



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Zavgorodniaia, Albina; Tilanterä, Artturi; Korhonen, Ari; Seppälä, Otto; Hellas, Arto; Sorva, Juha

Algorithm Visualization and the Elusive Modality Effect

Published in: ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research

DOI: 10.1145/3446871.3469747

Published: 16/08/2021

Document Version Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:

Zavgorodniaia, A., Tilanterä, A., Korhonen, A., Seppälä, O., Hellas, A., & Sorva, J. (2021). Algorithm Visualization and the Elusive Modality Effect. In *ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research* (pp. 368–378). ACM. https://doi.org/10.1145/3446871.3469747

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Algorithm Visualization and the Elusive Modality Effect

Albina Zavgorodniaia albina.zavgorodniaia@aalto.fi Aalto University Espoo, Finland

> Otto Seppälä otto.seppala@aalto.fi Aalto University Espoo, Finland

Artturi Tilanterä artturi.tilantera@aalto.fi Aalto University Espoo, Finland

Arto Hellas arto.hellas@aalto.fi Aalto University Espoo, Finland Ari Korhonen ari.korhonen@aalto.fi Aalto University Espoo, Finland

Juha Sorva juha.sorva@aalto.fi Aalto University Espoo, Finland

ABSTRACT

The modality effect in multimedia learning suggests that pictures are best accompanied by audio explanations rather than text, but this finding has not been replicated in computing education. We investigate which instructional modality works best as an accompaniment for algorithm visualizations. In a randomized controlled trial, learners were split into three conditions who viewed an instructional video on Dijkstra's algorithm, with diagrams accompanied by audio, text, or both. We find neither a modality effect in favor of the audio condition nor a verbal redundancy effect in favor of using only a single modality rather than both. Taken together with earlier research, our findings suggest that the modality effect is difficult to apply reliably and computing educators should not rush to integrate audio into visualizations in expectation of the effect. We discuss theoretical viewpoints that future research should attend to; these include alternative part-explanations of the modality effect and attention-based models of working memory, among others.

CCS CONCEPTS

• Social and professional topics \rightarrow Computing education.

KEYWORDS

modality effect, verbal redundancy effect, data structures and algorithms, algorithm visualization, algorithm animation, Dijkstra's algorithm, replication

ACM Reference Format:

Albina Zavgorodniaia, Artturi Tilanterä, Ari Korhonen, Otto Seppälä, Arto Hellas, and Juha Sorva. 2021. Algorithm Visualization and the Elusive Modality Effect. In Proceedings of the 14th ACM Conference on International Computing Education Research (ICER 2021), August 16–19, 2021, Virtual Event, USA. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3446871.3469747

ICER 2021, August 16-19, 2021, Virtual Event, USA

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 978-1-4503-8326-4/21/08...\$15.00 https://doi.org/10.1145/3446871.3469747

1 INTRODUCTION

The *modality effect* or *modality principle* [11, 41, 42] is the oftencited finding in educational psychology that audio explanations of diagrams or other pictures tend to be more effective than textual ones. The related *verbal redundancy effect* [29, 30] suggests that accompanying pictures with the same information as both audio and text tends to be less effective than either explanation alone. The empirical evidence for these principles is plentiful but comes with a number of caveats and has not been replicated in a computing education context [41, 42].

The potential implications of the modality and verbal redundancy effects on computing education are substantial. Teachers in formal education routinely employ pictures accompanied by explanations in several ways, including classroom teaching, books and ebooks, and instructional videos. Moreover, learners informally study computing from a variety of materials that combine pictures with text and/or audio; online videos in particular are popular.

Since program code has some diagram-like qualities, it has been hypothesized [20, 44] that the modality and verbal redundancy effects could apply to explanations of program code. Should that be so, the effects' implications for computing education would be greater still. After all, computing education is replete with example programs that come with textual or auditory explanations.

However, the modality and verbal redundancy effects have not been demonstrated in computing education research (CER). As far as we are aware, two studies have examined whether the effects manifest when explaining program code in text and/or audio [44, 69]. Neither found evidence for the effects. There are many possible explanations for these findings. Thus, replication of the modality and verbal redundancy effects in computing education remains an open problem—and, importantly, it remains unclear which kinds of computing contents and contexts the effects might apply to.

One of the explanations offered for the lack of a modality effect in CER studies to date is that those studies use program code instead of a picture. There is much that is unknown about how humans cognitively process code, and if that processing is not sufficiently similar to the processing of diagrams, that alone could account for the past results.

Our study looks into that suggested explanation. The study's primary motivation is to find out whether the modality effect can be observed on materials that teach computing content and are unequivocally diagrammatic. The materials in this study do not explain program code but a graph algorithm illustrated at a high

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

level of abstraction; in this respect, the present study differs from the earlier modality studies in CER. Our secondary motivation is to explore the modality effect in the context of algorithms education.

To put our results in a nutshell, we did not find evidence for the modality effect or the verbal redundancy effect. In addition to the empirical results that we present, we contribute a discussion on plausible theoretical explanations for this lack of an effect in this and prior CER studies.

2 RELATED WORK

2.1 Cognition and Instructional Modalities

2.1.1 Theoretical Basis of the Modality Effect. The modality effect in the most common sense of the term has its roots in research on multimedia learning, in particular the *Cognitive Theory of Multimedia Learning* (CTML) by Richard Mayer and his colleagues [41, 42]. Paivio's dual-coding theory [13, 48] and Baddeley's model of working memory [3] are key influences on CTML. Citing these influences, CTML identifies two information-processing "slave" systems in working memory: the *visuo-spatial sketchpad* and the *phonological loop*. The former system deals with visual information, whereas the latter system deals with auditory information. Both of these slave systems are very limited in capacity and duration and thus can easily overload during learning.

According to CTML, when a learner looks at pictures accompanied by textual explanations, the visual component of working memory may be overloaded, as the learner needs to split their visual attention between picture and text and to reconcile those inputs. When the explanations are provided in spoken form, the learner can exploit auditory working memory in parallel, making cognitive overload less likely. This opportunity to use both channels simultaneously during processing is the most prominent explanation for the empirical findings [24, 29, 35, 41, 42] that indicate that people tend to learn better from a combination of pictures and audio than from a combination of pictures and text—i.e., the modality effect. Following the lead of some other researchers (e.g., [55]), we will refer to this account of the modality effect as the *visuo-spatial load* explanation.

The verbal redundancy effect [29, 30, 41, 42] is closely related and likewise often explained in terms of visuo-spatial load. A learner who receives the same explanations of pictures as both text and audio runs the risk of overloading visual working memory; moreover, the learner needs to relate the two explanatory modalities to each other, which is a further burden on working memory.

Another partial or alternative explanation for the effects involves the limitations of sensory perception [7, 12, 43]; perception precedes working-memory processing but influences what is available for processing. Delays between perceiving parts of a presentation make it less certain that all relevant inputs will be noticed and successfully integrated with each other. Specifically, a combination of text and a picture necessarily involves at least some shifting of attention between the two, whereas one can listen to spoken words while viewing a picture, without any delay. This is known as the *temporal contiguity* explanation.

The core pedagogical recommendations related to modality that arise from CTML are to prefer audio explanations over textual ones and especially to avoid using both at the same time when explaining pictures. However, this advice comes with significant qualifications.

2.1.2 Boundary Conditions. Although there is evidence for the modality and verbal redundancy effects from many studies, other studies have reported *not* finding support for the effects or, in some cases, finding a reverse modality effect where textual explanations outperform audio. This has led to the suggestion that there are a number of conditions on when the effects apply. Some of the conditions highlighted in the literature [24, 29, 35, 41, 42, 53] are:

- Mutual unintelligibility: If a picture can be understood without the verbal explanations, or the explanations without the picture, the modality effect does not apply and the redundant information may hinder learning. Intelligibility depends in part on prior knowledge.
- (2) Appropriate complexity: If the materials are too simple (or too complex) for the learners, there will be little difference between instructional modalities. The modality effect is more likely when learning materials are sufficiently complex in the sense of having various interacting elements, so that working memory is strained.
- (3) Sufficient signaling: "When a picture is accompanied by spoken explanations, learners usually search through the picture to locate its components that are referred to in the text while holding auditory information in working memory. If this search is complex [...], the modality effect may not be demonstrated unless appropriate visual signals are used to assist the learner in locating the relevant sections of the picture." [29]
- (4) Segment length: The effectiveness of spoken explanations suffers (in comparison to textual ones) if the explanatory segments are long or incoherent.
- (5) Limited learner control: If the learner controls when to resume a presentation and receive the next explanatory segment, the benefits of textual explanations are emphasized, as the learner can take their time integrating the textual and pictorial information. The modality effect is more likely without that control, such as in a video that the learner cannot pause (or fails to pause even though that would be beneficial) or in a mass lecture.
- (6) *Familiar language:* If the learners are not fluent in the language of instruction, text can be relatively effective, negating the modality and verbal redundancy effects.

2.1.3 Alternative Theoretical Explanations. Visuo-spatial load and temporal contiguity are the most common and well-known explanations of the modality effect. However, the visuo-spatial load explanation especially has been criticized on theoretical and empirical grounds, and other explanations of the effect have been put forward [15, 53, 55–57, 59]. The plurality of part-explanations and the varied assortment of boundary conditions have led some scholars to suggest that what is known as the modality effect in CTML may not be best treated as a single phenomenon but as a confluence of distinct perceptual and cognitive phenomena [53, 56].

For the majority of this article, we will not break with CTML's tradition of treating the modality effect as a unitary phenomenon,

which tradition is also characteristic of CER to date. We will, however, return to the criticisms in Discussion (Section 6) as we consider explanations for our findings and opportunities for future work.

2.2 The Modality Effect in Computer Science

As we directly build on and partially replicate two modality studies in CER [44, 69], we will describe those study designs in some detail.

2.2.1 The Morrison study. Morrison [44] recruited 61 participants with little programming knowledge from introductory CS courses at U.S. universities. Each participant was arbitrarily assigned to one of three conditions: Audio, Text, or Both. All participants viewed three instructional videos that explained Python code using one or both modalities; the explanations' modality was the only difference between the conditions. Each video presented a toy "real-world" problem, its solution as example code, and a trace of the program's execution. The videos were 5, 23, and 12 minutes long and involved assignment, nested selection, and iteration, respectively. Two of the programs incorporated in-code comments (i.e., some textual explanations). Learners could not control the videos' pacing, with the exception of a single pause that ended when the learner wished. The elements of program code that were being explained (as speech and/or text) were visually highlighted. The participants were posttested on their understanding of the purpose of the program and its components, on their ability to trace the program's segments, and on transfer to a similar program.

In accordance with prior research outside of CER, Morrison hypothesized that students in the Audio condition would outperform the students in the Text condition and would experience lower cognitive load, thus demonstrating the modality effect. In line with the verbal redundancy effect, the Both condition was hypothesized to perform the worst. However, the differences between the three conditions did not even descriptively support the hypotheses. Subjectively assessed cognitive load scores did not support the hypotheses, either.

2.2.2 The Zavgorodniaia et al. study. Morrison [44] suggested that a possible explanation for her results was the learning materials' excessive complexity. Zavgorodniaia et al. [69] replicated Morrison's study using a less challenging video tutorial and a larger and different cohort of learners. They crowdsourced 186 paid participants with little to no self-reported programming experience and randomly assigned each to Audio, Text, or Both. Every participant viewed a single, system-paced, 24-minute video about the basics of variables and assignment in Python. The video explained a sequence of tiny, decontextualized, uncommented example programs in detail and directly addressed some known misconceptions about variables and assignment. In this study, like Morrison's earlier, the only difference between conditions was the modality, and relevant parts of code were visually signaled for clarity.

In Zavgorodniaia and colleagues' study, too, the groups' performance on a post-test did not suggest a modality or verbal redundancy effect for instructional videos that explain program code.

Is Code a Diagram? 2.3

There are many plausible explanations for why the modality effect did not occur in the CER studies. The learning materials may



H* 0

L 5

Та ~ v ∞ ~ к

Next, the algorithm searches for the shortest available route it has seen so far. From the table, we can see that it is to the node Tu with distance of 2. The route is marked as shortest with grey.

Figure 1: A still from an algorithm animation video that explains Dijkstra's algorithm. An example graph is shown, along with a priority table that tracks the algorithm's progress starting from the source node (H).

have inadvertently violated the boundary conditions listed above; the assessment may have lacked validity; the study participants' motivation may have been limited and they may have engaged insufficiently with the materials. Another explanation is that one of the fundamental assumptions behind these studies does not hold, namely, that program code is not sufficiently similar to a diagram in cognitive terms.

The original modality effect concerns pictures accompanied by words, which CER has interpreted as *programs* accompanied by words: "cognitive science research has shown that programs are read more like diagrams than like prose" [20]. There is indeed evidence that visual layout affects code-reading efficiency [27] and that expert programmers visually imagine program structures while designing [52]; visual attention during program reading depends complexly on expertise and the program [10, 50, 51]. This evidence to date is fragmentary, however, and an overall picture has not yet formed of how "picture-like" or "language-like" programs are-and how that depends on notations and expertise. Highlighting some of these issues, Zavgorodniaia et al. [69] prompted CER to investigate the modality effect on materials that are more unambiguously diagrammatic than program code is. We do that in an algorithm visualization context.

Algorithm Visualization 2.4

Algorithm visualization is the use of images for conveying information about data structures and algorithms. In teaching, algorithm visualizations may be employed in various ways: for instance, a visualization may simply illustrate an example or it may form the basis of an interactive learning activity [45, 62]. In contrast to program visualizations [63], which illustrate a concrete program, algorithm visualizations have a relatively high level of abstraction.

Two kinds of algorithm visualizations feature in our study.

An algorithm animation is an instructional video or sequence of images that shows how an algorithm operates on a data structure step-by-step. A screenshot from an algorithm animation appears in Figure 1. That example is from an instructional video that illustrates Dijkstra's algorithm and is used in the present study.



Figure 2: A visual algorithm simulation exercise on Dijkstra's algorithm, in progress. The software displays a weighted graph with nodes A–O. Only the source node A was initially highlighted, but the learner has since (correctly) clicked three edges, which highlights the edges with wider, yellow lines; the corresponding nodes also get a highlight. If the learner correctly continues from here, they will eventually highlight the single-source shortest-paths tree.

A visual algorithm simulation exercise [34, 39] similarly displays a data structure but requires the learner to reproduce the steps of the algorithm by interacting with the visualization. The learner's trace of the algorithm on given input can be automatically compared to a model solution to generate feedback and/or a summative grade. The screenshot in Figure 2 also illustrates Dijkstra's algorithm but is taken from a visual algorithm simulation exercise. We adapted this exercise from the OpenDSA textbook [33] and used it as part of our experiment's assessment step as described in the next section.

There is evidence of the success of algorithm visualization as a pedagogical tool [45, 62]. Nevertheless, challenges in algorithms education remain, such as the misconceptions that students are known to have despite visualization-based pedagogy [16, 21, 23, 25, 32, 47, 49, 60]. Support for audio narration is rare in current program visualization [63] and algorithm visualization systems, but there are exceptions; for example, the JSAV library supports audio [67], which feature has seen some limited use in the OpenDSA ebook [9].

The evidence from educational psychology in support of the modality effect suggests that (system-paced) algorithm and program visualizations would be more effective if accompanied by audio explanations, but there has been little research around this suggestion.

3 RESEARCH OBJECTIVES

We ask: Is there a modality effect and a verbal redundancy effect when explaining diagrammatic algorithm visualizations on video? That is, we test these hypotheses:

- H1 Learners who receive audio explanations of an algorithm visualization learn better than learners who receive the same explanations as text.
- H2 Learners who receive identical explanations of an algorithm visualization simultaneously as audio and text learn less than learners who receive the same explanations as audio only or text only.

Our study focuses on learners who do not have disabilities and who are beginners in graph algorithms.

4 METHODS

University students were presented with an instructional video that explained Dijkstra's algorithm on a high level of abstraction, in English, accompanied by a visualization.

Three variants of the video were created: one with explanations in the audio modality only, another in the textual modality, and a third that used both modalities simultaneously. The students were randomly assigned to view one of the three videos; we will henceforth refer to these conditions as the *Audio* group, the *Text* group, and the *Both* group. After viewing their assigned video, each group did the same set of assessment tasks.

Figure 3 gives an overview of our experiment, and the subsections below provide more details.

4.1 Procedure

The experiment was conducted at Aalto University, a European research institution, in November 2020. It was organised fully online in the A+ learning management system [31]. This platform supports multiple-choice questions, open-ended answers, and visual algorithm simulation exercises from the OpenDSA ebook [33]. For our experiment, A+ displayed the videos, presented the pre-survey and assessment items, and collected participants' responses.

Students from local computing courses (described below) participated in the study. As a reward for participation, each student was granted a predefined amount of course credit that did not depend on performance. During the experiment, the participants first filled a background survey and indicated whether they allowed their response data to be used in research. They then viewed a randomly assigned variant of the instructional video; the only difference between the variants was instructional modality. Each student could only watch the video once. The assessment that followed the video was identical for all groups. The videos and assessment are further detailed in Section 4.3.

The students had a time window of three weeks to visit the research site and participate in the experiment. Apart from this, there were no time constraints. We could not control whether students discussed the experiment, used external learning materials, experienced distractions, or multi-tasked while participating. (The fully online setup was in part due to the ongoing COVID-19 pandemic.)

4.2 Participants

Students from two large-class courses were invited to the study—a *Basics in Programming* (BP) course (over 1000 students a year) and a *Data Structures and Algorithms* (DSA) course (ca. 350 students a year). Both are service courses that target non-CS majors. The BP students studied introductory programming in Python and had no



Figure 3: An overview of our experiment and participant counts. Activities are shown in blue and participant counts are shown in orange circles. The gray circles at the bottom indicate discarded data.

prior experience with graphs and algorithms. The DSA students had previously passed the BP course (or similar) as a prerequisite. Graph algorithms are taught in DSA, but only at the end of the course; the experiment was timed so that it preceded the teaching of these topics.

In total, 173 students from the two courses volunteered. As shown in Figure 3, some of the data was excluded from analysis, either because the participant did not give research consent or because they were already familiar with shortest-paths spanning trees. Familiarity was assessed with a combination of DSA course logs and self-reports in our background survey.

This left us with 77 participants. Due to the uneven number of participants from the two courses (67 BP students; 10 DSA) and the fairly low number of participants overall, we pooled all the data together.

According to self-reported demographics, 64 participants were of age 18–24, 10 of age 25–34, and 3 were 35 years or older. 36 of the participants were female, 41 male; no other genders were reported. 58 spoke Finnish natively, 5 Swedish, and 14 spoke some other first language. The self-rated English proficiency of the non-native English speakers was at the upper-intermediate level or higher.

4.3 Instructional materials

The topic of the instructional video was Dijkstra's algorithm [19]. The video covered the algorithm on an abstract level, explaining its purpose, principles, and—especially—the order of steps that the algorithm follows as it covers a graph. This explanation did *not* cover the algorithm's implementation as code or graph representation as an adjacency matrix; nor did it go into graph theory or performance characteristics. Complex terminology was avoided, such as "priority queue," but "node," "edge," and "weight" were introduced and used as key terms. In addition, selecting the next node to be visited (without the concept of priority queue) was explained by illustrating the priorities in a simple table. The next node selected into the minimal spanning tree was determined in alphabetical order in case there were two or more otherwise equal alternatives. The expectation was that after watching the video, the participants might be able to answer questions about and/or to simulate the algorithm's stepwise behavior on a given (but previously unseen) example graph.

The video was eleven minutes long and consisted of two parts.

The first part began with a brief explanation of a graph and its structure. We based the explanation on an analogy with cities (nodes) connected by roads (paths), each having a distance (weight). That explanation was followed by a concrete example with a detailed description of how, and in what order, the algorithm adds routes to a spanning tree; Figure 1 shows a screenshot as seen by the Text and Both groups. The term shortest-paths spanning tree was mentioned but—to avoid unessential terminology—only at the end. The first part finished with a quick review of the order in which the nodes and edges are dealt with.

The second part of the video commented on another example graph in similar fashion. This part also demonstrated a *relaxation* in which a new shorter path to a previously seen node is found; we informally brought up this key idea (but not the term) a number of times across the video, but it was illustrated only once with animation. The part also introduced a case where the new path is exactly the same length as the previous one, in which case the priority table was not updated. Another recap concluded the second part and the video.

We designed the video with the modality effect's boundary conditions (Section 2.1.2 above) in mind:

- The visualizations (graphs and tables) were *mutually un-intelligible* with the explanations (audio and/or text). The learners would surely have failed had they been presented with just the visualizations or just the explanations.
- We are confident that the lesson had high enough *complexity* in the sense of element interactivity. At numerous steps, multiple nodes and edges required consideration as a whole. The priority table was displayed so that the learner would not have to track route weights and visited nodes in working memory, which might easily lead to cognitive overload.
- Each modality variant of our video made heavy use of *sig-naling* to ease visual navigation. Each step was supported by

highlights (color-blind friendly). Moreover, arrows pointed at specific table rows while they were being explained, and routes on the graph were similarly highlighted while they were being compared.

- Each segment of the explanation (per slide) had a fairly short length between 3 to 44 words, with an average of 14.7 and a median of 11.
- The video was *system-paced*, and the participants could not stop or rewind it. The narration speed was approximately 146 words per minute.
- Being familiar with this student population, we expected their English skills to be good enough that they would not have trouble with a well designed presentation in English.

4.4 Assessment

Immediately after the video, assessment tasks were revealed. The assessment consisted of thirteen transfer questions and a visual algorithm simulation task.

4.4.1 Transfer Questions. The transfer questions tested near transfer to cases similar to those that the students had seen on video. Each question required the student to apply Dijkstra's algorithm to a given graph and to consider the steps that the algorithm takes through the graph, in order. Some questions, which involved a very simple graph, asked the student for the full shortest path from one node to another. Other questions prompted the student for only the next node or the previously visited node, or to select which of given nodes would be visited before the others. All but the last question were multiple-choice. The last question differed from the others: it asked the student to describe all the algorithmic steps in their own words and to produce an ordered list of nodes that corresponds to the shortest-paths spanning tree (but the question did not feature the term).

Each question was evaluated with a binary grade: one point for a correct answer and zero for an incorrect one. The last question, even though the most demanding, was evaluated similarly, so that only a fully correct list of nodes scored a point.

4.4.2 Visual Algorithm Simulation Task. Each student was asked to solve a visual algorithm simulation task three times, meaning that they had to simulate the algorithm's behavior on three given graphs in sequence. The first graph is shown in Figure 2, and the other two were similar. The graphs were identical for all students. The simulation task differed from the transfer questions in that it asked the students to select edges rather than nodes. That is, the students had to recreate the behavior of Dijkstra's algorithm by clicking edges in the order that the algorithm follows. The visualization did not include the priority table shown in the video.

The graphs were designed to test all the essential features of Dijkstra's algorithm such as (i) relaxation—the shortest path to a certain node must be updated as a new path is discovered that is shorter than a previously discovered one; (ii) no relaxation—a new path to a certain node is discovered but discarded as being longer or equal in length with a previously discovered one; and (iii) nodes that are unreachable from the source node.

We wanted to keep the graphs small enough that the task would not be too laborious. We used a total of 15 nodes and 17 edges, which are enough to produce the properties listed above [65, pp. 38–40, 46–49]. The solution for each graph had the same number of steps (11); this was deliberate to simplify quantitative analysis.

Since the simulation task was meant as a post-instructional test rather than a learning experience, students received no feedback on their solutions.

The order in which each student clicked the edges was recorded, along with timestamps. This data was collected with an algorithm animation recorder developed for the system [40].

During data analysis, we assigned each participant a score for the simulation task. This score was the number of consecutive correct steps (edge clicks) counted from the beginning; the maximum of 11 equals the the number of edges in each graph's shortest-paths spanning tree.

4.5 Statistical Tests

We followed typical methodology for inferential statistics. To determine differences between groups, we used a one-way ANOVA for normally distributed values and a non-parametric Kruskal–Wallis test for non-normal distributions. Pairwise differences between groups were examined with a Tukey–Kramer post hoc test. For checking normality, we used the Shapiro–Wilk test, which was supported by Levene's test for homogeneity. All these tests were conducted using SPSS Statistics (version 2.6).

For each of the tests, where applicable, we report an effect size (eta-squared; η^2), a confidence interval, and a *p*-value, all of which contribute to our interpretation of the data. Where the eta-squared statistic estimates an effect size as negative, we report it as it is, rather than as zero [46].

5 RESULTS

5.1 Transfer Questions

Figure 4 shows the distribution of total scores on the thirteen transfer questions for each modality group, and Table 1 shows the mean and median scores for each group. Figure 5 further illustrates the percentage of correct answers on each question.

The total scores on the transfer questions passed checks for normality (Shapiro–Wilk test; p = 0.2) and homogeneity (Levene's test; p = 0.4) and did not contain significant outliers.

Examination of Figure 5 confirms that very few participants provided fully correct descriptions of the shortest-paths spanning trees in Question 13, which was open-ended and particularly complex. Another question with low scores is Question 6, which was fairly complex and, in hindsight, may have been misleadingly phrased.

Since the complex and open-ended Question 13 was different in nature from the other questions and had particularly low scores, we decided to check for a difference between the modality groups twice: with that question included and without it. Either way, the results from a one-way ANOVA do not suggest that any observed difference between the groups' performance was conditioned by instructional modality, especially given the negligible effect size (first twelve questions: F(2,75) = 1.3, p = 0.3, $\eta^2 = 0.03$, 95% confidence interval [0.00; 0.13]; all thirteen questions: F(2,75) = 1.4, p = 0.3, $\eta^2 = 0.04$, 95% confidence interval [0.00; 0.13]). Tukey–Kramer post hoc tests did not suggest any pairwise differences between groups (first twelve questions: Audio-vs-Text p = 0.3,



Figure 4: Distribution of total scores on transfer questions.



Figure 5: Percentages of correct answers on each transfer question.

Audio-vs-Both p = 1, Text-vs-Both p = 0.3; similar for all thirteen questions).

5.2 Visual Algorithm Simulation Task

Table 1 shows the means, medians, and standard deviations of the students' performance on the visual algorithm simulation task, as well as the average time each student spent per simulated graph.

The learners' simulation scores were not normally distributed (Shapiro–Wilk test; $p = 3 \cdot 10^{-8}$), so we ran a Kruskal–Wallis test to compare the groups. As with the transfer questions, we did not find evidence of a modality effect in the simulation data ($\chi^2(2) = 2.1$, p = 0.3, $\eta^2 = 0.00$).

Students' times on the simulation task were likewise not normally distributed ($p = 2 \cdot 10^{-9}$), and we did not find a difference between the groups in time usage, either ($\chi^2(2) = 0.04$, p = 0.9, $\eta^2 = -0.03$).

There was a notable floor effect on the simulation scores, which limits the usefulness of the simulation data for present purposes. We will therefore base the discussion that follows largely on the results obtained from the transfer questions. Despite the floor effect, there was a strong positive correlation between the transfer-question scores and the simulation scores (Pearson's correlation for the first twelve questions: r(75) = 0.59, $p = 6 \cdot 10^{-8}$; similar for all thirteen questions).

simulation-task scores, and simulation times.				
		Mean	Median	Std
Audio (n=28)	Transfer questions (max. 13)	8.1	8.0	2.5
	Simulation task (max. 11)	2.7	2.0	2.1
	Time per graph simulated (s)	99	81	79
Text	Transfer questions	7.1	8.0	2.8

2.0

104

8.1

2.4

86

1.7

79

8.0

2.0

84

1.7

88

2.1

1.7

39

Table 1: Descriptive statistics on transfer-question scores, simulation-task scores, and simulation times.

6 **DISCUSSION**

(n=21)

Both

(n=28)

6.1 No Modality Effect Observed

Time per graph simulated

Time per graph simulated

Simulation task

Simulation task

Transfer questions

We found no support for there being a modality effect or verbal redundancy effect when Dijkstra's algorithm is taught using visualizations accompanied by textual and/or auditory explanations.

This result is analogous to the null results obtained previously by Morrison [44] and Zavgorodniaia et al. [69] on program code accompanied by explanations in different modalities. Compared to findings outside of CER, our result contrasts with the studies that have found evidence of a modality effect but is far from being unique in finding no support for the effect (see, e.g., [24, 53]).

Our findings are insufficient for concluding that no modality effect exists for algorithm visualizations in general or Dijkstra's algorithm in particular. Even less do these findings enable the broader conclusion that there is something unusual about computing itself that affects such results, which has been occasionally suggested as a possibility [44, 68, 70]. What we *can* conclude is that the earlier null results from modality studies in CER are not due to the nature of program code alone: irrespective of the cognitive characteristics of code, there is no guarantee of the effect even on CS materials that—unlike program code—are unambiguously diagrammatic.

The possibility of a false negative remains, and we must be cautious in interpreting our null result. Nonetheless, large effect sizes have been quoted for the modality effect [24], which point has been previously discussed by Schüler et al. [58], who note that a sample size of roughly n = 26 per group should have sufficient statistical power for detecting such an effect. Our study (just about) reaches this criterion. It is, of course, possible and perhaps even likely that any modality effect is more modest in size [53], and even a small effect size can be relevant in education if it is inexpensive to achieve, applies widely, and accumulates over time [14].

Questions of publication bias have been raised concerning the modality effect [36, 53]. We find these concerns reasonable. Through this study, we somewhat alleviate the problem.

6.2 **Boundary Conditions and Limitations**

As outlined in Section 4.3, we had the modality effect's boundary conditions in mind while designing the video. Nevertheless, we must consider the possibility that they have been violated or that our design has otherwise reduced the chance of a modality effect.

The *complexity* of materials is difficult to assess both in general and in our specific case. Relative to what the DSA course eventually

teaches about Dijkstra's algorithm, our short video had very modest goals: we taught only the basic idea rather than, for example, how to implement the algorithm. Nevertheless, the lesson did cover a significant amount of new content in a short time and demanded attention to several interrelated things: the graph, the explanations, the state of the algorithm. This might have been too complex for beginners and contributed to their struggles with the algorithm simulation task. On the other hand, there was no floor (or ceiling) effect on the transfer questions, suggesting that the lesson did have a level of difficulty where a modality effect could have emerged.

We cannot rule out the *segment length* of the explanations as a factor. We did keep each individual explanatory segment fairly short, but it is possible that there was nevertheless too much content at each animation step, reducing the benefits of audio. Further research would be needed to evaluate this explanation.

It is possible that *second-language issues* may have influenced our results, as the vast majority of our participants were not native English speakers. This seems an unlikely explanation, however, as the participants were competent in English and, moreover, feedback collected at the end of the experiment featured only a single complaint about the presentation style (and that from the Both group).

Student engagement may have been an issue. Although our cohort of university students was possibly better motivated on average than the crowdsourced cohort studied by Zavgorodniaia et al. [69], for example, it remains the case that in each CER study of the modality effect, including ours, the learners were recruited for an experiment, rather than the experiment being fully integrated in their usual studies. Our study's online setup muddies the waters, as the students participated independently in an uncontrolled environment and may have been tired of "yet another instructional video" during this period of emergency online teaching due to COVID-19. That being said, uncontrolled environments are common in independent study. Moreover, the low scores on the simulation task suggest that the participants largely did not access external sources.

The floor effect on the visual algorithm simulation task may be due to a combination of factors. For example: most participants will have been unfamiliar with this type of task, and the video may have insufficiently prepared them for it; the user interface may not have been intuitive enough; participants may not have noticed the instruction to break ties between nodes in alphabetical order; and many may not have used pen and paper as a memory aid even though it was suggested in the materials. (Without writing down a table such as the one in the video, it is hard for students to track the priorities during the simulation.) Furthermore, the simulation task demanded the concept of edge relaxation, but the participants may have overlooked this concept while viewing the video, which presented three cases where relaxation was not necessary, followed by a single case where relaxation occurred.

6.3 Theoretical Considerations for CER

The best-known research on the modality effect originates in CTML, extending CTML's "multimedia principle" that recommends adding pictures to words [56]. However, not all research on the effect bases itself on CTML—or fully aligns with it. Other researchers have expanded on CTML's explanations of the effect—visuo-spatial load

and temporal contiguity—and critiqued the former especially. In this section, we review some of these critiques and additional theoretical considerations, which do not appear to have drawn attention in CER; we recommend that future research on instructional materials for computing education is mindful of them.

Schnotz [56] and others have argued that what has been researched under the label of "modality effect" is not a unitary effect but an aggregate of many different phenomena. This, in part, explains the numerous boundary conditions and difficulties in replication. Schnotz [56] concludes that for progress to be made, the underlying causal mechanisms must be unraveled "rather than simply collecting more and more design principles, whose interrelations are not well understood."

Rummer et al. [55] presented a third explanation (besides visuospatial load and temporal contiguity) for the modality effect. According to their *auditory recency* explanation, the audio modality's benefit is that acoustic-sensory information is available longer than visual-sensory information. However, this benefit only applies to the most recently heard information, thus limiting it to very short explanatory segments and the final sentences of longer segments.

There are a number of working-memory models that place attentional resources at the forefront [1, 5, 6, 38]. From the attentionalresources perspective, the concurrent presentation of pictures and audio reduces the need for attention-consuming visual search (also referred to as "split attention") in visual-only materials. Building on other attention-emphasizing models, Sepp et al. [59] recently proposed a new model of working memory and suggested another interpretation of the modality effect and its explanations. In their model, there is a single attentional resource (as opposed to CTML's separate visual and auditive resources). This resource must attend to various competing foci of attention, which may correspond not only to the modalities of external stimuli but to many other factors, such as sociocultural knowledge or emotional state. Information presented in whichever modality places demands on this singular resource. The modality effect is (partially) explained by interference between foci of the same type.

CTML's dominant visuo-spatial load explanation has repeatedly come in for criticism. Numerous scholars (e.g., [15, 53, 55–57, 59]) have noted that the explanation does not cohere with its stated theoretical basis, namely, Paivio's dual-coding theory [13, 48] and Baddeley's [3] subsequent working-memory model. The heart of this criticism is that, according to Baddeley's model, *any* verbal information—whether presented in auditory, written, or haptic form—ends up in the phonological loop, which implies that written and audio explanations would *both* impose cognitive load on it.

Moreover, some researchers assert that presentation modality may, in fact, have limited bearing on how the presented information is internally processed. In recent work, Baddeley et al. [4] argue that inputs compete for the focus of attention regardless of their modality. Similarly, Depoorter and Vandierendonck [18, 66] and Logie et al. [37] state that working memory allows information to be encoded in different forms, independently of the modalities of the original stimuli. This is supported by a recent neuroimaging study [17].

Findings on the modality effect are also influenced by differences in people's reading and listening comprehension. There is some evidence that listeners, having less control than readers, tend to try to extract the gist of what is presented, whereas readers have more freedom to direct their attentional processes and tend to concentrate more on details [28, 54]. Thus, reading and listening to the same information can result in different mental representations [56]. This may mean that textual presentation is advantageous for certain types of materials and learning objectives, but not for others.

Finally, it is known that there are individual differences in people's working-memory capacity and that these differences influence the effectiveness of instructional methods. There is a limited amount of research on how individual differences interact with the modality effect [2, 26, 41, 61].

Since so many factors influence which instructional modality—if any—is superior in a given situation, we propose that CER is better served by targeting the contributing mechanisms rather than the modality effect as a single unit. When that future research compares the effectiveness of different types of materials for teaching algorithms, programming, and other computing content, it should consider alternative memory models beyond the traditional dual-processing model, explore differences in reading and listening comprehension, and attend to individual differences in learners' cognitive capacity.

6.4 Pedagogical Implications

In advice that cites research on multimedia learning and targets practitioners, the recommendation to prefer audio explanations of pictures, especially in system-paced materials, continues to have high visibility. For algorithm visualizations, following that advice means replacing textual explanations of a graph or other data structures with auditory ones. Our study, taken together with earlier studies within and outside CER, suggests that teachers should not rush headfirst to implement the advice. While it may not be harmful to do so, it may not bring the expected rewards, either, or be an efficient use of instructor time. Even if we were to suppose that our materials inadvertently violated the effect's boundary conditions, despite our careful attempt not to, that serves to highlight the challenge of exploiting the effect for students' benefit.

Careful judgment is needed to determine where audio might work better than text and why. Meaningful improvements may be likelier from pedagogical interventions other than adjusting instructional modality. Given the complications of research on the modality effect, it might be best for computing educators to think of the effect, for now, as an unsettled principle that is very sensitive to context and is difficult to apply reliably in practice. No easy, general answer is available; whether to incorporate audio remains something for teachers to decide in context.

Below, we list recommendations to assist computing educators with such decisions. These points arise from modality studies including ours and from the theoretical considerations noted above.

- The best-supported explanation for dual-modality presentation is temporal contiguity: simultaneous audio and pictures help learners avoid split attention. Analyze your teaching materials for suspected split-attention issues and consider using audio especially where those exist.
- Do not expect a generic cognitive benefit from dual modality on grounds that pictures and audio end up in different working-memory channels and thus effectively expand the

learner's active working memory. (As discussed above, that visuo-spatial load explanation is questionable, and presentation modality does not dictate how people process the presented information.)

- Note that explanatory segments may need to be very short for the benefits of audio to appear.
- In both multi-modal and mono-modal materials, the benefits of visual signalling are well established. Highlight relevant parts of the picture. If you use written text, integrate it with the pictures.
- Although there are many qualifications on the recommendations for audio that are based on the modality effect, there are of course unrelated reasons why audio can be a good choice. For example, audio enables intonational emphasis [22, 64] and may be used to support an engaging personal presentation style [8]. Audio is also preferable if learners are not sufficiently literate or have reading difficulties or visual impairments.
- Bear in mind that the present study and discussion have focused on system-paced materials. In learner-paced materials, the relative merits of text over audio are more pronounced than in system-paced ones.
- Consider whether listening or reading comprehension strategies might best match the learning objectives. As suggested in the previous subsection, listening might be better for obtaining a useful albeit vague overview whereas text might prompt more detailed study.
- Note that most modality studies have been carried out in artificial settings and have assessed short-term learning only. Their recommendations may not generalize to your actual classroom or to long-term learning.

7 CONCLUSION

In this article, we have explored whether audio explanations of algorithm visualizations would be more efficient than textual ones in system-paced instructional videos. Our findings do not suggest such a modality effect, and we thus contribute to a body of evidence that shows modality-related phenomena to be nuanced and challenging to apply in computing education and elsewhere. We contribute these findings and a review of theoretical factors that can instruct future research and advise teachers as they consider which instructional modalities to apply.

Videos, lectures, and other formats of instruction where learners cannot or do not control pacing are common and likely to remain so. Further research is needed on how to optimize such instruction.

Computing educators may wish to keep an eye on developments in CER and educational psychology in regard to modality-related phenomena. That research may shed more light on the mechanisms that constitute what is known as the modality effect. It may also eventually produce specific recommendations for different kinds of computing content, such as algorithm visualizations, and for different learner profiles.

REFERENCES

 Eryn J. Adams, Anh T. Nguyen, and Nelson Cowan. 2018. Theories of working memory: Differences in definition, degree of modularity, role of attention, and purpose. Language, Speech, and Hearing Services in Schools 49, 3 (2018), 340–355.

- [2] Katherine Ann Austin. 2009. Multimedia learning: Cognitive individual diffences and display design techniques predict transfer learning with multimedia learning modules. *Computers & Education* 53, 4 (2009), 1339–1354.
- [3] Alan Baddeley. 1992. Working memory. Science 255, 5044 (Jan. 1992), 556-559.
- [4] Alan Baddeley, Graham Hitch, and Richard Allen. 2021. A Multicomponent Model of Working Memory. In Working Memory. Oxford University Press, Oxford, United Kingdom, 10–43.
- [5] Pierre Barrouillet, Sophie Bernardin, and Valérie Camos. 2004. Time Constraints and Resource Sharing in Adults' Working Memory Spans. *Journal of Experimental Psychology: General* 133, 1 (2004), 83–100.
- [6] Pierre Barrouillet and Valérie Camos. 2021. The Time-Based Resource-Sharing Model of Working Memory. In Working Memory. Oxford University Press, Oxford, United Kingdom, 85–115.
- [7] Mireille Bétrancourt and André Bisseret. 1998. Integrating textual and pictorial information via pop-up windows: An experimental study. *Behaviour & Information Technology* 17, 5 (Jan. 1998), 263–273.
- [8] Cynthia J. Brame. 2016. Effective Educational Videos: Principles and Guidelines for Maximizing Student Learning from Video Content. CBE–Life Sciences Education 15, 4 (Dec. 2016), es6.
- [9] Dan Breakiron, Cliff Shaffer, Jackson Wonderly, sublime09, Eric Fouh, Kasper Hellström, Samnyeong Heo, Hossameldin Shahin, and Alex Hicks. 2021. Compiling Book Instances: OpenDSA Configuration. https://github.com/OpenDSA/ OpenDSA/blob/master/Doc/source/Configuration.rst
- [10] Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H. Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye movements in code reading: Relaxing the linear order. In *IEEE International Conference on Program Comprehension*. 255–265.
- [11] Juan C. Castro-Alonso and John Sweller. 2019. The Modality Effect of Cognitive Load Theory. In Advances in Intelligent Systems and Computing. Springer International Publishing, Berlin Heidelberg, Germany, 75–84.
- [12] Paul Chandler and John Sweller. 1991. Cognitive Load Theory and the Format of Instruction. Cognition and Instruction 8, 4 (Dec. 1991), 293-332.
- [13] James M. Clark and Allan Paivio. 1991. Dual coding theory and education. Educational Psychology Review 3, 3 (Sept. 1991), 149–210.
- [14] Robert Coe. 2002. It's the effect size, stupid: What effect size is and why it is important. http://www.leeds.ac.uk/educol/documents/00002182.htm In 2002 Annual Conference of the British Educational Research Association, Exeter, England.
- [15] Steven M. Crooks, Jongpil Cheon, Fethi Inan, Fatih Ari, and Raymond Flores. 2012. Modality and cueing in multimedia learning: Examining cognitive and perceptual explanations for the modality effect. *Computers in Human Behavior* 28, 3 (2012), 1063–1071.
- [16] Holger Danielsiek, Wolfgang Paul, and Jan Vahrenhold. 2012. Detecting and Understanding Students' Misconceptions Related to Algorithms and Data Structures. In Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE '12). ACM, New York, NY, USA, 21–26.
- [17] Fatma Deniz, Anwar O. Nunez-Elizalde, Alexander G. Huth, and Jack L. Gallant. 2019. The Representation of Semantic Information Across Human Cerebral Cortex During Listening Versus Reading Is Invariant to Stimulus Modality. *The Journal of Neuroscience* 39, 39 (Aug. 2019), 7722–7736.
- [18] Ann Depoorter and André Vandierendonck. 2009. Evidence for Modality-Independent Order Coding in Working Memory. *Quarterly Journal of Experimen*tal Psychology 62, 3 (March 2009), 531–549.
- [19] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. Numer. Math. 1 (1959), 269–271.
- [20] Barbara Ericson, Steven Moore, Briana B. Morrison, and Mark Guzdial. 2015. Usability and usage of interactive features in an online ebook for CS teachers. In WiPSCE '15. ACM.
- [21] Mohammed F. Farghally, Kyu Han Koh, Jeremy V. Ernst, and Clifford A. Shaffer. 2017. Towards a Concept Inventory for Algorithm Analysis Topics. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17). ACM, New York, NY, USA, 207–212.
- [22] Clive Frankish. 1995. Intonation and auditory grouping in immediate serial recall. Applied Cognitive Psychology 9, 7 (1995), S5–S22.
- [23] Judith Gal-Ezer and Ela Zur. 2004. The efficiency of algorithms-misconceptions. Computers & Education 42, 3 (2004), 215 - 226.
- [24] Paul Ginns. 2005. Meta-analysis of the modality effect. *Learning and Instruction* 15, 4 (2005), 313–331.
- [25] Tina Götschi, Ian Sanders, and Vashti Galpin. 2003. Mental Models of Recursion. In Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '03). ACM, New York, NY, USA, 346–350.
- [26] Kevin Greenberg, Robert Zheng, Michael Gardner, and Matthew Orr. 2020. Individual differences in visuospatial working memory capacity influence the modality effect. (2020). Preprint in the Journal of Computer Assisted Learning.
- [27] Michael Hansen, Robert L. Goldstone, and Andrew Lumsdaine. 2013. What makes code hard to understand? https://arxiv.org/abs/1304.5257
- [28] Angela Hildyard and David R. Olson. 1978. Memory and inference in the comprehension of oral and written discourse*. Discourse Processes 1, 2 (April 1978),

91-117.

- [29] Slava Kalyuga. 2011. Instructional benefits of spoken words: A review of cognitive load factors. *Educational Research Review* 7 (2011), 145–159.
- [30] Slava Kalyuga, Paul Chandler, and John Sweller. 1999. Managing split-attention and redundancy in multimedia instruction. *Applied Cogn. Psychology* (1999), 351–371.
- [31] Ville Karavirta, Petri Ihantola, and Teemu Koskinen. 2013. Service-Oriented Approach to Improve Interoperability of E-Learning Systems. In 2013 IEEE 13th International Conference on Advanced Learning Technologies. IEEE, 341–345.
- [32] Ville Karavirta, Ari Korhonen, and Otto Seppälä. 2013. Misconceptions in Visual Algorithm Simulation Revisited: On UI's Effect on Student Performance, Attitudes, and Misconceptions. In 2013 Learning and Teaching in Computing and Engineering. IEEE, Piscataway, New Jersey, United States, 62–69.
- [33] Ville Karavirta and Clifford A. Shaffer. 2016. Creating Engaging Online Learning Material with the JSAV JavaScript Algorithm Visualization Library. IEEE Transactions on Learning Technologies 9, 2 (4 2016), 171–183.
- [34] Ari Korhonen and Lauri Malmi. 2000. Algorithm Simulation with Automatic Assessment. SIGCSE Bull. 32, 3 (July 2000), 160–163.
- [35] Wayne Leahy and John Sweller. 2016. Cognitive load theory and the effects of transient information on the modality effect. *Instructional Science* 44, 1 (2016), 107–123.
- [36] Stefanie Lindow, Heather M. Fuchs, Anne Fürstenberg, Janet Kleber, Judith Schweppe, and Ralf Rummer. 2011. On the robustness of the modality effect: Attempting to replicate a basic finding. *Zeitschrift fur Pädagogische Psychologie* 25, 4 (2011), 231–243.
- [37] Robert H. Logie, Clément Belletier, and Jason M. Doherty. 2021. Integrating Theories of Working Memory. In Working Memory: State of the Science, Robert H. Logie, Valérie Camos, and Nelson Cowan (Eds.). Oxford University Press, Oxford, United Kingdom, 389–430.
- [38] Robert H. Lögie, Valérie Camos, and Nelson Cowan (Eds.). 2021. Working Memory: State of the Science. Oxford University Press, Oxford, United Kingdom.
- [39] Lauri Malmi, Ville Karavirta, Ari Korhonen, Jussi Nikander, Otto Seppälä, and Panu Silvasti. 2004. Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2. Informatics in Education 3 (10 2004), 267–288.
- [40] Giacomo Mariani. 2020. Design of an Application to Collect Data and Create Animations from Visual Algorithm Simulation Exercises. https://aaltodoc.aalto. fi/handle/123456789/44448 Master's Thesis.
- [41] Richard E. Mayer (Ed.). 2014. The Cambridge Handbook of Multimedia Learning (2nd ed.). Cambridge University Press.
- [42] Richard E. Mayer. 2020. Multimedia Learning (3rd ed.). Cambridge University Press.
- [43] Roxana Moreno and Richard E. Mayer. 1999. Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of Educational Psychology* 91, 2 (1999), 358–368.
- [44] Briana B. Morrison. 2017. Dual modality code explanations for novices: Unexpected results. In *The 2017 ACM Conference on International Computing Education Research (ICER '17)*. ACM, Tacoma, Washington, USA, 226–235.
- [45] Thomas L. Naps, Guido Rößling, Vicki Almstrum, Wanda Dann, Rudolf Fleischer, Chris Hundhausen, Ari Korhonen, Lauri Malmi, Myles McNally, Susan Rodger, and J. Ángel Velázquez-Iturbide. 2002. Exploring the Role of Visualization and Engagement in Computer Science Education. *SIGCSE Bull.* 35, 2 (June 2002), 131–152.
- [46] Kensuke Okada. 2017. Negative estimate of variance-accounted-for effect size: How often it is obtained, and what happens if it is treated as zero. *Behavior Research Methods* 49, 3 (2017), 979–987.
- [47] Nesrin Özdener. 2008. A comparison of the misconceptions about the timeefficiency of algorithms by various profiles of computer-programming students. *Computers & Education* 51, 3 (2008), 1094 – 1102.
- [48] Allan Paivio. 1986. Mental representations : a dual coding approach. Oxford University Press, Oxford, United Kingdom.
- [49] Wolfgang Paul and Jan Vahrenhold. 2013. Hunting High and Low: Instruments to Detect Misconceptions Related to Algorithms and Data Structures. In Proceeding of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE '13). ACM, New York, NY, USA, 29–34.
- [50] Patrick Peachock, Nicholas Iovino, and Bonita Sharif. 2017. Investigating Eye Movements in Natural Language and C++ Source Code - A Replication Experiment. In 11th International Conference on Augmented Cognition. 206–218.
- [51] Norman Peitek, Janet Siegmund, and Sven Apel. 2020. What drives the reading order of programmers? An eye tracking study. In *ICPC*.
- [52] Marian Petre and Alan F. Blackwell. 1999. Mental imagery in program design and visual programming. Int. J. of Human Computer Studies 51, 1 (1999), 7–30.
- [53] Joachim Reinwein. 2012. Does the modality effect exist? And if so, which modality effect? *Journal of Psycholinguistic Research* 41, 1 (2012), 1–32.
- [54] Donald L. Rubin, Teresa Hafer, and Kevin Arata. 2000. Reading and listening to oral-based versus literate-based discourse. *Communication Education* 49, 2 (April 2000), 121–133.
- [55] Ralf Rummer, Judith Schweppe, Anne Fürstenberg, Katharina Scheiter, and Antje Zindler. 2011. The perceptual basis of the modality effect in multimedia learning.

Journal of Experimental Psychology: Applied 17, 2 (2011), 159-173.

- [56] Wolfgang Schnotz. 2011. Colorful bouquets in multimedia research: A closer look at the modality effect. *Zeitschrift fur Pädagogische Psychologie* 25, 4 (2011), 269–276.
- [57] Anne Schüler, Katharina Scheiter, Ralf Rummer, and Peter Gerjets. 2012. Explaining the modality effect in multimedia learning: Is it due to a lack of temporal contiguity with written text and pictures? *Learning and Instruction* 22, 2 (2012), 92–102.
- [58] Anne Schüler, Katharina Scheiter, and Florian Schmidt-Weigand. 2011. Boundary conditions and constraints of the modality effect. *Zeitschrift fur Pädagogische Psychologie* 25, 4 (2011), 211–220.
- [59] Stoo Sepp, Steven J. Howard, Sharon Tindall-Ford, Shirley Agostinho, and Fred Paas. 2019. Cognitive Load Theory and Human Movement: Towards an Integrated Model of Working Memory. *Educational Psychology Review* 31, 2 (Feb. 2019), 293–317.
- [60] Otto Seppälä, Lauri Malmi, and Ari Korhonen. 2006. Observations on Student Misconceptions—A Case Study of the Build – Heap Algorithm. *Computer Science Education* 16, 3 (2006), 241 – 255.
- [61] Tina Seufert, Maren Schütze, and Roland Brünken. 2009. Memory characteristics and modality in multimedia learning: An aptitude-treatment-interaction study. *Learning and Instruction* 19, 1 (2009), 28–42.
- [62] Clifford A. Shaffer, Matthew L. Cooper, Alexander Joel D. Alon, Monika Akbar, Michael Stewart, Sean Ponce, and Stephen H. Edwards. 2010. Algorithm Visualization: The State of the Field. ACM Trans. Comput. Educ. 10, 3, Article 9 (Aug. 2010), 22 pages.

- [63] Juha Sorva, Ville Karavirta, and Lauri Malmi. 2013. A review of generic program visualization systems for introductory programming education. ACM Transactions on Computing Education 13, 4 (2013), 1–64.
- [64] Susan Elizabeth Thompson. 2003. Text-structuring metadiscourse, intonation and the signalling of organisation in academic lectures. *Journal of English for Academic Purposes* 2, 1 (Jan. 2003), 5–20.
- [65] Artturi Tilanterä. 2020. Towards Automatic Advice in Visual Algorithm Simulation. http://urn.fi/URN:NBN:fi:aalto-202005243300 Master's thesis. Aalto University, 2020.
- [66] André Vandierendonck. 2021. Multicomponent Working Memory System with Distributed Executive Control. In Working Memory. Oxford University Press, Oxford, United Kingdom, 150–174.
- [67] Jackson Wonderly and Ehsan Elgendi. 2017. Narration by jwonderly · Pull Request #207 · vkaravir/JSAV · GitHub. https://github.com/vkaravir/JSAV/pull/207
- [68] Zhen Xu, Albert D. Ritzhaupt, Karthikeyan Umapathy, Yang Ning, and Chin-Chung Tsai. 2021. Exploring college students' conceptions of learning computer science: A draw-a-picture technique study. *Computer Science Education* 31, 1 (2021), 60-82.
- [69] Albina Zavgorodniaia, Rodrigo Duran, Arto Hellas, Otto Seppälä, and Juha Sorva. 2020. Should explanations of program code use audio, text, or both? A replication study. In Proceedings of the 20th Koli Calling International Conference on Computing Education Research (Koli Calling '20). ACM, New York, NY, USA.
- [70] Rui Zhi. 2019. Design and Evaluation of Instructional Supports for Novice Programming Environments. Doctoral dissertation. North Carolina State University.