Visualizing Algorithms: Schematic Computer Animations versus Realistic Dance Choreography Illustrations

Zoltán Kátai, Erika Osztián

Sapientia Hungarian University of Transylvania, Mathematics and Informatics, Corunca 1C, 540485 Targu Mures, Romania
katai_zoltan@ms.sapientia.ro, osztian@ms.sapientia.ro

Abstract: In this paper, we analyze the relative effectiveness of schematic computer animations and Human Movement Effect (HME) enhanced realistic videos in the context of algorithm visualization. Our investigation was implemented in the AlgoRythmics learning environment, which includes realistic dance choreographies and abstract animations of ten basic computer algorithms. Previous research regarding the AlgoRythmics videos focused on their motivational value. The present study investigates these visualizations as tools for helping students understanding the strategies the illustrated algorithms apply. Our most important conclusion is that HME enhanced realistic algorithm visualizations (such as the AlgoRythmics videos) could be as effective as, or even more effective, than abstract animations, in getting students to understand computer algorithms. The results also confirmed that one of the strengths of the AlgoRythmics videos (as realistic visualizations), is that they are easier to remember.

Keywords: Algorithm visualization; Schematic visualization; Realistic visualization; Human movement effect

1 Introduction

Algorithms are central to Computer Science (CS) education and they are closely related to computational thinking (CT) too [21]. Turing [42] highlights that a deep understanding of how computer algorithms work requires that students are able to construct “a clear mental picture of the state of the machine at each moment in the computation”. Visual aids that illustrate the procedural behavior of an algorithm are meant to bolster this internal building process [29]. Due to the dynamic nature of algorithms, dynamic visualizations such as animations and videos are becoming more popular in learning environments designed for teaching computer algorithms [20] [38]. In this paper, we examine the relative effectiveness of animated visual aids and realistic video footage (by contrasting them) in the context of algorithm visualization.

2 Promoting Computational Thinking by Algorithm Visualization

A major task of the educational systems of the XXI century is to provide future generations the tools to handle the challenges associated with the increasingly computerized nature of our day-to-day lives and to prepare them for the demands of one of the fastest-growing job markets: computing [27, 44]. In accordance with this, the Future Work Skills Report of the Institute for The Future considers computational thinking to be among the top 10 skills that are likely to be essential for success in 2020 [19]. For example, Echeverría et al. points out that computing is a necessary skill for any engineering field [23].

The term computational thinking has been the subject of much debate since its introduction in 2006 [47]. Wing suggested that CT is "a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" [47]. Aho redefines the concept as the thought process involved in formulating problems so that "their solutions can be represented as computational steps and algorithms" [2]. He emphasizes that "the computational models are the heart of computation and computational thinking". Denning appraises the revised definition of Aho, and outlines the importance of the computational models and algorithms, describing the steps of algorithms as the design of the "way to control any machine that implements the model" [22].

Other researchers addressed this issue by studying the differences and similarities between CT and other types of thinking [9] [27]. For example, Shute et al. [39], based on [40], identified a set of shared concepts of computational and mathematical thinking: problem-solving, modeling, data analysis and interpretation, and statistics and probability [39]. In addition, Baranyi and Gilanyi [8] defined the concept of mathability as a branch of cognitive infocommunications that investigates any combination of artificial and natural cognitive capabilities relevant to mathematics [16] [18]. In the context of computer-assisted learning Gilanyi et al. [24] [25] associate the concept of mathability with the dynamic visualization of approaching some mathematical problems [17].

Since computer algorithms are abstract processes just like many mathematical concepts, in this paper we focus on promoting learners' computational thinking by inviting them in an exciting algorithm visualization environment.

3 Schematic versus Realistic Algorithm Representation

Educational dynamic visual aids can be divided into two categories: schematic and realistic representations [35]. During the previous decades, a number of studies, all from different fields of science, have examined the relative value of these types of visual aids [22, 35, 37]. On the other hand, the topic of algorithm visualization still lacks proper research from this perspective. A possible explanation for this could be that computer algorithms are inherently abstract in nature; therefore, they lack tangible real-world representations. Therefore, animations are most often used to illustrate computer algorithms. However, such representations exist, especially in the field of unplugged CS education, which are closer to realistic visualizations. A relevant example for this would be the AlgoRythmics [5] learning platform which includes ten videos (besides schematic computer animations) illustrating basic computer algorithms with the aid of dance choreography footage.

Prior research which compares the efficacy of these types of visualizations concludes that both types may have their own benefits depending on the learning goals [35]. For example, the coherence principle of multimedia learning suggests that a schematic animation may help learners focus on the relevant aspects of the visualization, since it contains fewer irrelevant/distractive elements [11] [34]. On the other hand, according to Goldstone and Son [26], the two main advantages of realistic visual aids that are:

(1) They are easier to remember (due to their more concrete nature)

(2) They trigger intrinsic motivation (students find them more appealing). These advantages might be present in the AlgoRythmics visual aids as well.

4 Human Movement Effect

It is rather peculiar that the studies that examined whether dynamic visual aids contribute to the learners' comprehension of dynamic phenomena have shown mixed results [3]. The most common reason to why animations are found to be inconsistent in their effectiveness is that they are transient by nature [7] [41]. An interesting fact is that dynamic visual aids have been unanimously found to be superior to static visual aids when the animations involved human movement [14]. In this particular case, this would suggest that the mirror neuron system aids the working memory in processing transitory information.

Therefore, a considerable strength of the AlgoRythmics videos is their ability to illustrate the basic modus operandi of the algorithms with the use of human movements. Recent studies emphasize that watching human movements might be cognitively beneficial [15]. According to Van Gog et al. [45], this "Human Movement Effect" (HME) [36] enhances dynamic visualizations. It does this by counteracting the negative effect of their transient nature.

The majority of the studies conducted in the field of realistic versus schematic visual aids report strong evidence that supports schematic visual aids. For example, Scheiter et al. [37] conducted an analysis about teaching settings that combined realistic and schematic dynamic visual aids in biology education. Results indicated that participants from the group that used only realistic visual aids (the same visualization presented in succession) performed significantly worse than their peers from the other three categories (schematic with schematic, schematic with realistic, realistic with schematic). Recently, the authors of study [35] reported similar results. They examined the following settings: schematic exclusive, realistic exclusive, sequential schematic and realistic, simultaneous schematic and realistic. Once more, the realistic exclusive group scored significantly lower than the other three. These results concord with the statement Tversky et al. [43] made that animated visual aids should aim for minimal realism because even appealing realistic details may hamper comprehension of the relevant movements of dynamic visual aids.

The question is what if the realistic animated visual aid includes visualization of human movements (HME-realistic). We aim to find out whether they benefit from the HME to the point that they surpass (or at least equally as effective as) their animated counterparts. This study uses the AlgoRythmics learning environment to answer these questions.

5 Computer Animation versus Dance Choreography illustration

The first videos illustrating six basic sorting algorithms by different folk-dance choreographies from Transylvania (part of Romania) were uploaded to the AlgoRythmics YouTube channel in 2011 [4]. This initial collection was extended with four new choreographies (an additional sorting strategy and three searching strategies) in 2018. The AlgoRythmics videos in total have amassed more than 6 million views as of January 2020. The authors of the videos highlight that while the videos have their evident advantages, they do not take the place of abstract animations, they only supplement them [30]. Therefore, the AlgoRythmics environment [5] was created where computer animations are associated to the majority of dance choreography illustrations [31].

AlgoRythmics animations are considered dynamic schematic visual aids and are characterized by their stylistic consistency. The arrays used for all searching and sorting algorithms are illustrated by rectangles (Figure 1). In each visualization the comparing and swapping operations are represented similarly. Another similarity is that once an element founds its position in the sorted sequence, it changes its color. In contrast to this, because of the variety of the dance styles, artistic performances have unique traits. A shared characteristic of each algorithm is that elements to be sorted are embodied by distinctively dressed dancing humans. In addition, in the case of two searching strategies the arrays that store the lists to be searched are represented by chairs (with dancers sitting on them). Other videos, illustrate these arrays by graphical entities added via video editing software (Figures 2 and 3).

The lack of a coherent style is also a product of the *compare* and *swap* operations being illustrated by a dance style unique to them.

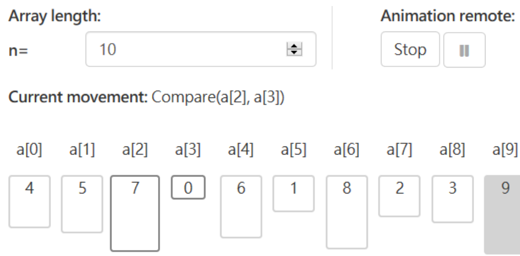

Figure 1

Bubble sort animation in the AlgoRythmics environment (comparing a[2] and a[3]; the largest number has already been moved to a[9])

Figure 2

Bubble sort with Hungarian (Csángó) folk dance (comparing and swapping a[4]=5 and a[5]=4; the three largest numbers have already been moved to their final positions; although the other elements are also in place, the algorithm only detects this in the next pass)

Figure 3

Selection sort with Gipsy folk dance (swapping a[4]=5 and a[7]=4; the four smallest numbers have already been moved to their final positions)

Höffler and Leutner [28] make a distinction between representational animations (the idea in question is explicitly depicted in the animation) and decorative animations (the main purpose of the animation is to motivate the learner) [13] in their meta-analysis in the field of "instructional animation versus static pictures". They found that animations are specifically superior to still images when the visual aid has a representational role (the motion depicted using animation explicitly refers to the topic to be learned).

Using this terminology as a basis, we identified two kinds of realistic/artistic elements in the AlgoRythmics videos: (1) decorative elements (certain costumes, music, male/female dancers, etc.) and (2) representative elements (certain dance moves illustrating the essential operations within the algorithms). Likewise, we separated the potentially distracting elements into two categories as well: (1) decorative distracting elements and (2) representative distractive elements. All decorative elements could potentially be decorative distracting elements. Representative distractive elements are realistic/artistic elements which obscure the link (the one-on-one relationship) between the crucial operations of the algorithm and their corresponding dance moves. We were especially interested in how these distracting elements might influence the effectiveness of the AlgoRythmics videos.

6 Hypotheses

Previous studies examined the algorithmic dances mainly from a motivational perspective. These papers focused mostly on the role of decorative elements: to arouse curiosity by providing novelty, incongruity and surprise [30]. In this study we concentrate on comparing the computer representations and dance choreography illustrations as equivalent tools for supporting learners in comprehending the algorithmic strategies.
Two algorithms were included in the experiment: Bubble and Selection sort. Both sorting strategies have $O(n^2)$ worst case time complexity. For the best case the time complexities are linear and quadratic, respectively. The corresponding videos visualize the algorithmic strategies by Hungarian (Csángó) and Gipsy folk dance performances (Figures 2 and 3). Naturally, each artistic representation contains a significant number of decorative elements.

Both strategies can be seen as a succession of compare and compare+swap operations. During the Selection sort dance the comparisons are represented constantly by the same dance steps (regardless whether, or not, they are linked to swap operations). As for the Bubble sort video, in the combined compare+swap scenes the two operations overlap to a certain extent. Consequently, the Bubble sort choreography was categorized as containing both decorative and representational distractors.

Two instructional conditions were examined: (1) Bubble-sort→Selection-sort – both algorithms visualized by schematic animations –; (2) Bubble-sort→Selection-sort – both algorithms illustrated by HME enhanced realistic videos –. Based on the above detailed literature review, we formulated the following hypotheses:

- **Hypothesis-1:** Because of the presence of HME, participants assigned to the “realistic group” will perform at least as well as their colleagues from the “schematic group”
- **Hypothesis-2:** Since realistic visualizations can be remembered more easily, the members of the “realistic group” will be more easily able to distinguish between the two algorithms
- **Hypothesis-3:** Representational distractors may obstruct the comprehension of the algorithmic content of the dance choreography visualizations

7 Method

A “pretest–study–posttest” (control vs. experimental group) experimental design was applied. The experimental group (group-V) was presented with the (v)ideos of the dance choreography illustrations (“realistic group”) and the control group (group-A) with the computer (a)nimations (“schematic group”).

7.1 Participants

The experiment was implemented at a public university (Eastern Europe) at the beginning of the 2019-20 academic year. The investigation was approved by the university human research ethics committee and informed consent was obtained from all individual students included in the study. All freshmen enrolled in the Information Science, Computer Science, Automation and Applied Informatics, Mechatronics and Computer-aided Operation Planning programs (160 students; aged from 19 to 22 years) were invited to participate in the study. Subjects were randomly assigned to the two groups. More precisely, participants from each above-listed educational programs were randomly assigned to one of the groups. 140 students took part in the pre-test and 120 of them in the post-test too. Since 4 students abandoned the post-test (after the first few questions), the resulting group of participants had 116 members: both group-V (14% females) and group-A (16% females) included 58 members.

At the beginning of the pre-test we asked participants to indicate their gender, the number of years they had studied programming during their high school studies (0 years, 1–2 years, 4 years,) and if they had studied the Bubble and Selection sort algorithms previously. Table 1 shows these data for the two groups. According to Fisher’s exact test, no significant differences were detected between the groups regarding these aspects.

Table 1
Comparing participants from the groups

	Prior programming years		
	0 years	1–2 years	4 years
Group-V	29%	22%	48%
Group-A	40%	29%	31%
Fisher exact test	(2×3; Freeman-Halton extension) $p = 0.18 > 0.05$		

	Bubble sort		Selection sort	
	yes	no	yes	no
Group-V	55%	45%	29%	71%
Group-A	47%	53%	19%	81%
Fisher exact test	(2×2) $p = 0.45 > 0.05$		(2×2) $p = 0.27 > 0.05$	

7.2 Procedure and Materials

The pre-test took place during the first week of the academic year. Since the group of participants was very heterogeneous regarding their prior knowledge in programming, we had proposed to test students' CT skills. The pre-test assignments were based on two, apparently, CS-free, tasks (2015-AT-03, 2015-CZ-09) selected from the website of the Bebras contest [10]. After the 30 minute pre-test, students (to become familiar with the environment) were given a brief "AlgoRythmics tasting" (based on the video and animation of the insertion sort algorithm). The way in which the compare and swap operations are visualized in this environment was emphasized.

The study and post-test phases took place in the 2nd week of the semester. Subjects were invited to follow a three step learning-testing session:

1. **Learning phase:**
   - group-V watched the bubble sort dance-performance
   - group-A watched the bubble sort computer-animation
2. **Learning phase:**
   - group-V watched the selection sort dance-performance
   - group-A watched the selection sort computer-animation
3. **Testing phase:**

   All participants were asked to answer 13 questions (using the Socrative classroom app) with respect to both algorithmic strategies

During the first two steps, the corresponding visualizations were presented (in succession) in front of the whole groups (in two different amphitheaters) from the computer of the teacher (using a video projector). All four visualizations illustrate the corresponding sorting strategy on a 10-length random sequence. The speeds of the animations were set so that the matching visualizations were approximately the same length.

The two Bubble sort visualizations were "algorithmically isomorph". Both of them presented an "optimized version" of the algorithm (for example, the current pass ends at the position of the last swap of the previous pass) which could help students realize that the best case time complexity of the algorithm is O(n). In the case of Selection sort, the animation visualized the common implementation of the algorithm when the number of swaps in worst-case is O(n). In the danced variant of the algorithm, the worst-case number of swaps was $O(n^2)$ (each comparison is followed by a swap).

During the testing phase, (post-test) participants from both groups had to answer the questions below using their mobile devices. In order to test how deeply the students understood the algorithm, they were required to apply the strategies for a new random sequence and imagine the best and worst case (ascending/descending arrays) behavior of them. The questions were as follows:

- Questions 1–4: Which is the first/second/third/fourth operation performed by the Bubble sort algorithm during its first traverse through the sequence a[0..6]={12, 7, 15, 5, 6, 4, 8}? Compare/Swap(a[..],a[..])
- Questions 5–8: Which is the first/second/third/fourth operation performed by the Selection sort algorithm during its first traverse through the sequence a[0..6]={12, 7, 15, 5, 6, 4, 8}? Compare/Swap(a[..],a[..])
- Question 9: Which is the first operation performed by the Bubble sort algorithm during its second traverse through the sequence a[0..6]={12, 7, 15, 5, 6, 4, 8}? Compare/Swap(a[..],a[..])
- Question 10: Which is the first operation performed by the Selection sort algorithm during its second traverse through the sequence a[0..6]={12, 7, 15, 5, 6, 4, 8}? Compare/Swap(a[..],a[..])
- Questions 11–14: How many compare/swap operations does the Bubble sort algorithm perform for an ascending/descending 5-length array?
- Questions 15–18: How many compare/swap operations does the Selection sort algorithm perform for an ascending/descending 5-length array?
- Questions 19–22: How many compare/swap operations does the Bubble sort algorithm perform for an ascending/descending N-length array?
- Questions 23–26: How many compare/swap operations does the Selection sort algorithm perform for an ascending/descending N-length array?

8 Statistical Analysis

For statistical computing we used the R software environment. To check the validity of the equal variances assumption we used Levene's tests (pre-test: $p=0.8>0.05$; post-test, bubble sort: $p=0.07>0.05$; post-test, selection sort: $p=0.72>0.05$). The two groups' prior knowledge level was compared by ANOVA (one-way Analysis of Variance) We chose as independent variable the instructional condition (animation vs. video). The dependent variable was the pre-test score. Results showed (see Figure 4) that the two groups did not differ significantly on the pre-test scores (group-V: 68%, group-A: 69%; $F(1,114)=0.1$, $p=0.74>0.05$).
We coded students' post-test answers as follows:

- The potential range of scores for each algorithm was: 0–13
- Received 2 points (instead of 4) for the first four questions (Questions 1–4 and 5–8) those participants who indicated the element-pairs to be processed consequently correctly, but the associated operations (compare or swap) were not the right ones in every case
- In the case of Questions 9, 12, 20, and 10, 16, 24 participants received 1/0 point for correct/incorrect answers
- In the case of Questions 11, 13, 14, and 15, 17, 18 received 0.5 points (instead of 1) those participants who answered 5 instead of 4 (n instead of $n-1$ when n was 5) or 15 instead of 10 ($n(n+1)/2$ instead of $n(n-1)/2$ when n was 5)
- In the case of Questions 19, 21, 22 and 23, 25, 26 received 0.75 points (instead of 1) those participants who answered n instead of $n-1$ or $n(n+1)/2$ instead of $n(n-1)/2$; those who indicated a wrong formula but the degree of the polynomial was correct (linear or quadratic) received 0.25 points for the corresponding answer.

9 Results and Discussion

9.1 Abstract Animation versus Dance Choreography

For both algorithms testing phases results were analyzed with a one-way Analysis of Covariance (ANCOVA) (Figure 4). We chose as independent variable the instructional condition (animation vs. video), as dependent variable the post-test score, and as covariate the pre-test score. On the one hand, we found no significant differences between the two groups with respect to Bubble sort results (group-V: 65%, group-A: 68%; $F(1,113)=0.78$, $p=0.37>0.05$). On the other hand, group-V outperformed group-A on the Selective sort algorithm (group-V: 58%, group-A: 45%; $F(1,113)=9.52$, $p=0.002<0.05$; partial $\eta^2 = 0.07$). As for the prior knowledge (pre-test score), this component was not related to post-test score (Bubble sort: $F(1,112)=0.00$, $p=0.99>0.05$; Selection sort: $F(1,112)=0.23$, $p=0.62>0.05$).

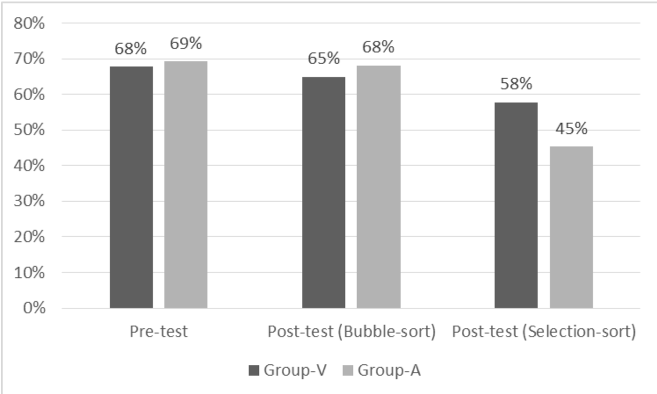

Figure 4
The pre-and post-test results of the two groups

These results confirm Hypothesis-1 that the "realistic group" will perform at least as well as their colleagues from the "schematic group". Although this finding is contrary to some previous research, such as [35, 37], it is in accordance with recent results in HME (as detailed above). For example, the authors of study [37] to explain why their realistic-twice condition scored lowest, recall that learning from dynamic visualizations (contrary to learning from static pictures) could be a highly demanding process due to the rapid changes in the display [33]. They suggest that, consequently, realistic dynamic visualizations (involving extra cognitive demand) may prove unsuitable for learning under any circumstance.

However, in the special case of learning with visualizations that link the content to learners' own body or action repertoire, it seems that the mirror neuron system aids working memory to cope with transitory information [14]. Therefore, the generated human movement effect [36] could be a possible explanation why our group-V participants performed at least as well as their colleagues from group-A.

The reason why group-V performed significantly better on the second algorithm may be more complex. The higher scores with respect to the first algorithm (group-V: 65% vs. 58%, $p<0.0001$; group-A: 68% vs. 45%, $p<0.0001$; see Figure 4) suggest that both groups were more focused during the first visualization. In this case, no significant differences were detected between the two conditions. Since the second visualization was played immediately after the first one, it is possible that all participants were a bit confused during this phase of the learning process. However, it seems that group-V was able to handle this situation more effectively.
As proposed by Goldstone and Son [22], a possible strength of realistic visual aids is that they are easier to remember. Regarding AlgoRythmics videos, we can state

without any exaggeration that, due to their unique elements, each artistic performance has its own personality. In order to answer step-3 items (which alternate between the two strategies) participants had to remember and clearly separate the two sorting methods in their minds. Group-V seemed to accomplish this easier, which could explain why this group's performance was superior to the other one's on the Selection sort strategy. Since the computer representations contained the same "decorative elements", group-A students were more likely to confuse the algorithmic content of the two strategies. According to hypothesis-2, this result highlights that one of the advantages of the AlgoRythmics representations could be that they are easier to remember.

9.2 Decorative versus Representational Distractors

As the next step, we considered participants' answers with respect to each item of the questionnaire (or group of items). In the case of Bubble, sort algorithm (see Table 2.a) significant differences were detected between the results of the two groups, considering Questions 1–4 (first four operations; p=0.01), 9 (first operation of the second pass; p=0.03), 11 (number of comparisons in the best case; 5-length sequence; p=0.01) and 19 (number of comparisons in the best case; n-length sequence; p=0.01). In all these cases group-A performed better than group-V, except Question 9.

Table 2a

Post-test result (%) regarding the Bubble sort algorithm

	Q1–4	Q9	Q11	Q12	Q13	Q14	Q19	Q20	Q21	Q22
Group-V	82	78	66	83	48	71	58	72	21	21
Group-A	92	62	82	86	40	67	73	76	15	17

In the case of the Selection sort algorithm (see Table 2.b), significant differences were detected between the results of the two groups, considering Questions 5–8 (first four operations; p=0.001), 17 (number of comparisons in the worst case; 5-length sequence; p=0.0001) and 25 (number of comparisons in the worst case; n-length sequence; p=0.04). In each of these cases, group-V performed better than group-A.

Table 2b

Post-test result (%) regarding the Selection sort algorithm

	Q5–8	Q10	Q15	Q16	Q17	Q18	Q23	Q24	Q25	Q26
Group-V	78	62	51	91	59	53	20	59	22	22
Group-A	56	66	45	86	29	16	13	66	12	34

With respect to Questions 1–4 (first four operations of the Bubble sort algorithm) we observed that 16 members from group-V associated incorrect operations (compare/swap) to correctly selected element-pairs to be processed. This is strange because once the element-pair to be processed has been selected, the associated operation is implicit ("compare"/"compare+swap" if the elements are in right/wrong order). Only three errors of this type were detected in the case of group-A. A possible reason could be, as detailed above, that in the bubble sort video the "compare+swap" dance-step does not include a clearly separable "comparing-phase" and "swapping-phase". On the other hand, the animated comparisons are the same regardless of whether the operation to be performed is "compare" or "compare+swap". We did not observe a similar phenomenon with regard to the Selection sort strategy. Again, a possible reason could be that in this case the comparing and swapping operations are illustrated (consistently) by clearly distinguishable dance steps. Interestingly, group-V performed significantly better on Questions 5–8 (first four operations of the Selection sort algorithm).

In the best case, sorting algorithms only perform comparing operations. The reason why group-V had a significantly lower score on both questions (11, 19) regarding the best-case behavior of the Bubble sort algorithm could be the same: because of the presence of representational distractor subjects might find it difficult to imagine a swapping free scenario.

Considering our third hypothesis it can be said that while it is possible that HME has the potential to compensate for the distractive effects generated by the more artistic decorative elements, representational distractors may also adversely affect the efficacy of realistic visualizations. These distractive elements may directly affect the movements that serve as illustrations for the crucial operations of the algorithm.

9.3 Results Grouped by Gender

To address the male versus female topic, two two-way ANOVAs were conducted. We chose as independent variables the instructional condition (video or animation) and the gender (male or female). The dependent variable was participants' posttest results. No interaction was detected for either algorithmic strategy (Bubble sort: $p=0.91>0.05$; Selection sort: $p=0.65>0.05$). A number of previous research also found no significant differences between the programming course outcome of male and female students [1, 12, 46]. For example, Lau and Yuen [32] found no significant correlation between gender and programming performance of 217 secondary school learners (aged from 14 to 19). More recently, the authors of the [6] study also concluded that both male and female students attained the same level of CT skills development.

9.4 Theoretical and Practical Implications

This study reveals new insight into the research field of dynamic visualization (with a particular focus on the topic of algorithm visualization) and provides useful suggestions for instructional design. We confirmed and extended some previous findings in the topic of schematic and realistic dynamic visualization to the field of algorithm visualization. Our most important conclusion is that HME enhanced realistic algorithm visualizations (such as the AlgoRythmics videos) could be as effective as, or even more effective than abstract animations in getting students to understand computer algorithms. Since algorithmic contents can quite naturally be associated with human movements (even definitions of the algorithm include expressions like "step-sequence", "proceeding through", etc.), instructional designers are encouraged to explore further ways in which the HME can be adopted in algorithm visualizations.
Realistic algorithm visualizations (just like computer animations) are designed and created with educational purposes while other realistic representations are visualizing given real systems or processes. Our results regarding the negative effect of the representational distractors underline the importance of preventing from the design phase of such representations the phenomenon when appealing realistic/artistic elements impede the comprehension of the visualized algorithmic operation-sequence.

The results also confirmed that one of the strengths of the AlgoRythmics videos (as realistic visualizations) is that they are easier to remember. As Computer Science educators, we found the following in-class scheduling to be the most effective: (1) the teacher explains the algorithm to be assimilated by using an expressive abstract animation; (2) at the end of the class students are watching the dance choreography illustration of the studied algorithm. Since at this stage of the learning process students are already familiar with the logic of the algorithm, they can be more focused on the relevant movements of the visualization and, consequently, ignore potentially distracting elements easier.

Interestingly, we found many YouTube comments that also underline the importance of the above-discussed aspects: "I have data structures exam tomorrow morning this is what I needed"; "Thank you for helping me with passing the exam"; "My Computer Science professor told us to watch this video to understand merge sort and I'm happy now"; "Wow, this helped me understand how quicksort works way better than the tutorial video I just watched"; "Now I never forgot bubble sorting", etc.

9.5 Limitations

A limitation of this research is that the AlgoRythmics videos are very special variants of realistic dynamic visualizations and this makes it difficult to generalize the results.

Another limitation could be that: we examined only two sorting algorithms; the visualizations of the second algorithm were not "algorithmically identical" which could perturb participant scores especially with respect to the worst-case questions; the realistic visualizations were not completely free of schematic elements (the dancers appear in indexed rectangles representing the array that stores the numbers).

The fact that in this research we ignored the motivational aspects of the generated learning experience could also be seen as a limitation of our investigation since the motivational elements could influence the results.

Conclusions

In this investigation we report on the relative efficacy of abstract computer animations (schematic representations) and dance choreography illustrations (human movement affect enhanced realistic videos), in the context of algorithm visualization. The investigation we performed was implemented within the AlgoRythmics learning environment, which includes realistic dance choreographies and abstract animations of ten basic computer algorithms. Previous research regarding the AlgoRythmics videos focused on their motivational value. The present study investigated these visualizations as tools for supporting students understanding the algorithmic content of the studied sorting strategies.

References

[1] A. Ahadi, R. Lister, S. Lal, J. Leinonen, A. Hellas: Performance and consistency in learning to program, Proceedings of the Nineteenth Australasian Computing Education Conference, 2017, pp. 11-16

[2] A. V. Aho: Computation and computational thinking, The computer journal, 55(7), 2012, pp. 832-835

[3] S. Ainsworth, N. VanLabeke: Multiple forms of dynamic representation, Learning and instruction, 14(3), 2004, pp. 241-255

[4] AlgoRythmics channel: https://www.youtube.com/user/AlgoRythmics, 2011

[5] AlgoRythmics environment: https://algorythmics.ms.sapientia.ro, 2020

[6] S. Atmatzidou, S. Demetriadis: Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences, Robotics and Autonomous Systems, 75, 2016, pp. 661-670

[7] P. Ayres, F. Paas: Making instructional animations more effective: A cognitive load approach. Applied Cognitive Psychology, The Official Journal of the Society for Applied Research in Memory and Cognition, 21(6), 2007, pp. 695-700

[8] P. Baranyi, A. Gilányi: Mathability: Emulating and enhancing human mathematical capabilities, in 2013 IEEE 4th international conference on cognitive Infocommunications (CogInfoCom), 2013, pp. 555-558

[9] D. Barr, J. Harrison, L. Conery: Computational thinking: A digital age skill for everyone, Learning & Leading with Technology, 38(6), 2011, pp. 20-23

[10] Bebras. International Challenge on Informatics and Computational Thinking, https://www.bebras.org, 2020

[11] K. R. Butcher: Learning from text with diagrams: Promoting mental model development and inference generation, Journal of educational psychology, 98(1), 2006, p. 182

[12] P. Byrne, G. Lyons: The effect of student attributes on success in programming, Proceedings of the 6th annual conference on Innovation and technology in computer science education, 2001, pp. 49-52

[13] R. N. Carney, J. R. Levin: Pictorial illustrations still improve students' learning from text, Educational psychology review, 14(1), 2002, pp. 5-26

[14] J. C. Castro-Alonso, P. Ayres, F. Paas: Dynamic visualisations and motor skills, in Handbook of human centric visualization, Springer, New York, NY, 2014, pp. 551-580

[15] J. C. Castro-Alonso, C. Juan: Learning symbols from permanent and transient visual presentations: Don't overplay the hand, Computers & Education, 116, 2018, pp. 1-13

[16] K. Chmielewska, A. Gilányi: Mathability and computer aided mathematical education, in 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2015, pp. 473-477

[17] K. Chmielewska, A. Gilányi: Computer assisted activating methods in education, in 2019 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2019, pp. 241-246

[18] K. Chmielewska, A. Gilányi, A. Łukasiewicz: Mathability and mathematical cognition, in 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 245-250

[19] A. Davies, D. Fidler, M. Gorbis: Future work skills 2020, Institute for the Future for University of Phoenix Research Institute, 2011

[20] B. B. De Koning, H. K. Tabbers: Facilitating understanding of movements in dynamic visualizations: An embodied perspective, Educational Psychology Review, 23(4), 2011, pp. 501-521

[21] P. J. Denning: The profession of IT Beyond computational thinking, Communications of the ACM, 52(6), 2009, pp. 28-30

[22] P. J. Denning: Remaining trouble spots with computational thinking, Communications of the ACM, 60(6), 2017, pp. 33-39

[23] L. Echeverría, R. Cobos, L. Machuca, I. Claros: Using collaborative learning scenarios to teach programming to non- CS majors, Computer applications in engineering education, 25(5), 2017, pp. 719-731

[24] A. Gilányi, N. Merentes: Mathability and an animation related to a convex-like property, in 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 227-232

[25] A. Gilányi, N. Merentes, R. Quintero: Presentation of an animation of the m-convex hull of sets, in 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 307-308

[26] R. L. Goldstone, J. Y. Son: The transfer of scientific principles using concrete and idealized simulations, The Journal of the learning sciences, 14(1), 2005, pp. 69-110

[27] S. Grover, R. Pea: Computational thinking in K–12: A review of the state of the field, Educational researcher, 42(1), 2013, pp. 38-43

[28] T. N. Höffler, D. Leutner: Instructional animation versus static pictures: A meta-analysis, Learning and instruction, 17(6), 2007, pp. 722-738

[29] Z. Katai: Selective hiding for improved algorithmic visualization, Proceedings of the 2014 conference on Innovation & technology in computer science education, 2014, pp. 33-38

[30] Z. Kátai: The challenge of promoting algorithmic thinking of both sciences- and humanities- oriented learners, Journal of Computer Assisted Learning, 31(4), 2015, pp. 287-299

[31] Z. Katai, L. Toth, A. K. Adorjani: Multi-Sensory Informatics Education, Informatics in Education, 13(2), 2014, pp. 225-240

[32] W. W. Lau, A. H. Yuen: Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy, British Journal of Educational Technology, 40(4), 2009, pp. 696-712

[33] R. K Lowe: Extracting information from an animation during complex visual learning, European journal of psychology of education, 14(2), 1999, pp. 225-244

[34] R. E. Mayer, R. Moreno: Nine ways to reduce cognitive load in multimedia learning, Educational psychologist, 38(1), 2003, pp. 43-52

[35] M. L. Nugteren, H. K. Tabbers, K. Scheiter, F. Paas: Simultaneous and sequential presentation of realistic and schematic instructional dynamic visualizations, in Handbook of human centric visualization, Springer, New York, NY, 2014, pp. 605-622

[36] F. Paas, J. Sweller: An evolutionary upgrade of cognitive load theory: Using the human motor system and collaboration to support the learning of complex cognitive tasks, Educational Psychology Review, 24(1), 2012, pp. 27-45

[37] K. Scheiter, P. Gerjets, T. Huk, B. Imhof, Y. Kammerer: The effects of realism in learning with dynamic visualizations, Learning and Instruction, 19(6), 2009, pp. 481-494

[38] C. A. Shaffer, M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, S. H Edwards: Algorithm visualization: The state of the field, ACM Transactions on Computing Education (TOCE), 10(3), 2010, pp. 1-22

[39] V. J. Shute, C. Sun, J. Asbell-Clarke: Demystifying computational thinking, Educational Research Review, 22, 2017, pp. 142-158

[40] C. Sneider, C. Stephenson, B. Schafer, L. Flick: Computational thinking in high school science classrooms, The Science Teacher, 81(5), 2014, p. 53

[41] K. Stenning: Distinguishing semantic from processing explanations of usability of representations: applying expressiveness analysis to animation, Intelligence and multimodality in multimedia interfaces: research and applications, 1998

[42] A. M. Turing: Mind, Mind, 59(236), 1950, pp. 433-460

[43] B. Tversky, J. B. Morrison, M. Betrancourt: Animation: can it facilitate?, International journal of human-computer studies, 57(4), 2002, pp. 247-262

[44] Us-Bls (Us Bureau of Labor Statistics), http://www.bls.gov/ooh/

[45] T. Van Gog, F. Paas, N. Marcus, P. Ayres, J. Sweller: The mirror neuron system and observational learning: Implications for the effectiveness of dynamic visualizations, Educational Psychology Review, 21(1), 2009, pp. 21-30

[46] L. H. Werth: Predicting student performance in a beginning computer science class, ACM SIGCSE Bulletin, 18(1), 1986, pp. 138-143

[47] J. M. Wing: Computational thinking, Communications of the ACM, 49(3), 2006, pp. 33-35