

Research Article

Developing an Algorithm Learning Tool for High School Introductory Computer Science

Aimee Theresa Avancena,¹ Akinori Nishihara,¹ and Chika Kondo²

¹Tokyo Institute of Technology, 2-12-1-W9-108 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

²Tokyo Tech High School of Science and Technology, 3-3-6 Shibaura, Minato-ku, Tokyo 108-0023, Japan

Correspondence should be addressed to Aimee Theresa Avancena; aimeetheresa@yahoo.com

Received 30 November 2014; Accepted 21 February 2015

Academic Editor: Shu-Sheng Liaw

Copyright © 2015 Aimee Theresa Avancena et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the initial stage of developing an algorithm learning tool for the students of the Information Systems course at Tokyo Tech High School of Science and Technology in Japan. The tool applies the concept of Algorithm Visualization (AV) technology and was used as an aid for learning basic algorithms such as searching and sorting. Two AV types were included in the tool, one with more input options and control and the other with less. Previously proposed AV evaluation properties and the Categories of Algorithm Learning Objectives (CALO) were considered in designing the tool's evaluation questionnaire. Written tests based on CALO were also designed. Posttest results indicate moderate improvement in the performance of the students. Test results also show that student abilities match some of the algorithm learning objectives. The students who used the AV with more options have a slightly higher gain score average in the posttest compared with those who used the AV with limited control. Overall assessment indicates a positive evaluation of the tool and signifies the students' preferred AV characteristics. After factor analysis of the evaluation questionnaire, three factors were extracted which correspond to the suggested AV evaluation properties. These results may be used in improving the learning tool and the evaluation questionnaire.

1. Introduction

With computer science (CS) becoming a more regular part of the K-12 curriculum the need to address the learning performance of the students has increased. There is also a need for appropriate tools that assist learning among novice programmers. In relation to these, a tool for learning basic search and sorting algorithms was created for the students of the Information Systems course track of Tokyo Tech High School of Science and Technology. The students of this track undergo a specialized curriculum that is designed to prepare them for a computer and engineering related degree in the university. The target participants for the initial stage of this study belong to Information Systems Class 2014. According to their instructor, some of the students in the said class are not performing as expected and have low motivation for learning computer science. As these students were entering their final year in high school it was deemed necessary that they have a good grasp of fundamental CS topics before taking up

the required advanced courses. In order to address this problem, a special lecture on fundamental algorithms was conducted for the class using the learning tool as instructional aid. This is in accordance with the ACM Computing Curricula's proposition that a good foundation on algorithms and their implementation is necessary to gain programming skills and advanced computer science concepts [1].

In this paper, the initial stage of the design and development of the online algorithm learning tool and its pilot implementation among the students of Information System Class 2014 are introduced. One of the two phases of the entire research is also discussed in this paper. For the preliminary stage of the study, the goal is to verify if there is an improvement in the learning performance of students after using the algorithm learning tool, which incorporates Algorithm Visualization technology or AV. Another objective is to compare the effects of the AV that offers more control and interaction with the one that offers limited menu options. Hence, the learning tool was designed to have two types of AV,

one with more input options and control and the other with less.

This paper also tackles one phase of the research which entails the design, implementation, and analysis of two evaluation instruments. One instrument is a questionnaire for evaluating the usability and pedagogical effectiveness of the algorithm learning tool and the other is a written test on algorithms. The design of both instruments was based on proposed AV evaluation properties and algorithm learning objectives. The learning tool's evaluation questionnaire was used to verify the intended differences in the features between the two types of AV offered by the learning tool. It was also examined to see how it can be improved and revised. The revisions are to be verified in the next stages of the learning tool's implementation. The written test on algorithms, on the other hand, was mainly used to measure the effects of the tool on the learning performance of the student participants.

A brief background on Algorithm Visualization, which is the main feature of the learning tool created for this research, is the topic of the next section. The research framework and future stages of the study are explained in Section 3. The development of the algorithm tool is discussed in Section 4. Further details on the two evaluation instruments designed for this study are provided in Section 5. The results of the data analysis for the initial implementation of the algorithm learning tool are presented in Section 6 and the summary and conclusion for this research phase are stated in Section 7.

2. Algorithm Visualization

Algorithm Visualization or AV is a technology that uses graphics and animation of algorithms. Simulation of the algorithm process is done through graphical images which the user can control [2]. The papers of Saraiya [3, 4] provide a more comprehensive report of the existing and nonaccessible AVs. Another good resource on AVs is the Algorithm Visualization (AlgoViz) portal created by Virginia Tech University [5].

The main goal of AV is to help improve computer science education [6]. In the mid-1990s research on AV shifted from innovative features such as displays, specification techniques, and interaction techniques to its educational contribution [3, 4]. Recent experiments were carried out to validate the effectiveness of AV as an instructional material [7]. These studies present varying results from “no significance” to positive educational impact [8]. Studies that showed positive impact of AV systems focus on the features that make them effective [9]. Features considered helpful for learning are narrative and textual contents, feedback on students' actions, extra time to use AV for nonanimated tasks, input and control menus for the animation, variable state changes, integrated development environments, window management, and pseudocode display [3, 10]. A visualization that allows more control of the simulation and supports student interaction and active learning is found to be more helpful and effective [3, 4, 11].

Student “engagement” is considered to be a factor that can make AV educationally effective [11]. Moreover, the manner with which the students use visualization is deemed more

important than the visualizations themselves [6]. An “engagement” taxonomy defined in the working group “Improving the Educational Impact of Algorithm Visualization” is proposed to serve as a framework for researches in determining the pedagogical effectiveness of AV [11]. This taxonomy is composed of six categories.

- (1) *No viewing*: refers to instruction without using any form of Algorithm Visualization.
- (2) *Viewing*: refers to having users watch several visual representations of the algorithm being studied.
- (3) *Responding*: requires the learners to reply to questions related to the visualization displayed by the system.
- (4) *Changing*: entails modifying the visualization such as setting different input values to test various cases.
- (5) *Constructing*: allows the users to make their own visualization of the algorithm.
- (6) *Presenting*: requires the students to present visualization to an audience for feedback and discussion.

For the algorithm learning tool created for this study, the “no viewing,” “viewing,” and “changing” categories were employed. The learning tool offers “no viewing” through the lecture notes on the algorithms. “Viewing” and “changing” were incorporated in the menu and control options for setting and running the Algorithm Visualization.

As any software system requires assessment, Algorithm Visualization tools also have to be evaluated in terms of their pedagogical effectiveness. The study of Lee and Rößling proposed three properties with which AVs can be analyzed and evaluated.

- (1) *Symbol system*: refers to texts, graphics, sounds, and animations.
- (2) *Interactivity*: deals with user input engagement.
- (3) *Didactic structure*: refers to pedagogical-based system design [12].

According to the said study, the third property needs more investigation. In connection to this, they proposed the Categories of Algorithm Learning Objective or CALO to serve as a pedagogical framework for designing and structuring AV. They suggested the use of CALO in setting the objectives for exams and as a self-evaluating tool for learners [12]. For this research, CALO was used as basis for the contents of the written tests on algorithms and for some of the items in the questionnaire for the usability and pedagogical assessment of the learning tool.

3. Research Design and Methodology

This study entails the design and development of an algorithm learning tool intended for a high school introductory computer science class. The tool was designed with the initial objective of creating an instructional aid for the students of the Information Systems course at Tokyo Tech High School. The ultimate goal is to develop a tool that addresses both the learning motivation and performance of these students.

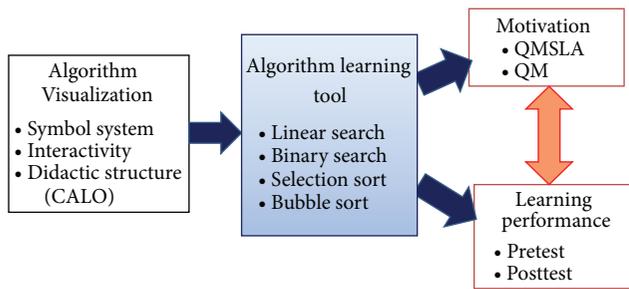


FIGURE 1: Research framework.

Hence, the entire research was divided into two phases. This paper presents only the phase that deals with the effects of the tool on learning performance. The other phase, which deals with learning motivation, is only briefly mentioned in this section. The subsections below describe the proposed framework of the entire research and the implementation plans specific to the phase presented in this paper.

3.1. Research Framework. As shown in Figure 1, the main component of this research is the algorithm learning tool which incorporates Algorithm Visualization (AV) as its main feature. The learning tool tackles four basic algorithms, Linear Search, Binary Search, Selection Sort, and Bubble Sort. These algorithms were chosen because they are included in the curriculum of the target students. The other algorithms included in the school's curriculum may be added in the future extensions of this research.

The framework in Figure 1 also depicts that the suggested evaluation properties for AVs [12] were incorporated in the assessment of the learning tool. In particular, the items of CALO were used in the tool's evaluation questionnaire. One objective is to determine which among the learning tool's features based on the suggested AV evaluation properties can help increase the learning performance and motivation of the students.

Aside from the evaluation questionnaire that was designed to assess the usability and pedagogical effectiveness of the learning tool, other instruments were also developed for this study. These instruments include written tests on algorithms and two questionnaires on motivation. One questionnaire (QMSLA: Questionnaire on Motivation, Self-Efficacy, and Learning Attitudes) was based on the Motivation and Learning Strategies Questionnaire (MLSQ) [13] and the other (QM: Questionnaire on Motivation) was based on the ARCS model [14]. The goal for designing these questionnaires is to determine the motivation components for learning fundamental computer science topics specifically algorithms. The analysis of the motivation questionnaires is included in the other phase of the study, which is not presented in this paper. The phase of the study presented in this paper involves only the analysis of the questionnaire on the usability and pedagogical effectiveness of the learning tool and the written test on algorithms. These two evaluation instruments are further discussed in Section 5.

In general, the main research question this study would like to answer is, "How can an online learning tool with Algorithm Visualization enhance the learning performance and motivation of high school students in an introductory computer science course?" In order to address specific issues relevant to the main research problem, the following questions for analysis were formulated.

- (1) Is there an effect in the learning performance of students after using the algorithm learning tool?
- (2) Is there a difference in the learning improvement between the group that had more input options and control of the Algorithm Visualization and the group with fewer options and control?
- (3) What corresponding tasks based on CALO can the students perform after using the learning tool?
- (4) Which among the features of the algorithm learning tool with Algorithm Visualization are favorable to the learners?
- (5) Are the scales and items used for the questionnaire appropriate for evaluating the algorithm learning tool?

The above questions for analysis were considered in the phase of the research presented in this paper. These questions deal mainly with the effects of the algorithm learning tool on the performance of the students and with the usability and pedagogical design and assessment of the tool. Other questions for analysis are addressed in the other phase of the study, which deals mainly with the effects of the learning tool on the motivation of the students.

3.2. Implementation and Data Gathering. The implementation of the learning tool is planned to be carried out in several stages. The first stage presented in this paper is the pilot implementation conducted among the students of the Information Systems course Class 2014 of Tokyo Tech High. It can be said that the original plan for the learning tool was to be an instructional aid for this class which was entering their final year of high school. Thirty-five (35) students from the said class participated in the study. These students have already studied the lesson on algorithms six months prior to the implementation of the research. However, their performance in the midterm examination on algorithms was not satisfactory according to their instructor. A learning reinforcement activity was thought to be necessary for the class because these students still had one more year of computer science course track and would undergo advanced CS subjects. Therefore, a special remedial lecture was given to them around the end of the school term.

For the implementation among Class 2014, the algorithm learning tool was used as an instructional material during the lecture. The lecture lasted for forty (40) minutes. The students were also given another forty (40) minutes to use the tool for individual learning of the algorithms during which the class was divided into two groups based on their score in the midterm exam on algorithms. Eighteen students who had scores of 76% and above were assigned

to group A and seventeen students with scores below 76% were placed in group B. This grouping scheme is based upon the request of the class instructor and is in accordance with the original intention for creating the learning tool, that is, to have the lower performing students (group B) benefit from the learning tool with more control and menu options (AlgoVis1).

Three weeks before the lecture and individual study using the algorithm learning tool, the students had to take the written pretest on algorithms. They also answered the presurvey motivation questionnaires. The students took the same written test on algorithms as posttest after the lecture and self-study. The evaluation questionnaire on the usability and pedagogical effectiveness of the learning tool and the postsurvey on motivation were also answered by the students.

The next stages of the implementation of the algorithm learning tool are to be conducted among the lower batches of students of the Information Systems course track of Tokyo Tech High. For these subsequent implementations, another grouping scheme for the students will be used. The plan is to have an almost equivalent distribution of students that will lessen the qualifications gap between the two groups. Moreover, the evaluation questionnaires designed for this study, namely, the questionnaire on the usability and pedagogical effectiveness and the questionnaire on motivation, are to be revised based on the results of the initial implementation. The revised questionnaires are to be conducted and validated in the succeeding implementations of the algorithm learning tool.

4. Development of the Algorithm Learning Tool

The algorithm learning tool comes as a web-based lesson on four basic algorithms included in the curriculum of the participating class in the Japanese high school. These algorithms are Linear Search, Binary Search, Bubble Sort, and Selection Sort. The design of the tool is based mainly on the “engagement” taxonomy levels, “no viewing,” and “viewing” [11] so it provides both lecture notes and visualizations. The lecture notes include descriptions, pseudocode, and illustrations of the algorithms all designed for novice learners. The notes also offer English and Japanese translations. Figure 2 shows the screenshot of the lecture notes on the Linear Search Algorithm.

In order to provide student interaction, the Algorithm Visualization part incorporates features such as textual contents, feedback, input and control menus for the animation, variable state changes, and pseudocode display [3, 10]. Two types of visualizations are offered by the learning tool: AlgoVis1, which allows more input options and control, and AlgoVis2, which has limited input options and control of the animation. They were named as such only for the purposes of this research.

The main features of the algorithm learning tool are enumerated as follows.

- (a) *Input and Control Panel.* This is where the users can manage the settings on how the algorithm simulation



FIGURE 2: Screenshot of the lecture notes on Linear Search.

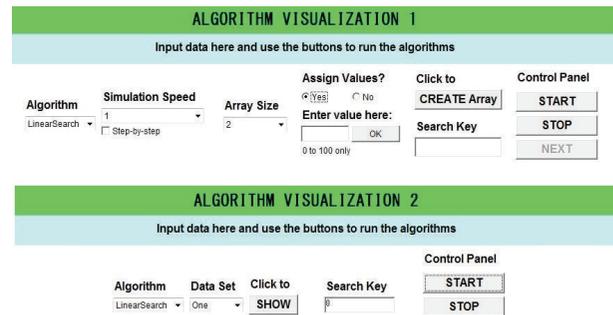


FIGURE 3: Input and control panels for AlgoVis1 and AlgoVis2.

should run. Figure 3 shows the input and control panels of the two visualization types. AlgoVis1 allows the users to choose the algorithm, the speed of simulation, and the manner of simulation, whether step-by-step or straightforward. The data array used in the simulation may vary in size and can be initialized. Boxes and buttons for entering values and for running and terminating the algorithm simulation are also provided. Users of AlgoVis2 can only set the algorithm to simulate and choose from five sets of values for the data array. These features were incorporated following the taxonomy of learner engagement particularly the “viewing” and “changing” levels [11].

- (b) *Algorithm Simulation Field.* This is considered as the main part of the Algorithm Visualization where the data array used for the searching and sorting animation is shown. The only difference between AlgoVis1 and AlgoVis2 is the height of the arrays. For AlgoVis1 the height of the array element corresponds to the assigned number value while for AlgoVis2 all the array elements are of the same height.
- (c) *Pseudocode Display.* To the right of the simulation field, a C-like code of the algorithm being run is displayed. Code tracing is done during simulation by highlighting the particular line that is being executed.
- (d) *Variable Display and Message Box.* These two sections show the changes in the local variables and the line by line descriptions of the running program and other appropriate messages. AlgoVis1 provides

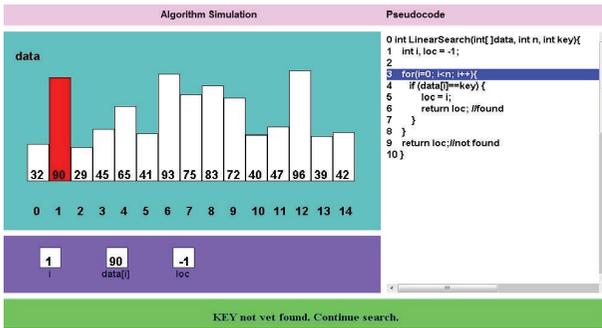


FIGURE 4: Algorithm simulation field, pseudocode display, variable, and message fields for AlgoVis1.

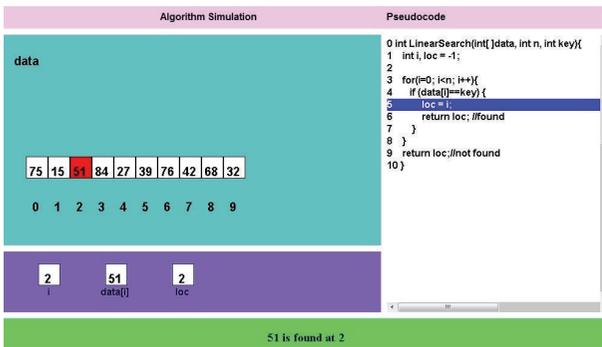


FIGURE 5: Algorithm simulation field, pseudocode display, variable, and message fields for AlgoVis2.

more feedback to the user compared to AlgoVis2. The last three stated features of the two types of visualization for AlgoVis1 and AlgoVis2 are shown in Figures 4 and 5, respectively.

5. Evaluation Instruments

The research phase presented in this paper is concerned with the analysis of the evaluation questionnaire and the written test on algorithms. The 35-item evaluation questionnaire specifically developed for this study was used to assess the usability and pedagogical effectiveness of the algorithm learning tool. Five categories or scales were initially considered in the questionnaire: (1) General Ease of Use, (2) Interface Assessment, (3) Algorithm Visualization's Characteristics, (4) User's Opinion, and (5) Algorithm Learning Objectives (see the Appendix). These scales and their corresponding items were designed only for the purposes of this research, except for the items that were based on CALO. The eight (8) items of the last category of the evaluation questionnaire were patterned on the seven nonhierarchical learning objectives normally used in CS education on which CALO is based:

- (1) *Descriptive*: discerning and describing algorithms;
- (2) *Demonstrative*: demonstrating algorithms with graphics or objects;
- (3) *Decoding*: following and tracking algorithms;

- (4) *Coding*: reproducing learned algorithms;
- (5) *Evaluative*: analyzing, comparing, and evaluating algorithms that solve the same set of problems;
- (6) *Appropriative*: writing a complete program; evoking, extending, or modifying learned algorithms to solve a given problem;
- (7) *Originalive*: developing own algorithms to solve unfamiliar problems [12].

The above learning objectives as well as the standard test used by the school were used as guidelines for the format and contents of the written test on algorithms. The 30-point algorithm test is composed of three parts: identification, code completion, and algorithm simulation. Conceptual and procedural question items on the four algorithms were included in the design of the test. Four of the learning objectives from CALO were integrated in each part of the test. Part I of the test was designed after the "Descriptive" category with items that require the student to identify the algorithms. Part II deals with the "Coding" category because this part entails filling in the missing lines or codes of the algorithm. In Part III, the students are asked to manually demonstrate the algorithm steps and to provide the output of the algorithm. These tasks correspond to the "Demonstrative" and "Decoding" categories.

The evaluation questionnaire and the written test on algorithms were translated to Japanese. Moreover, the written test on algorithms had to be checked and approved by the class instructor to ensure that the contents are within the scope of the students' learning goals. It was conducted before and after the implementation of the learning tool as pretest and posttest, respectively. The evaluation questionnaire on the other hand was conducted as a postsurvey among the student participants.

6. Results and Discussions

The written pretest and posttest on algorithms and the questionnaire on the usability and pedagogical effectiveness of the learning tool were implemented as evaluation instruments. In order to determine the effects of the algorithm learning tool on the performance of the students, a series of statistical analysis was conducted using the data gathered from the said instruments. The results are presented in the subsections below.

6.1. Effects on Learning Performance. This subsection answers the following questions: "Is there an effect on the learning performance of students after using the algorithm learning tool?" "Is there a difference in the learning improvement between the group that had more input options and control of the Algorithm Visualization and the group with fewer options and control?," and "What corresponding tasks based on CALO can the students perform after using the learning tool?"

The chart in Figure 6 depicts the scores in the tests. The blue line indicates the pretest scores and the red line refers to the posttest scores. Comparing the results of the pretest

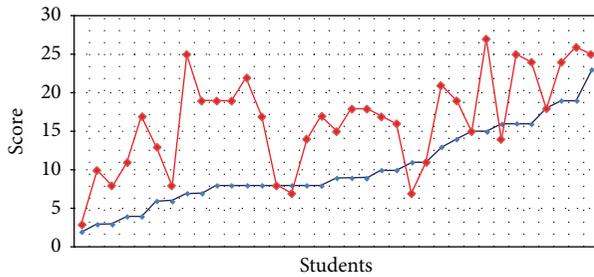


FIGURE 6: Line graph of the scores in the pretest and posttest.

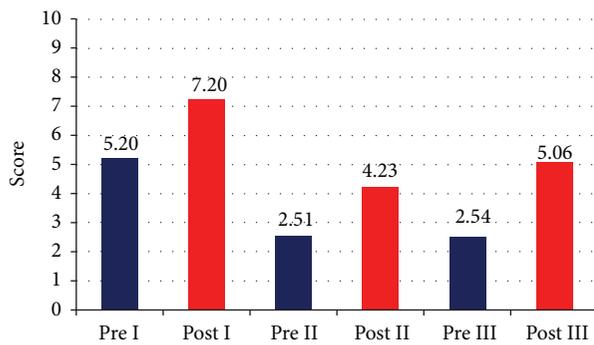


FIGURE 7: Breakdown of the scores for each part of the algorithm test.

and posttest on algorithms, the scores of the participants generally improved after having used the learning tool except for three students who scored lower in the posttest and four who retained their pretest score. This result is relevant to the plan of using the tool as instructional aid for the class' remedial lecture and to the goal of reinforcing the students' knowledge on basic algorithms.

The mean scores in the three parts of the tests as shown in Figure 7 also signify that there is a moderate increase in the performance of the students for each part type of the exam: (1) identification, (2) code completion, and (3) algorithm simulation. Pre I is the score in part I of the pretest and Post I is the score in part I of the posttest, Pre II is the score in Part II of the pretest, and so on.

In order to further verify if the increase in the performance of the students after using the learning tool is significant, *paired-samples t-test* was conducted to compare the scores of the 35 students in the pretest and posttest. Results indicate a significant difference in the scores of the students in the pretest ($M = 10.26$ and $SD = 5.135$) and posttest ($M = 16.49$ and $SD = 6.242$) with $P < 0.001$. There is also a significant increase for each group, $P < 0.001$ for both group A and group B. The difference in the performance in the posttest of the two groups was also checked using *independent samples t-test*, and a P value 0.036 was obtained.

When considering the mean scores of the two groups in the pretest and posttest, it can be noticed that students of group A are the higher performing group while those in group B are the lower performing students. Table 1 presents the mean scores in the pretest and posttest of the two groups.

The scores in the posttest still indicate that group A students performed better than the students of group B. Therefore, there is a need to determine the difference in the increase in the test performance between the two groups. This was done by calculating the gain score, that is, by subtracting the pretest score from the posttest score. Table 2 shows the total gain scores and the breakdown of the scores in each test part. Looking closely at the values, group B students have a little higher average gain score compared to the students of group A. Group B's gain score average in the identification (Part I) and code completion (Part II) parts is also higher than those of group A.

Based on the differences in the mean scores of the pretest and posttest, the group that used the algorithm learning tool that has more control of the visualization or group B has a slightly higher increase in the posttest performance as indicated by the average gain score compared to the group (group A) that used the version with less input options and control. To find out the percentage of the lower performing students whose gain scores have been raised in comparison to the higher performing students, independent sample *t-test* and ANOVA were run for group B students with positive gain scores ($N = 12$) and a P value of 0.028 was obtained. This result implies that 70% or more than two-thirds of the students in group B have a higher gain score average than the students of group A after using the algorithm learning tool.

Analysis of Covariance or ANCOVA was then used to support the claim that the higher gain score increase for group B students is an effect of using the Algorithm Visualization with more input options and control. The average gain score was chosen as dependent variable and the posttest score as covariate. This choice was done to prove that despite the higher raw scores in the posttest of group A, the higher average gain score of group B is still significant. The result shows that there is an effect of the covariate posttest on the average gain score with $P = 0.001$. It can then be said that the visualization which offers more input options and control had an effect in raising the scores of the students in group B in comparison to the scores of group A students who used the visualization with limited menu options and control.

Lastly, considering the posttest performance of the students, they have proven to be capable of performing certain tasks based on the CALO categories after using the learning tool. Based on their posttest scores the students improved in their ability to identify algorithms "Descriptive," to fill in missing lines of codes "Demonstrative," and to provide the output of an algorithm simulation "Decoding."

6.2. Evaluation of the Algorithm Learning Tool. In order to find out the opinion of the students about the algorithm learning tool, the evaluation questionnaire on the usability and pedagogical effectiveness of the tool was examined. The question, "Which among the features of the algorithm learning tool with Algorithm Visualization are favorable to the learners?" was answered by examining the responses of the students to the evaluation questionnaire. Primarily, similarities in the favored features between the groups were noticed. As regards the interface, the students of both groups think that the graphics and animation used are appropriate

TABLE 1: Scores of groups A and B in the pretest and posttest.

	Group	N	Mean	Std. deviation	Kurtosis	Skewness
Pretest	A	18	12.50	4.743	-0.460	0.597
	B	17	7.88	4.526	1.250	1.077
Posttest	A	18	18.61	5.479	-0.232	-0.453
	B	17	14.24	6.359	-0.457	0.178

TABLE 2: Average gain scores of groups A and B.

Group	Average gain score	Part I identification	Part II code completion	Part III algorithm simulation
A	6.11	1.78	1.44	2.89
B	6.36	2.24	2.00	2.12

to visualize the algorithms (mean = 3.89 for group A; mean = 4.00 for group B). The two groups also agree that the algorithm animation is helpful in understanding how the algorithm works (mean = 4.17 for group A; mean = 4.47 for group B). This may be due to the fact that there is not much difference in the algorithm simulation field between the two types of AV offered by the learning tool.

The two groups differ in a number of features they prefer which may be due to the varying input and control menu options provided for each AV type. Considering the original category General Ease of Use category, group A students favor the clarity of instructions of the learning tool (mean = 3.89) while group B found *easy navigation* as the most notable aspect (mean = 3.94). Regarding the AV characteristics, group A students think that the capability of the learning tool to do step by step tracing of the algorithm (mean = 3.94) is the most important while group B students think that being able to choose the speed of the algorithm animation is the best feature (mean = 4.47). For the category on the learning objectives, the students of group A think that they are more confident to provide the output of the algorithm simulation using a set of data (mean = 3.44) while group B students give importance to the ability of describing how the algorithms work (mean = 3.35).

The differences in the responses of the two groups to the evaluation questionnaire were determined by using *independent samples t-test*. Table 3 shows that there are significant differences in the answers of the two groups particularly in the items related to the characteristics of the AV types used in the learning tool. These results denote the intended differences in the observation and assessment of the two groups and further confirm the planned variation in the design of the two types of AV, AlgoVis1 with more control and AlgoVis2 with limited control.

The questionnaire was designed specifically for this research so further analysis is needed in order to test its reliability and validity. Construct validity also needs to be established which will be necessary for future revisions of the questionnaire. These issues correspond to the question, "Are the scales and items used for the questionnaire appropriate for evaluating the algorithm learning tool?"

TABLE 3: Differences in the assessment of AV characteristics of the two groups.

Items on Algorithm Visualization characteristics	P
The Algorithm Visualization allows the user to choose the speed of the algorithm animation.	0.000
The Algorithm Visualization allows the user to set the size of the array.	0.005
The Algorithm Visualization allows the user to stop and restart algorithm animation.	0.002
The Algorithm Visualization allows the user to assign the elements of the array.	0.039
The Algorithm Visualization gives appropriate feedback to the user.	0.006

TABLE 4: Alpha values of the evaluation questionnaire categories.

Category	Alpha reliability (N = 35)	Number of items
General ease of use	0.764	4
Interface assessment	0.757	7
AV characteristics	0.861	8
User's opinion	0.766	8
Algorithm learning objectives	0.929	8

In order to answer the question above, Cronbach's Alpha was used to test the internal reliability of the questionnaire and the resulting Alpha value when considering all the items is 0.867. The same test was run to check the reliability of each of the scales of the evaluation questionnaire and the results are shown in Table 4. Algorithm learning objectives has an Alpha value greater than 0.9 which indicates "excellent" internal consistency; AV characteristics have 0.8 which is considered "good" and the rest have Alpha values greater than 0.7 describing "acceptable" internal consistency. Taking into account the categories with "acceptable" internal consistency, revising the evaluation questionnaire is an essential future plan.

TABLE 5: Factor analysis of the evaluation questionnaire.

	Factor loading
Interface assessment factor	
The Algorithm Visualization allows the user to choose the speed of the algorithm animation.	0.801
The Algorithm Visualization gives appropriate feedback to the user.	0.742
The Algorithm Visualization allows the user to assign the elements of the array.	0.711
The Algorithm Visualization allows the user to set the size of the array.	0.710
The Algorithm Visualization asks questions about the next steps in the algorithm simulation.	0.693
The Algorithm Visualization allows the user to choose the algorithm to study.	0.684
The algorithm animation is helpful in understanding how the algorithm works.	0.671
The menu choices for the algorithm learning tool are adequate.	0.662
Setting the size and values of the array is helpful in learning the algorithms better.	0.647
The Algorithm Visualization allows the user to stop and restart algorithm animation.	0.631
The Algorithm Visualization allows step by step tracing of the algorithm.	0.597
It is easy to use control buttons and choice lists in the Algorithm Visualization.	0.480
The graphics and animation used are appropriate to visualize the algorithms.	0.347
Algorithm learning objectives factor	
I can complete the missing code for all the four algorithms I learned.	0.918
I can give the output for a set of data by using algorithm simulation.	0.883
I can compare and analyze algorithms that solve the same problems, for example, search and sorting.	0.868
I can demonstrate how the algorithm works using drawing simulations.	0.793
I can now develop my own algorithms to solve other problems.	0.791
I can describe how the algorithms work.	0.781
I can now identify the algorithm by just looking at the pseudocode.	0.759
I can easily code the algorithms using C programming language or another language I know.	0.726
There is too much text on the pages of the algorithm learning tool.	-0.377
AV characteristics factor	
The algorithm learning tool and the Algorithm Visualization are generally easy to use.	0.796
The instructions on how to use the algorithm learning tool and the Algorithm Visualization are clear.	0.718
The algorithm learning tool and the Algorithm Visualization provide enough user interaction.	0.661
It would be better if there is a “back” button when tracing the algorithm.	0.656
It is better if actual coding or programming is allowed in algorithm learning tool.	0.637
The control buttons to start, stop, and restart the Algorithm Visualization and to run the algorithm step by step are useful for learning the algorithms better.	0.628
It is easy to navigate through the algorithm learning tool and the Algorithm Visualization.	0.599
The displayed changes in values of the variables are useful in learning the algorithm.	0.528
The layout of the algorithm learning tool and the Algorithm Visualization are good.	0.502
The pseudocode display is helpful in better understanding the algorithm.	0.490
The colors of the algorithm learning tool and Algorithm Visualization are pleasing to the eyes.	0.487
It is easy to modify the input values in the Algorithm Visualization.	0.485
The menu that allows selection of the algorithm and speed is helpful.	0.366

Finally, factor analysis was conducted in order to establish the construct validity and to propose an enhanced classification of the items of the evaluation questionnaire on the usability and pedagogical effectiveness of the algorithm learning tool. Using principal components analysis as extraction method and varimax for rotation, three factors that correspond to the three properties proposed for evaluating AVs [12] were extracted. The three extracted factors may be considered in revising the questionnaire. The items that

have low factor loadings (less than 0.6) may be excluded in the revised version of the questionnaire. Table 5 presents the result of the factor analysis and the corresponding factor loadings.

Items that deal mainly with the input menu loaded on the first factor, which may be referred to as the interface assessment factor. This corresponds to the “interactivity” property. All the items based on CALO loaded on the second factor which corresponds to the “didactic structure” property.

This factor may be called the algorithm learning objectives factor. This particular outcome confirms that the use of the CALO taxonomy was suitably incorporated in the design of the questionnaire. The questionnaire items that deal with general characteristics of the Algorithm Visualization and its execution loaded on the third factor which may be called the AV characteristics factor. This factor corresponds to the “symbol system” property for AV evaluation.

7. Summary and Conclusions

An online algorithm learning tool that uses Algorithm Visualization (AV) technology was designed and developed for the students in an introductory computer science course at Tokyo Tech High School of Science and Technology in Japan. The results of the pretest and posttest on algorithms show an increase in the scores of most of the participants. The AV type that offers more input options and control is found to have an effect in raising the scores of the low performing students.

The Categories of Algorithm Learning Objectives (CALO) proposed in a previous study by Lee and Rößling [12] were used as basis for the design of the algorithm test, the evaluation questionnaire, and the Algorithm Visualization itself. After using the learning tool, the students have proven to be capable of performing certain tasks based on the CALO categories, namely, “Descriptive,” identifying algorithms, “Demonstrative,” completing missing lines of codes, and “Decoding,” providing the output of an algorithm simulation.

Considering the responses of the students to the questionnaire for evaluating the usability and pedagogical effectiveness of the algorithm learning tool, there is a collective satisfaction level among the students in using the tool. The student responses indicate that the graphics and animation were appropriate and helpful in understanding the algorithms. The two groups, however, vary in their responses due to the differences in the two AV types provided by the tool. The students who used the AV with limited options and control were satisfied with the instructions and code tracing. They also feel more confident in providing the output of the algorithm simulation. On the other hand, the students who used the AV with more control were satisfied with the navigation and the menu choices, in particular, that of setting the speed of the simulation. These students also feel more confident in describing how the algorithms work after having used the tool.

The result of the factor analysis done on the evaluation questionnaire indicates that its design corresponds to CALO and to the three properties proposed by Lee and Rößling [12] for analyzing and evaluating Algorithm Visualization tools. The three factors obtained are (1) Interface Assessment (“Interactivity”), (2) Algorithm Learning Objectives (“Didactic Structure”) and (3) AV Characteristics (“Symbol System”). These three factors may be used in revising and improving the learning tool’s evaluation questionnaire which is a future plan intended for this study.

Taking into account the results of the initial stage of this study, another implementation of the algorithm learning tool

and the corresponding evaluation instruments is found to be necessary. The lower batches of the Information Systems course will be asked to participate. A grouping scheme, which allows an almost equivalent distribution of students, will have to be done in order to prove if there is indeed a difference in the learning performance between the users of the AV type that allows more input options and control and those who use the AV with limited features. Further validation and analysis of the evaluation questionnaire and the algorithm test will also be done with the objective of determining the usability and pedagogical components specific for learning fundamental algorithms.

The results of the analysis of the other phase of the research, which focuses on the learning motivation of the students, will have to be related with the results presented in this paper. This is in connection with the ultimate goal of the study, which is to propose a model that relates AV design, performance, and motivation of novice learners of introductory computer science.

Appendix

Questionnaire on the usability and pedagogical effectiveness of the algorithm learning tool and the Algorithm Visualization.

- (5) Strongly agree
- (4) Agree
- (3) Not sure
- (2) Disagree
- (1) Strongly disagree

(5) (4) (3) (2) (1)

General Ease of Use

- (1) The algorithm learning tool and the Algorithm Visualization are generally easy to use.
- (2) It is easy to navigate through the algorithm learning tool and the Algorithm Visualization.
- (3) The instructions on how to use the algorithm learning tool and the Algorithm Visualization are clear.
- (4) The colors of the algorithm learning tool and Algorithm Visualization are pleasing to the eyes.

Interface Assessment

- (5) The menu choices for the algorithm learning tool are adequate.
- (6) There is too much text on the pages of the algorithm learning tool.
- (7) The layout of the algorithm learning tool and the Algorithm Visualization is good.

(8) The algorithm learning tool and the Algorithm Visualization provide enough user interaction.

(9) The graphics and animation used are appropriate to visualize the algorithms.

(10) It is easy to modify the input values in the Algorithm Visualization.

(11) It is easy to use control buttons and choice lists in the Algorithm Visualization.

Algorithm Visualization's Characteristics

(12) The Algorithm Visualization allows the user to choose the algorithm to study.

(13) The Algorithm Visualization allows the user to choose the speed of the algorithm animation.

(14) The Algorithm Visualization allows the user to set the size of the array.

(15) The Algorithm Visualization allows step by step tracing of the algorithm.

(16) The Algorithm Visualization allows the user to stop and restart algorithm animation.

(17) The Algorithm Visualization asks questions about the next steps in the algorithm simulation.

(18) The Algorithm Visualization allows the user to assign the elements of the array.

(19) The Algorithm Visualization gives appropriate feedback to the user.

User's Opinion

(20) The menu that allows selection of the algorithm and speed is helpful.

(21) Setting the size and values of the array is helpful in learning the algorithms better.

(22) The control buttons to start, stop, and restart the Algorithm Visualization and to run the algorithm step by step are useful for learning the algorithms better.

(23) The pseudocode display is helpful in better understanding the algorithm.

(24) The algorithm animation is helpful in understanding how the algorithm works.

(25) The displayed changes in values of the variables are useful in learning the algorithm.

(26) It is better if actual coding or programming is allowed in algorithm learning tool.

(27) It would be better if there is a "back" button when tracing the algorithm.

After Using the Algorithm Visualization (Items Based on CALO)

(28) I can now identify the algorithm by just looking at the pseudocode.

(29) I can describe how the algorithms work.

(30) I can demonstrate how the algorithm works using drawing simulations.

(31) I can give the output for a set of data by using algorithm simulation.

(32) I can complete the missing code for all the four algorithms I learned.

(33) I can compare and analyze algorithms that solve the same problems, for example, search and sorting.

(34) I can easily code the algorithms using C programming language or another language I know.

(35) I can now develop my own algorithms to solve other problems.

Conflict of Interests

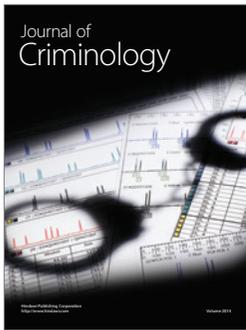
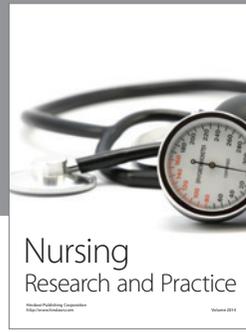
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the administration, teachers, and students of Tokyo Tech High School of Science and Technology for supporting and participating in the study.

References

- [1] ACM, "Chapter 7. Introductory courses," in *ACM Computing Curricula 2001: Computer Science*, pp. 22–34, 2001, <http://www.acm.org/education/curric.vols/cc2001.pdf>.
- [2] C. A. Shaffer, M. L. Cooper, A. J. D. Alon et al., "Algorithm visualization: the state of the field," *ACM Transactions on Computing Education*, vol. 10, no. 3, article 9, 2010.
- [3] P. Saraiya, *Effective Features of Algorithm Visualizations*, Virginia Polytechnic Institute & State University, 2002.
- [4] P. Saraiya, C. A. Shaffer, D. S. McCrickard, and C. North, "Effective features of algorithm visualizations," in *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '04)*, pp. 382–386, ACM Press, March 2004.
- [5] Algoviz portal, 2009, <http://algoviz.org/>.
- [6] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," *Journal of Visual Languages and Computing*, vol. 13, no. 3, pp. 259–290, 2002.
- [7] S. Grissom, M. F. McNally, and T. L. Naps, "Algorithm visualization in CS education: comparing levels of student engagement," in *Proceedings of the ACM 2003 Symposium on Software Visualization (SoftVis '03)*, pp. 87–94, ACM, San Diego, Calif, USA, June 2003.
- [8] C. A. Shaffer, M. Cooper, and S. H. Edwards, "Algorithm visualization: a report on the state of the field," *ACM SIGCSE Bulletin*, vol. 39, no. 1, pp. 150–154, 2007.
- [9] G. Rößling, "A first set of design patterns for algorithm animation," in *Proceedings of the 5th Program Visualization Workshop (PVW '08)*, vol. 224 of *Electronic Notes in Theoretical Computer Science*, pp. 67–76, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 2009.
- [10] J. Urquiza-Fuentes and J. Á. Velázquez-Iturbide, "A survey of successful evaluations of program visualization and algorithm animation systems," *ACM Transactions on Computing Education*, vol. 9, no. 2, pp. 1–21, 2009.
- [11] T. L. Naps, G. Rößling, V. Almstrum et al., "Exploring the role of visualization and engagement in computer science education," in *Proceedings of the Working Group Reports from ITiCSE on Innovation and Technology in Computer Science Education (ITiCSE-WGR '02)*, pp. 131–152, ACM, New York, NY, USA, June 2002.
- [12] M.-H. Lee and G. Rößling, "Integrating categories of algorithm learning objective into algorithm visualization design: a proposal," in *Proceedings of the 15th Innovation and Technology in Computer Science Education Conference (ITiCSE '10)*, pp. 289–293, June 2010.
- [13] P. R. Pintrich and E. V. de Groot, "Motivational and self-regulated learning component of classroom academic performance," *Journal of Educational Psychology*, vol. 82, no. 1, pp. 33–40, 1990.
- [14] J. M. Keller, "First principles of motivation to learn and e3-learning," *Distance Education*, vol. 29, no. 2, pp. 175–185, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

