

Critical Analysis on Algorithm Visualization Study

Ahmad Affandi Supli
College of Arts Sciences
School of Multimedia
Technology and
Communication
Universiti Utara Malaysia

Norshuhada Shiratuddin
College of Arts Sciences
School of Multimedia
Technology and
Communication
Universiti Utara Malaysia

Syamsul Bahrin Zaibon
College of Arts Sciences
School of Multimedia
Technology and
Communication Universiti
Utara Malaysia

ABSTRACT

This paper reports on an ongoing study, which intends to propose a principle of interactive algorithm visualization on hybrid mobile application (INAVOHMA) that is created in order to help IT students learn data structure and algorithm (DSA) subject. Totally, 8 existing AV guidelines and models were reviewed comprehensively with the main purposes (1) to determine the research gaps in proposing principles of INAVOHMA and (ii) to identify their common components. Through a systematic and critical analysis, this study discovers there is still lack of inclusive principles or guidelines of AV that focused on mobile platform, mostly for desktop or website platform. Only, two guidelines draw attention to mobile platform, yet the focus of them just for sorting algorithm only. It is noted that this is the research gap that should be the focal point for further study.

General Terms

Algorithm, Software engineering, Data structure and algorithm

Keywords

Algorithm Visualization, Critical Analysis, Finding Gaps, Paper Submission.

1. INTRODUCTION

Understanding Data Structures and Algorithm (DSA) course which contains the arrangement of algorithm theory is a truly difficult errand in faculty of computer science [1-2]. DSA is one of the most imperative subjects, but the abstract nature of it makes this subject becoming the hardest one to learn [3]. Generally, the difficulty relies on algorithm itself which derived from dynamic step by step processes.

Reference [4] mentioned four difficulties that students and lecturers have to face: first problem is students have low motivation. The second is the abstract of DSA concepts makes it difficult and tricky to explain. Third, the cooperation between students in doing assignment is still low. Fourth, it is hard to implement the algorithm into real-life setting.

Primarily, this course is taught through conventional approach (PowerPoint and books). Particularly, the set of images are used to elaborate the processes occurred in an algorithm. Nevertheless, this way is still problematic during data structure presentation (image transition and its process) [5]. This is due to the dynamic and complicated transitions in an algorithm will be hard to be grasped by students if it just depends on static images. Moreover, another possibility is that the tutors could teach them so fast before students can digest the previous process [6]. The sequences of the illustration' processes of an algorithm can be also drawn into a board. This way, in terms of understanding, the students maybe are able to follow, but it is time consuming.

Therefore, to resolves the issues, tutors or lectures started to use algorithm visualization (AV) system [7]. AV is a system that is created to visualize the processes of an algorithm in DSA subject. Meaning, the presentation of dynamic visualizations (animation) can ease in learning and teaching environment.

Originally, AV system was utilized passively as animation movie. Nevertheless, the researchers, for instance, [8] confirmed from the result of their research that AV movie, which is presented passively, only gave a little influence to students' comprehension.

The AV system needs to be built actively. In other words, interactivity has to become the most important components in AV system. As proved from studies by [8, 9, 10] that stated the students' engagement through active learning environment is the key success of leaning AV system. The motivation of students is boosted up through active AV learning.

Naps et al. [9] have listed six interactive levels of student during learning AV system, namely: no viewing, viewing, responding, changing, constructing, and presenting. The first is the lowest level, which is no viewing. No viewing only provides plain explanation towards an algorithm, which is usually presented using texts and images. On the other hand, presenting is the highest level of interaction. Hence, many studies of AV [11-15] highly suggested to include interactivity as the most important elements in order to guarantee engaging learning environment among DSA students.

Not only to be active, the variety of interaction level is also needed to keep the more engagement and motivation in long run [16-17]. For examples, game concept activity, problem-solving exercises, mouse viewing activity, and so forth.

Understanding Data Structures and Algorithm (DSA) course which contains the arrangement of algorithm theory is a truly difficult errand in faculty of computer science [2, 18]. DSA is one of the most imperative subjects, but the abstract nature of it makes this subject becoming the hardest one to learn [14]. Generally, the difficulty relies on algorithm itself which derived from dynamic step by step processes.

However, despite many studies have conducted AV study, but the study of AV on mobile platform is still lacking [4]. In conjunction, this study tries to review previous studies, involving 10 models and guidelines, to define the research gap in proposing the principles of interactive algorithm visualization on hybrid mobile application (INAVOHMA).

2. REVIEW AND CRITIQUE ON GUIDELINES AND MODEL IN PREVIOUS AV STUDY

This section as the main purpose of this paper tries to review the existing studies of AV: guidelines, model and principle.

2.1 Algorithm Visualization Pedagogical Guidelines

Initially, reference [19] have created the first guideline for an algorithm visualization system:

1. **AV for huge audiences:** this means AV system has to be in platforms that can reach audiences in the world as much as possible.
2. **General purposed system:** for supporting multitude animation in AV, it is much recommended to have a common user interface.
3. **Input in AV system:** being active is a key success in learning AV system. Thus, input feature within AV system is really imperative to grasp the lesson faster.
4. **Capability to rewind:** sometimes students cannot catch up the lesson straight away. Thus, it is very suggested to have rewind feature in displaying step by step processes, guarantying the lesson is digest well.
5. **Structural view:** the AV need to be flexible also for advance users. Meaning, the students can go to any lines in order to skip through the lines they have mastered before.
6. **Interactive prediction:** Besides triggering students with input feature, interactive prediction is also highlighted. The AV should have feature like pop-up question during animation, asking student to predict what the next line will be.
7. **Quiz interaction:** after the sequences of watching the algorithm visualization and answering pop-up question features. The student is then directed to strengthen his understanding through quiz activity.
8. **Hypertext description:** this means the plain explanation will be displayed for every single line during presentation of algorithm animation.
9. **Smooth animation:** During watching animation, student has control to make the presentation becoming smooth, from one step and afterward. Also, he can make another mode, which is discrete mode animation.

2.2 Design guidelines Algorithm Visualization from constructivism learning theory

Reference [26] in the other side proposed five guidelines that puts focus on constructivism learning theory:

1. Principle experimentation is the essential role.
2. Performs the algorithm visualization based on real world model. For example: demonstrating bubble short algorithm animation in form of American football games animation (see Figure 2.3) [25].

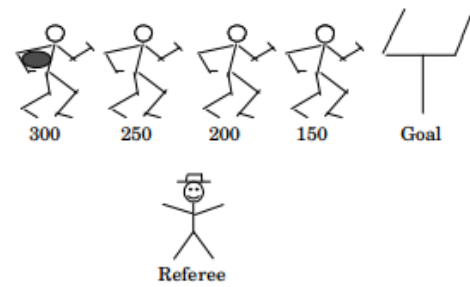


Fig 1: Learning Bubble short algorithms in form of American football games

3. **Direct manipulation:** this direct manipulation is a guideline based on constructivism theory that prompts student to change the algorithm line or animation, such as variable values or positions in array, generating the feedback of algorithm visualization.
4. Provide sound as additional narration to strengthen the understanding of plain description during animation.
5. **Sharing:** To get wider feedback from audiences. This guideline enables students to share their own algorithm visualization to be viewed and commented by friends or others.

2.3 The Data Structure Learning (DSL)

Reference [16] highly emphasized on his AV guidelines to incorporate aural instruction within algorithm visualization. This has proven in his experimental study that the students' response time and performance is better than the one who just used textual explanation only. The reason is based on cognitive load theory [20] that elucidate the usage of dual channel (the integration of voice and text at the same time) to reinforce better comprehension towards algorithm animation by reducing cognitive load to the most minimum point. In other words, concurrent for both (text and sound) is highly recommended, instead of using consecutive way.

2.4 Algorithm Visualization with Platform of VizAlgo

Reference [21] utilized a plugin provided by VizAlgo to enhance data structure learning. He declared some techniques to present AV system, as follows:

1. **Screen design and multiple views:** This is arranged for beginner user in order to reduce cognitive load through abstraction. The conversion of complex portion into simpler forms is highly suggested. For instances, some complicated operation in algorithm can be hidden or excluded at first in which the basic concept is displayed first and then deeper explanation follows based on user control. Another, it can be in form of skipping step refinement approach. It is also so recommended to make subdivision in screen as functional areas. These divisions contain related information. The different facet of an algorithm is shown in different view in which this way is conceptually easier and simple to construct than united outlooks.
2. **Colors:** colors are used to give attention to particular information or data, helping to classify structural elements in order to catch the alterations in time. According to [11], color improves attraction, comprehensibility, memorability, and believability. Nevertheless, the principle use of color is really not easy to adapt. This is due to the subjectivity of color

perception; the complicated design from color can provoke disadvantages or drawbacks. It is much recommended to be more cautious when using colors.

3. **Sound:** Sound is able to decrease the visual clutter of presented user interfaces. Yet, some problems can be raised too when applying sound. For example, the problem happens when all student play all software in the same room, meaning there are more than one audio output can make confusion. The same goes to sound; it is advocated to be more careful to apply it.
4. **Interaction:** Interactive feature has to become the most important element in visualization, allowing student to step through the operational algorithm in control, providing input activity for an algorithm, etc.

2.5 The Data Structure Learning (DSL)

Reference [22] itemized AV principles according to Visual LinProg (AV web-based educational tool) (see Figure 2.5), they are:

1. The usage of images in line with cognitive load theory.
2. Active user with interactivity method.
3. The Appropriate color has to be selected in presented AV.
4. The control that provides skip feature in lines algorithm, once it's been mastered.
5. The exhibition of one interesting event per step.

2.6 AV System on Mobile Platform

Reference [23] endeavored to develop an AV system, specifically for sorting algorithm, as mobile application. Nonetheless, the AV system only puts attentions with the lowest level of interactivity only, which is viewing level. Here are some guidelines that he suggested for AV system on mobile platform:

1. Since mobile has limited screen size, omitting or decreasing extra amount of textual explanation is really vital.
2. To make sure the readability and clarity of AV on mobile, the proper color has to be selected carefully.
3. The use of symbols (icon and images) has to be limited.
4. The system should support landscape and portrait mode.
5. Pop-up feature to enable users seeing advance information on small screen.
6. Using tabs to provide users seeing other additional information in limited screen.
7. The hierarchical structure is utilized in order to expand and hide groups of items on limited screen size.

2.7 Sortko System Model

The same as before, Sortko system is a mobile app for android that is also developed for learning sorting algorithm [24]. Different with previous, this app uses switching method for interactivity. Particularly, the student is driven to switch a sequence of arrays in order to sort based on particular algorithm that he has chosen before.

All in all, the Sortko was created based on four following modules:

1. The interactive system module: this module consists of interactive algorithm sorting that has capability to check either the answers given by user is right or wrong.
2. Scaffolding module: this module is in charged to help students automatically when student performing wrong step of switching by showing him a clue or guideline message. The message contains the logic clue so that student can understand it.
3. Motivational module tracks: this module is responsible to calculate the performance of a user, such as speed, accuracy, time, etc. These all points are the calculated at last, to be showed towards student.
4. The graphical user interface (GUI) module: this module is about user interface that Sortko has, which entails of three views: opening, sorting, and outcome (result). The all views are arranged in a horizontal view, guaranteeing enough use of limited space in small device screen.

2.8 Method Visualization DSA Model

Different from all, the research by [2] tried to integrate program animation approach into algorithm visualization. In other words, the user is able to see both animation in perspective of algorithm visualization (concept), but also in terms of programming language (coding). The programming languages they used are C and Pascal. They called this as DSA model.

3. METHODOLOGIES

In order to achieve the objectives, this research utilizes two phases of activities, namely (i) literature search and (ii) critical analysis, as displayed in Figure 2.

The gathering of existing (guidelines and models) of AV study is in the first phase. In this phase comprehensive literature search were conducted. There were 10 previous (guidelines and models) gathered. Then, in the second phase, these gathered studies were critically analyzed and reviewed, which means to attain the first objective of this ongoing study.

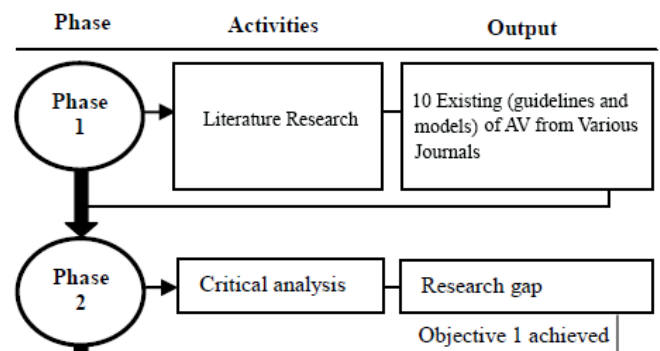


Fig 2: Summary of Activities

4. FINDINGS AND DISCUSSIONS

The review and critical analysis over the 8 existing study of AV has been performed to identify the research gaps in this study.

4.1 Research Gap

Therefore, according to the critical analysis and review, the following research gaps are then summarized:

1. Principles or Guidelines of AV particularly settled on mobile platform are highly rare.

2. Most of the previous studies of AV are much focusing on Desktop and Web Platform.
3. Even though, there are two AV on mobile platform, but they just solely focus on sorting algorithm and also the interactivity level of them are still low.
4. Some design elements and guidelines that should be considered when developing the AV system on mobile platform are not clearly identified in the existing guidelines.
5. The two existing mobile AV study still focus on one mobile operating system only, which is Android.

5. CONCLUSION AND FUTURE WORKS

Overall, the existing literatures elucidate that all the reviewed models and guidelines have their own particular characteristic. Nevertheless, it was indicated lacking of principles or guidelines that directs on mobile platform. Mostly, existing studies address towards desktop platform and website. Even though, there are two guidelines address upon mobile platform, yet they are still lacking of variety of interactions and user interface principles. Additionally, the principles such as design elements and user interface are not identified comprehensively. Therefore, it is highlighted that these become the research gaps in this study.

At this rate, this paper has elaborated the achievement in finding research gaps in this study by conducting comparative analysis from existing studies. Next, this study will continue to form meaningful the principles of interactive algorithm visualization on hybrid mobile application (INAVOHMA). The reason of applying on hybrid mobile application (HMA) is to guarantee that the app can be applicable for various mobile operating systems (such as, Android, IOS, BlackBerry, etc.). Particularly, the future works will be working on comparative analysis in order to find the frameworks for strategy of interactivity in algorithm visualization as well as some other theories of design elements and user interface.

6. REFERENCES

- [1] Crescenzi, Pilu, Malizia, Alessio, Verri, M Cecilia, Díaz, Paloma, & Aedo, Ignacio. (2012). Integrating Algorithm Visualization Video into a First-Year Algorithm and Data Structure Course. *Educational Technology & Society*, 15(2), 115-124.
- [2] Osman, Waleed Ibrahim, & Elmusharaf, Mudawi M. (2014). Effectiveness of Combining Algorithm and Program Animation: A Case Study with Data Structure Course. *Issues in Informing Science and Information Technology*, 11.
- [3] Sadikan, Siti Fairuz Nurr, & Yassin, Siti Fatimah Md. (2012). Role of Interactive Computer Programming Courseware in Facilitating Teaching and Learning Process Based on Perception of Students in Bangi, Selangor, Malaysia. *Learning*, 3, 0.51.
- [4] Patel, Shirin. (2014). A Literature Review on Tools for Learning Data Structures.
- [5] del Vado Vírveda, Rafael. (2010). *A visualization tool for tutoring the interactive learning of data structures and algorithmic schemes. NOTE FROM ACM: It has been determined that the author of this article plagiarized the contents from a previously published paper. Therefore ACM has shut off access to this paper.* Paper presented at the Proceedings of the 41st ACM technical symposium on Computer science education.
- [6] Halim, Steven, Koh, Zi Chun, Loh, V, & Halim, Felix. (2012). Learning Algorithms with Unified and Interactive Web-Based Visualization. *Olympiads in Informatics*, 6, 53-68.
- [7] Shaffer, Clifford A, Cooper, Matthew L, Alon, Alexander Joel D, Akbar, Monika, Stewart, Michael, Ponce, Sean, & Edwards, Stephen H. (2010). Algorithm visualization: The state of the field. *ACM Transactions on Computing Education (TOCE)*, 10(3), 9.
- [8] Hundhausen, Christopher D, Douglas, Sarah A, & Stasko, John T. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing*, 13(3), 259-290.
- [9] Naps, Thomas L, Rößling, Guido, Almstrum, Vicki, Dann, Wanda, Fleischer, Rudolf, Hundhausen, Chris, Rodger, Susan. (2002). *Exploring the role of visualization and engagement in computer science education.* Paper presented at the ACM SIGCSE Bulletin.
- [10] Urquiza-Fuentes, Jaime, & Velázquez-Iturbide, J Ángel. (2009). A survey of successful evaluations of program visualization and algorithm animation systems. *ACM Transactions on Computing Education (TOCE)*, 9(2), 9.
- [11] Adamchik, Victor. (2011). *Data structures and algorithms in pen-based computing environments.* Paper presented at the Global Engineering Education Conference (EDUCON), 2011 IEEE.
- [12] Burguillo, Juan C. (2010). Using game theory and competition-based learning to stimulate student motivation and performance. *Computers & Education*, 55(2), 566-575.
- [13] Moons, Jan, & De Backer, Carlos. (2013). The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, 60(1), 368-384.
- [14] Sadikan, Siti Fairuz Nurr, & Yassin, Siti Fatimah Md. (2013). A Conceptual Framework for C Programming Learning Through Basic Game Development Courseware. *International Journal of Scientific & Enggining Research*, 4(8).
- [15] Segura, Clara, Pita, Isabel, del Vado Vírveda, Rafael, Saiz, Ana Isabel, & Soler, Pablo. (2008). Interactive Learning of Data Structures and Algorithmic Schemes *Computational Science-ICCS 2008* (pp. 800-809): Springer.
- [16] Alhosban, Fuad, & Hamad, Mousa. (2011). *The Effectiveness of Aural Instructions with Visualisations in E-Learning Environments.* Durham University.
- [17] Karavirta, Ville, & Shaffer, Clifford A. (2013). *JSAV: the JavaScript algorithm visualization library.* Paper presented at the Proceedings of the 18th ACM conference on Innovation and technology in computer science education.
- [18] Crescenzi, Pilu, Malizia, Alessio, Verri, M Cecilia, Díaz, Paloma, & Aedo, Ignacio. (2012). Integrating Algorithm Visualization Video into a First-Year Algorithm and

- Data Structure Course. *Educational Technology & Society*, 15(2), 115-124.
- [19] Rößling, Guido, & Naps, Thomas L. (2002). A testbed for pedagogical requirements in algorithm visualizations. *ACM SIGCSE Bulletin*, 34(3), 96-100.
- [20] Sweller, John. (2011). Cognitive load theory. The psychology of learning and motivation: Cognition in education, 55, 37-76.
- [21] Simonák, Slavomír. (2013). Algorithm Visualization Using the VizAlgo Platform. *Acta Electrotechnica et Informatica*, 13(2), 54.
- [22] Lazaridis, Vassilios, Samaras, Nikolaos, & Sifaleras, Angelo. (2013). An empirical study on factors influencing the effectiveness of algorithm visualization. *Computer Applications in Engineering Education*, 21(3), 410-420.
- [23] Meolic, Robert. (2013). *Demonstration of Sorting Algorithms on Mobile Platforms*. Paper presented at the CSEDU.
- [24] Boticki, Ivica, Barisic, Ante, Martin, Sergio, & Drljevic, Neven. (2013). Teaching and learning computer science sorting algorithms with mobile devices: A case study. *Computer Applications in Engineering Education*, 21(S1), E41-E50.
- [25] Hundhausen, Christopher, & Douglas, Sarah. (2002). A language and system for constructing and presenting low fidelity algorithm visualizations *Software Visualization* (pp. 227-240): Springer.
- [26] Lee, Ming-Han, & Rößling, Guido. (2011). *Constructivist and Constructionist Approaches to Constructing Algorithm Visualizations: A Proposal*. Paper presented at the 2011 11th IEEE International Conference on Advanced Learning Technologies.