifiers + \#NumericalValues}$$

- Structure Error Rate (SER) involves structural components of a math formula such as fractions, roots, and arrays.
- Operator Error Rate (OER) is focused on mathematical operators e.g. plus, minus, and times.
- Identifier and Number Error Rate (INER) represent the number of errors for identifiers and numerical values within the expression.

The key step in the calculation of the error rates for the performance metrics SER, OER, and INER is the comparison of the intended mathematical expression and the one perceived by the users. It requires a representation and alignment method that allows for fine-grained labeling of the elements in the expressions as structural, operators, numerical, and identifiers. We compare three approaches that use tree transformations and discuss their pros and cons with respect to the adopted alignment process. For each approach we illustrate an example of alignment for two math expressions: the intended expression rendered by the system (Reference) and the one perceived by a user (Perceived). The box in the Perceived expression indicates the part which the user was unable to include in the response.

Reference: $x = \frac{a^2+\sqrt{b-1}}{c}$ Perceived: $x_2 = \frac{a_3+\sqrt{b}}{}$

3.1 Syntax Tree Alignment

As in [7], the first approach is to draw the Syntax Trees for both reference and perceived expressions. Next, we ‘manually’ perform the alignment of the perceived tree to the reference tree and count their differences such as insertion, deletion, and substitution, to calculate the error rate metrics. In this approach, the leaves of the tree are considered identifiers or numbers and the inner nodes operators (e.g. +, -, *) or as structural elements (e.g. frac, sub, (,), cos, log, sum). Fig. 1 illustrates the application of this error rate extraction method on the example expressions. While identifying the operator, identifier, and number errors is a straightforward process, structural errors may be more challenging, especially when the correct structural element is improperly positioned in the tree. The approach counts this disposition as a double error, deletion and re-insertion. A drawback of this approach could be the required cognitive load and ambiguity in the alignment process especially when the perceived expression is sparse compared to the reference.

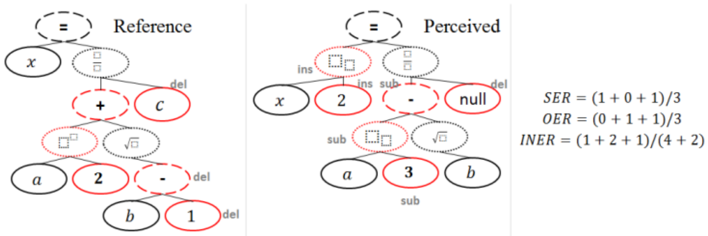

Fig. 1: Syntax Tree Alignment of the two math expressions.

3.2 Baseline Structure Tree Alignment

The Baseline Structure Tree [13], introduced for the recognition of handwritten mathematics, captures the layout of a formula without committing to any particular syntactic or semantic representation. Horizontally adjacent symbols in the expression, considered to be in the same region, are represented as ordered siblings in the tree. Thus, we can draw the tree unambiguously by exploiting its reading order even for a sparsely perceived math expression. Next, we ‘manually’ perform the alignment of the perceived tree to the reference tree and count their differences such as insertion, deletion, and substitution to calculate the error rate metrics. In this approach, the structural elements are defined by regions (e.g. center, over, under) and are placed at even tree depths, while the operators (e.g. frac, sub, +, -), numericals, and identifiers, i.e. all the printable elements, are positioned in odd depths. Fig. 2 illustrates the application of this error rate extraction method for the example expressions. The approach inherently counts the disposition of structural elements in the perceived tree. A drawback of this approach is that the resulting trees are more verbose than the Syntax Trees and is more tedious especially if the alignment and error rates are “manually” calculated.

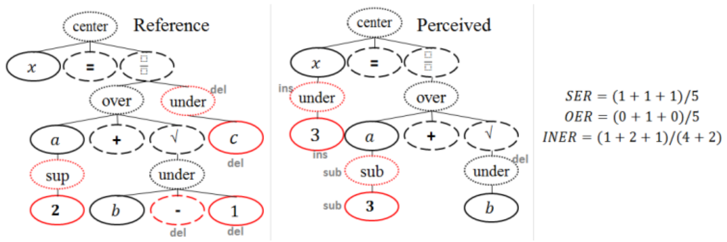

Fig. 2: Baseline Structure Tree Alignment of the two math expressions.

3.3 MathML Tree Edit Distance

Tree alignment is an optimization problem well defined in computational biology. It searches for the minimum operation number of node insertions, deletions and substitutions that are required to transform one tree into another, a measure called edit distance. To automate the calculation of the performance metrics, we apply the edit distance algorithm proposed in [14] to Presentation MathML encodings of the reference and perceived formulae. In particular, we parse the Presentation MathML trees into regular expressions and use the edit distance implementation in [15] (RTED) to obtain the optimal alignment of the trees. We further modified the RTED implementation to print the labels of the nodes in the tree and to assign them to the error rates categories: structural (<mrow>, <mfrac>, <msup>, etc.), operators (<mo>), and identifiers or numbers (<mi> and <mn>). Fig. 3 illustrates the application of this error rate extraction method for the example expressions.

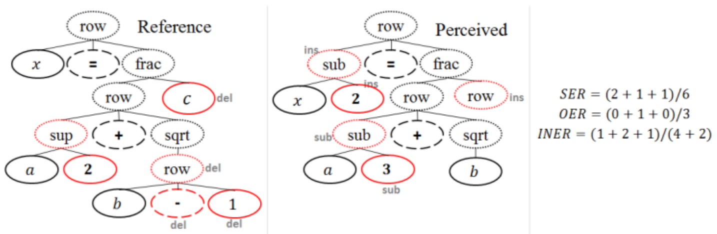

Fig. 3: MathML Tree Edit Distance Alignment of the two math expressions.

A requirement of this approach is that similar MathML notations should be used for both formulae, e.g. the same tool generates them. If the MathML code is already available for the reference formula, then it is suggested to copy and modify it accordingly for the perceived formula. One of the major advantages of this approach is that it provides more robust results while minimizing the required human intervention in the calculations of the proposed metrics.

4 Comparison of Extraction Methods in a Pilot User Study

While future researchers focusing on the pros and cons may choose one of the above approaches to evaluate their system performance, their results should be comparable. Therefore, we investigate the relationship of the obtained results from all three approaches within a pilot user study.

4.1 Pilot User Study

We revisited the results obtained from the pilot user study in [7] and calculated the metrics based on each of the approaches. The mathematical expressions, based on the set of formulae for ASTeR demonstration [16], were rendered through the Dimitris voice of Acapela Greek Text-to-Speech [17] driven by MathPlayer with lexical and prosodic cues.

Of the 7 participants recruited for the study: 2 were congenitally blind and 5 were sighted. All participants had been exposed to more complex mathematical expressions than the stimuli. There were 5 men and 2 women of ages 20-34 (average age 25.9). During the study, participants would listen to mathematical expressions and write down the perceived formulae. They were allowed to make changes to their initial guess two more times. We collected all three perceived versions for each of the expressions. After the experimental session, blind participants would read their notes and describe their answers to a sighted member of the team who would then visualize the expression in two-dimensions.

4.2 Results

As in [7], we 'manually' drew and aligned the trees for the perceived and reference formulas for the first and second approach and recorded the errors. For the third approach the MathML tree of the reference math expression was already available in MathML. To create the perceived MathML tree we copied the reference tree and edited it accordingly to the users' answers. Then, the alignment of the MathML trees was automatically performed as described in section 3.3. We calculated the error rates for the aggregated elements among all stimuli, as shown in Table 1.

Table 1: Overall error rates in the stimuli set for the three approaches.

	Syntax Tree (1)			*Visual Tree (2)*			*MathML Tree (3)*		
	SER	OER	INER	SER	OER	INER	SER	OER	INER
1st Attempt	0.18	0.12	0.11	0.16	0.17	0.14	0.24	0.14	0.13
2nd Attempt	0.1	0.06	0.04	0.08	0.09	0.06	0.14	0.07	0.07
3rd Attempt	0.07	0.04	0.02	0.04	0.06	0.03	0.09	0.05	0.04

Fig. 4 shows the distribution of the SER, OER, and INER for all three methods as boxplots with whiskers at the 1.5 IQR (inter-quartile range). To aid the comparison, mean values, illustrated with a star, are added as labels at the top of each plot. For all attempts, we observe that the MathML Tree Edit Distance results share similarities with the Syntax Tree results though the former shows higher variance and mean. We speculate this is due to the inherent verbosity of the MathML representation compared to the abstract representation of the Syntax Trees. The Visual Trees approach seems to have shifted the weight of the structural errors to the operators. This makes sense given that the structure in the mathematical expression is now represented by the layout and not by the semantics. We also observe that participants tend to improve their performance the second and the third time they listen to the mathematical expression and this is captured by all three approaches. This suggests that the audio rendering might have been accurate, but other factors (such as audio memory and familiarity with the system) may have an effect on the results and should be taken into account when designing the experiment.

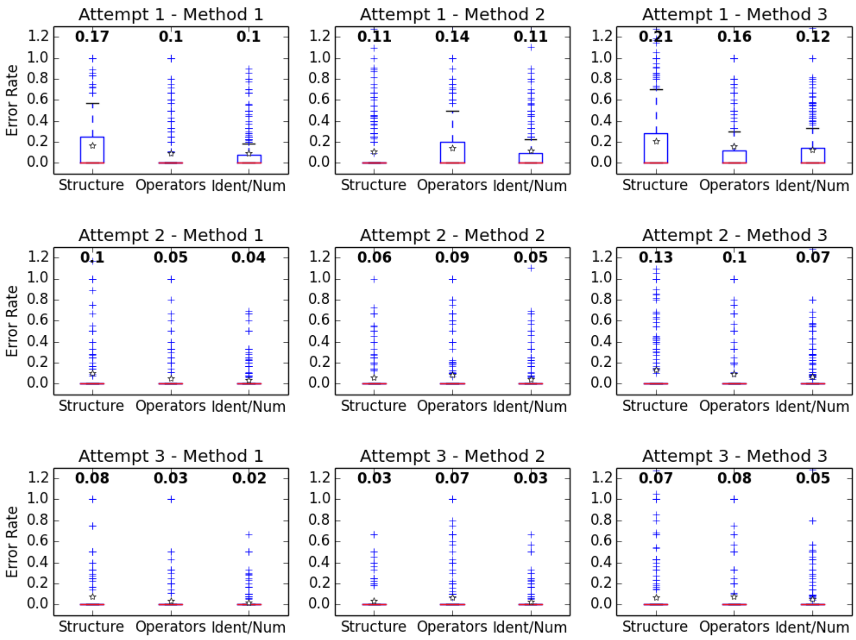

Fig. 4: Error rate distributions for user attempted responses by methods (Method 1: Syntax Tree, Mehod 2: Baseline Structure Tree, Method 3: MathML Edit Distance)..

We performed correlation analysis to the obtained results to further investigate the relationship of the error rates across the approaches. Table 2 displays the Spearman's rho correlation values for SER, OER and INER. The rho value is shown for each pair of approaches by error rate category. We note that all correlations were found to be significant. This indicates that future researchers may choose either of the approaches to calculate their metrics. We also observe that the aforementioned speculations about the similarities of the first and third approaches are supported. There is significantly strong correlation between SER1 and SER3. While the second approach shifts the structure errors to operators, the identifiers and numbers are almost identical to the first approach. This is also supported by the significantly strong correlation between their INER metrics.

Table 2: Correlations (Spearman's rho) between the three approaches. All values were found to be highly significant ($p < 0.001$).

	Method 1 & 2	*Method 1 & 3*	*Method 2 & 3*
SER	0.79	0.831	0.662
OER	0.73	0.529	0.789
INER	0.975	0.668	0.674

5 Conclusions

This paper has described and compared three approaches to calculate the EAR-Math performance metrics for user-based evaluation of math audio rendering systems: Syntax Tree alignment, Baseline Structure Tree alignment, and MathML Tree Edit Distance. While the first two approaches require "manual" tree transformation and alignment of the mathematical expressions, the third approach automatically derives the metrics using the minimum edit distance algorithm on the MathML representations of the math expressions. Our metrics and their extraction methods are demonstrated in a pilot user study evaluating the Greek audio rendering rules of MathPlayer with 7 participants and 39 stimuli. We observed that the obtained results for the metrics are significantly correlated between all three approaches.

This research makes three key contributions. First, it provides guidance for researchers conducting user-based evaluation studies to: (i) measure the performance of math audio rendering systems against a baseline, (ii) compare alternative systems, or (iii) iteratively evaluate improvements/styles. Second, it suggests that future researchers may use any of three ways to calculate the proposed metrics since they were found to be highly correlated. Finally, it provides results from a pilot study comparing the three alternative approaches to derive the metrics. This allows future researchers to compare and interpret results across studies irrespective of the extraction approach for the proposed metrics.

Acknowledgments. This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) under the Research Funding Project: "THALIS-University of Macedonia-KAIKOS: Audio and Tactile Access to Knowledge for Individuals with Visual Impairments", MIS 380442.

References

1. Freitas, D., Kouroupetroglou, G.: Speech Technologies for Blind and Low Vision Persons. Technology and Disability, 20, 135–156 (2008)
2. Archambault, D., Stoger, B., Fitzpatrick, D. and Miesenberger, K.: "Access to scientific content by visually impaired people". UPGRADE, Volume VIII (2), pp. 1-14, 2007.
3. Bates, E., Fitzpatrick, D.: Spoken mathematics using prosody, earcons and spearcons. Lecture Notes in Computer Science, 6180, 407-414 (2010) Springer Berlin Heidelberg
4. Kouroupetroglou, G., Kacorri, H.: Deriving Accessible Science Books for the Blind Students of Physics. Proceedings of the American Institute of Physics, 1203, 1308-1313 (2010)
5. Karshmer A., Gupta G., Pontelli E., Miesenberger K., Ammalai N., Gopal D., Batusic M., Stöger B., Palmer B., Guo H.: "UMA: A System for Universal Mathematics Accessibility". In proceedings ASSETS 2004, the 6th Intern. ACM SIGACCESS Conference on Computers and Accessibility, October 18-20, 2004, Atlanta, Georgia, USA, Volume 196, p. 55, ACM, 2004.
6. Tsonos, D., Kacorri, H., Kouroupetroglou, G.: A design-for-all approach towards multimodal accessibility of mathematics. Assistive Technology Research Series, 25, 393-397 (2009)
7. Kacorri, H., Riga, P., Kouroupetroglou, G.: EAR-Math: Evaluation of Audio Rendered Mathematics. Proceedings of the 8th Int. Conf. on Universal Access in Human-Computer Interaction. 22-27 June 2014, Crete, (2014) *in press*
8. Soiffer, N.: "A flexible design for accessible spoken math". Lecture Notes in Computer Science, vol. 5616, pp. 130-139, 2009.
9. Stevens, R.D.: Principles for the design of auditory interfaces to present complex information to blind people. Doctoral dissertation, University of York (1996)
10. Wongkia, W., Naruedomkul, K., Cercone, N.: I-Math: an Intelligent Accessible Mathematics system for People with Visual Impairment. Computational Approaches to Assistive Technologies for People with Disabilities, 253, 83-108 (2013)
11. Fitzpatrick, D.: Towards Accessible Technical Documents: Production of Speech and Braille Output from Formatted Documents. Doctoral dissertation, Dublin City University (1999)
12. Murphy, E., Bates, E., Fitzpatrick, D.: Designing auditory cues to enhance spoken mathematics for visually impaired users. In Proceedings of the 12th international ACM SIGACCESS Conference on Computers and Accessibility pp. 75-82 (2010)
13. Zanibbi, R., Blostein, D., Cordy, J. R.: Baseline structure analysis of handwritten mathematics notation. Proc. of the 6th Int. Conf. Document Analysis and Recognition, pp. 768-773. IEEE, (2001)
14. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems, 18(6), 1245-1262 (1989) SIAM journal on computing,.
15. Pawlik, M., Augsten, N.: RTED: A Robust Algorithm for the Tree Edit Distance. PVLDB 5(4):334–345, (2011)
16. Raman, T.V.: Mathematics for computer generated spoken documents–ASTeR Demonstration. http://www.cs.cornell.edu/home/raman/aster/demo.html.
17. Acapela text-to-speech http://www.acapela-group.com/