Abstract

Learning programming languages along with algorithm design and analysis as well as data structures play a vital role among many topics included in the higher education curriculum of studies in the field of Computer Science. Efficient reading, comprehension and prediction of algorithms are fundamental skills that should be acquired by every student of Computer Science prior to the skill of writing own programs. The most popular forms of representation of algorithms include verbal description, pseudo-code, graphical forms, diagrams and a code of a given formal language. The aim of this paper is evaluation of graphical ways of presentation of algorithms and their influence on the efficiency of analyzing of algorithms and, in consequence, on the efficiency of learning programming. The study used eye tracking that enables to obtain visual attention information of subjects tracing algorithms.

Keywords: algorithm comprehension, eye tracking, flowchart, Nassi-Shneiderman diagram.

1 INTRODUCTION

The exchange of visual information is currently a fundamental and commonly used form of communication among young people. For this reason visual information plays an important role in education, as it enables a more effective presentation of educational content.

The main aim of developing of different ways of algorithm representation is their better understanding. Some help is provided by employing a wide range of schemes used for a graphical representation of algorithms and for visualization of their operation e.g. by means of explanatory animations or trace tables. Reading and recording algorithms are fundamental skills that every programmer should possess. The graphical record and visualizations are particularly useful in the case of complicated algorithms, processes and data structures. An important element in their interpretation is faster speed of reading of visual information.

![Figure 1. Forms of presentation of algorithms.](image)
There are many ways to present algorithms. Each of the forms demonstrated in Figure 1 has both advantages and disadvantages and each of them may be useful, depending on the situation. The most popular graphical forms of algorithm representations are flowcharts and Nassi-Shneiderman diagrams. Flowcharts are a widely used visual method for presenting not only algorithms but also different recipes or instructions. Flowcharts are diagrammatic representations filled with text and at the same time a simplified record of a program source code. Nassi-Shneiderman charts constitute replacements for conventional flowcharts in structuring programs [1]. The charts are helpful in nearly every phase of program development starting with early design through walk-throughs, coding, testing, and ending with user education. Nassi-Shneiderman diagrams are usually more organized, more structured, and more intelligible than typical flowcharts. However, there are some problems with this type of diagrammatic representation, as modification of an NS diagram generally entails more work than the equivalent flowchart, the null-else is not evident, in addition, stressing the process one entry-one exit point rule is more difficult than in flowcharts.

The study of the process of learning by means of a technology for recording eye movements when subjects are asked to solve algorithmic problems while looking at different forms of algorithms may be a useful tool to broaden knowledge in the area of cognitive processes.

2 EYE TRACKING TECHNOLOGY

Eye tracking technology is employed for tracking eye movements, directions of gaze and the sequence of reading of visual information as well as identifying the elements that predominantly attract attention of content recipients. The activities are performed by means of eye trackers that is gaze-tracking devices for measuring saccades i.e. quick movements of the eye and fixations - maintaining of the visual gaze on a single location. Fixations are brief periods of relative stability lasting on average about 200-300 ms [2]. During these periods visual information is processed i.e. extracted, decoded, and interpreted. Saccades are fast, jerky, darting movements of the eye between points where fixations occur. However, during saccades visual content is not extracted and visual processing is principally suppressed [3], [4]. Saccades interleaved with fixations create a visual pattern that constitutes a trajectory called a scanpath. It refers to a whole saccade-fixate-saccade sequence and provides information on durations, locations and the order of fixations [5].

Eye trackers are devices used to estimate the direction of gaze. They capture saccades and fixations in the form of XY coordinates of the visual screen making it possible to determine where the gaze was focused in a visual presentation [6]. The majority of such devices available in the market use video image analysis of the center of the pupil and reflections from the cornea as well as infrared light emitters. Every time before the experiment is started, the device must be calibrated for each user.

Eyetracking facilitates identifying of elements separated from the presented picture which are connected with more intense engagement of visual attention, more intense processing of sensory data signaling problems with reading of the visual content or simply indicating increased interest. The analysis of saccadic movements enables tracking of decision-making processes that accompany intentional exploration of the environment aimed at reception of useful visual information conditioning human behavior [7].

Eye trackers are also employed in pupillometry studies. They may provide information on high working memory load on the basis of the increased size of the pupil during performing cognitive tasks [8].

Eye tracking technology may be helpful for identification of behaviour and reactions that subjects taking part in experiments or studies are not able or not willing to discuss. As regards algorithmics and programming, the technology may be successfully employed for observation and separating of strategies for reading information as well as for patterns of information processing.

3 INFLUENCE OF FORM OF ALGORITHM REPRESENTATION ON ALGORITHM COMPREHENSION

Issues related to reading, comprehension and development of algorithms have been of interest to researchers for many years.

One of conducted studies was an empirical study on the eye movement patterns that constitute the order in which keywords or some sections of keywords are read. The study involved summarization tasks during reading a source code. Programmers who participated in the study had to follow nearly
identical eye movement patterns. The patterns proved to be analogous to eye movement patterns of reading a natural language [9].

There were also experiments investigating differences in comprehensibility of textual and graphical notations for representing decision statements. The studies demonstrated that programmers find textual notations better than particular graphical notations. It results from the great amount of experience with textual notations in programming languages that programmers have and are exposed to [10].

In another study, Shneiderman and his colleagues investigated the utility of detailed flowcharts in composing, comprehending, debugging, and modifying programs. They found no statistically significant differences between the performance of subjects who used flowcharts in comparison with those who did not [1].

As far as the last of the above findings are concerned, opinions of researchers are divided. Scalan reached different conclusions with regard to the utility of structured flowcharts to aid program comprehension. He found a statistically significant advantage in using flowcharts in contrast to pseudo-code for comprehension of complicated algorithms. Scalan also measured comprehension by means of the response time [11].

Eye tracking technology was employed to study differences in comprehension of algorithms represented in the form of flowcharts as well as pseudo-code. They proved that structured flowcharts do aid algorithm comprehension. A large difference was discovered even for the simplest algorithm [11].

4 METODOLOGY

4.1 Procedure

The study conducted for the needs of this paper comprised two parts: a survey study i.e. a questionnaire and an experiment using eye tracking technology. The former pertained to aspects involving knowledge and use of different ways of algorithm representation and focused in particular on two schemes, i.e. a flowchart and N-S diagrams. The latter, that is the experimental part, consisted of two stages. The first one was devoted to assessment of how well the participants of the study comprehended the presented algorithms. The assessment was made on the basis of the results of algorithmic calculations given by the subjects. The second stage of the experiment used eye tracking technology which measured visual activity of the subjects. Tracking the gaze trajectory of participants of the study, while the subjects analyzed diagrams of particular algorithms, facilitated discovering strategies used by the subjects as well as identification of mistakes that had been made.

The questionnaire addressed the following issues:

1. use of different ways of algorithm representation during performing programming tasks: textual description, sequence of steps, pseudo-code, a flowchart or an N-S diagram,
2. the level of knowledge with regard to N-S diagrams as well as flowcharts,
3. used sources of information while learning about N-S diagrams as well as flowcharts,
4. need for using flowcharts as well as N-S diagrams in programming practice,
5. preference for using a graphical representation of algorithms (a flowchart or an N-S diagram),
6. advantages and disadvantages of using the above algorithm schemes.

After responding to the questions in the questionnaire, the subjects participated in the experiment conducted by means of the eye tracker. The experiment was conducted according to a scenario in which the participants were presented 10 slides containing 5 algorithm schemes of different level of complexity. Each of the slides was in 2 versions: both in the form of a flowchart and in the form of an N-S diagram. The slides with the schemes were presented to participants in random order. The subjects were supposed to analyze each of the algorithms and express its result in a verbal form. Algorithms selected for the experiment covered the following assignments:

- A1 – checking if the given year is a leap year (using a nested conditional statement)
- A2 – calculating the sum of the series (using a do-while loop statement)
- A3 – generating 6 successive Fibonacci words (using compound statements)
A4 – algorithm for finding the greatest common divisor (GCD) of two integers
A5 – sum of selected elements of a 2D matrix (using nested loops)

After conducting the experiment, 12 recordings were made. They were later analyzed by means of Tobii software. First, the recordings were analyzed with regard to comprehension of algorithms. The analyses involved the results given by the participants of the study. In further studies, however, it would be also valuable to investigate the time devoted to particular tasks as well as the number of the attempts made.

Finally, quantity analyses of eye tracking recordings were performed, which involved analyzing errors made by the subjects as well as the strategies used during analyzing of studied algorithms.

4.2 Apparatus

The study used a stationary eye tracker - Tobii TX300. The device provides non-contact interaction between the user and the computer. Tobii TX300 is integrated with the 23” high-resolution TFT panel. The system works in the range of infrared radiation (IR) and has a high precision in determining a fixation point on the screen. The spatial resolution of the system is 0.4 degrees. It means that for a user sitting 60 cm away from the monitor, the tracking gaze system determines the coordinates of fixation points with precision of up to 4.2 mm (horizontally and vertically). The remaining technical parameters of the device are as follows:

- time resolution: 300 Hz,
- diagonal and screen proportions: 23", 16:9,
- maximum screen resolution: 1920x1080.

The eye tracker is equipped with Tobii Studio software. It is a tool for recording and analyzing of the eyeball movement. Additionally, it easily facilitates to compare, interpret and present data. The software enables to conduct quality and quantity analyses. During the study subjects may be presented materials in different forms for analysis: graphics, movies, www sites etc. The software offers a possibility of randomization of presented materials, which eliminates the influence of the order of presentation on the results of the study [12].

4.3 Participants

The experiment involved 12 participants aged 20-25 years. They were all students of Computer Science of the Lublin University of Technology; some of them in the second year of first-cycle studies and some in the third year of second-cycle studies. Each of the subjects completed courses where different forms of algorithm representation, including N-S diagrams and flowcharts, were studied.

During the experiment the participants were seated in a quiet usability laboratory in ordinary office chairs near the experimenter and they were facing a 23” TFT display. The intensity of light in the laboratory where the study was conducted was the same for each of the subjects.

Before the study began, each participant was asked to complete a short questionnaire concerning the content of the study. Afterwards the subjects were informed about the purpose and the range of the study. They were also acquainted with eye tracking technology. Before taking part in the experiment, the subjects were instructed to comprehend the algorithms as well as possible. Afterwards each of the participants had to pass an automatic eye-tracking calibration. The calibration was followed by the main part of the study in which the subjects were presented algorithm schemes and after comprehending them, the participants were asked to give the obtained results.

5 RESULTS

5.1 Questionnaires

The purpose of this part of the study was a diagnosis of usefulness of different forms of algorithm representation for respondents, i.e. Computer Science students.

The conducted questionnaire proved that most respondents find prose or textual algorithm representation to be most useful (Figure 2). In the case of graphical algorithm representation
flowcharts proved to be more popular with students (35.7% respondents) than N-S diagrams (21.4% respondents).

The questioned subjects were also asked to rate the level of their skills in using of both types of schemes (on a scale of 0 to 3). The average for all respondents with regard to N-S diagrams was 1.7 whereas with regard to flowcharts it was 1.8. The main source of information on using the schemes was classes at the university (83.3%). However, lessons at schools (15.5%) as well as the Internet (12.7%) were also mentioned. Most respondents prefer flowcharts (69.2%) to N-S diagrams (30.8%). Nearly all questioned subjects emphasized time consumption as a disadvantage of development of algorithms in their graphical form.

The conducted questionnaire showed that respondents prefer the textual form of algorithms; however, they also see a need for using schemes. Flowcharts proved to be predominantly popular.

### 5.2 Study of algorithm comprehension

The participants of the experiment had to read successive algorithms presented for them in the form of two types of schemes. They were also asked to give obtained results. Giving the correct result meant comprehension of the algorithm. Figure 3 represents results of all participants showing also the level of difficulty of particular algorithms as well as differences in their comprehension with regard to different types of schemes.

![Figure 3. Results of all participants showing the level of difficulty of particular algorithms as well as differences in their comprehension with regard to different types of schemes.](image-url)
Respondents of the study had considerable problems with algorithms containing loops and demanding remembering of a few values: the Fibonacci sequence, a sum of a series of numbers and a sum of elements in a 2D matrix. The average result of correct answers for the whole questioned group was 55%. Taking into consideration the type of scheme, the level of accurate answers was similar: flowcharts - 56.7% and N-S diagrams - 53.3%.

5.3 Results of the eye tracking analysis

As a result of the performed experiment a big amount of research material was collected. Below there are examples in the form of heatmaps and scanpaths which show strategies used by participants of the experiment solving particular problems.

Due to the nature of problems connected with the algorithm analysis, obtaining the correct result was determined by performing a suitable succession of series of operations, which in this case depended on the appropriate interpretation of expressions recorded in block schemes.

Figure 4 presents an example of a heatmap which demonstrates the used strategy of analyzing of an algorithm when the subject gave an incorrect answer. A large concentration of attention represented by the red and yellow spots is visible in locations where the subject's gaze should be focused for a relatively short time. It occurs during an operation when the next variable assigns the value of the counter variable when the result of the condition statement is true (Y). A large concentration of attention on simple assignment statements in this case indicates a high cognitive load resulting from the need of a simultaneous calculation and memorizing values of 3 variables. This example posed the biggest difficulty for the subjects.
Figure 5. Record of the analysis of an algorithm calculating the Fibonacci sequence (scanpath).

Figure 5 shows a record of a part of the process of solving a simple problem – an algorithm using a nested conditional statement. It is called a scanpath representing fixations in the form of circles with different diameters proportional to the duration of maintaining of the visual gaze on a given location and a path of saccades visible as straight lines between fixations.

In this case, during the analysis the subject makes an error choosing the wrong flow line in the flowchart. The mistake stems from the wrong calculation of the condition statement in the first decision block. This example indicates that on the basis of tracking the scanpath, it is possible to discover errors that were made by students participating in the experiment.

Figure 6. Record of the analysis of an algorithm calculating the sum of the series (heatmap).

Another example demonstrated in Figure 6 presents an inappropriate strategy adopted by a subject analyzing the algorithm. In this case the participant of the study focused attention primarily on the block realizing the loop statement, containing a conditional statement and 3 simple math expressions. The subtraction expression attracted the most attention, as the operation was one of the most commonly made in this algorithm.
Figure 7. Record of the analysis of an algorithm summing selected elements of a 2D matrix (heatmap).

The last of the examples shown in Figure 7 demonstrates a choice of a good strategy of the analysis of an algorithm summing numbers in a 2D matrix. Here most attention was devoted to analyzing elements of the nested loop i.e. the termination condition of the loop, the internal condition in the loop, and the expression of summation of numbers and the expression of increment.

6 CONCLUSIONS

Although the graphical representation of algorithms was created in order to facilitate and accelerate comprehension of algorithms, to reduce the number of errors in reasoning and to foster confidence of science specialists in reading algorithms, currently many IT programmers tend to avoid using this form of representation of algorithms. Even at university level, which may be observed in the presented study, the most popular form of successful reading of algorithms is the textual record, especially with regard to the source code of a given language.

If using diagrams is limited only to classes at schools or at universities IT graduates and later also computer specialists are likely to have problems with efficient reading and comprehension of schemes included in specialist documentation. The results of the study shown in this paper prove that subjects had difficulty in dealing with complex algorithms presented in a graphical form. It indicates the need for more intensive practice of skills in this field.

Thanks to constant progress in technology and science the nature of cognitive processes is studied in many different ways. The eye tracking study presented in this paper focused on analyzing and comprehension of graphical forms of algorithms; however, eye tracking technology may be helpful in assessing of the level of visual competencies in other areas of education.

REFERENCES


