A Taxonomy for Animation-aided Visualization Tools

by

Aishwarya Majumder

A Creative Component Report submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Dr. Simanta Mitra, Co-major Professor
Dr. Gurpur Prabhu, Co-major Professor
Dr. Carl K. Chang

The student author, whose presentation of the scholarship herein was approved by the program of study committee is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

DEDICATION

To my parents, friends, family and teachers for their constant support and guidance throughout my life.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AV applications – Animation-aided visualization applications
PV – Program visualization
Pseudo code – In this report "Pseudo code" refers to an outline of steps for a concept or process or algorithm.

ACKNOWLEDGEMENTS

I would like to express my immense gratitude to Dr. Simanta Mitra for his continuous guidance and support throughout the course of this research. I would also like to extend heart-felt appreciation to my co-major professor Dr. Gurpur Prabhu for his regular constructive feedback and for being a great mentor. I want to thank my committee member Dr. Carl Chang for his help and support.

I would like to especially thank the Iowa State University Computer Science departmental faculty and staff for making this journey a smooth and memorable one.

In addition, I am grateful to my friends in India and in the US, my teachers from kindergarten to Graduate school, for their constant support and counsel.

Finally, I want to express sincere gratitude to my family and best friend for their boundless love and for making me who I am today.

MOTIVATION

Visualization is the usage of various techniques to aid human understanding of a concept (software or hardware), algorithm, program or process. Visualization tools are the go-to solution for teachers to effectively explain algorithms and hardware/software concepts. Multiple studies have conclusively proved that students learn and grasp concept(s) better when they have access to a visualization application which demonstrates the concept(s).

Animation acts as a key tool to enhance the effectiveness of a visualization application. Apart from animations, if these systems provide interactions for users, it increases the general educational effectiveness. For example, in case of an algorithm animation system, user ability to forward or rewind through the animation is very useful. These types of interactions help increase clarity and promote better understanding of the algorithm for learners, thereby enhancing the overall efficacy of the application.

The primary motivation behind my work, is to identify features that contribute to making a program visualization tool educationally effective. I have focused on animation-aided visualization applications. A taxonomy based on such attributes would be useful in more than one way. Firstly, it will summarize in a concise manner the multitude of features available in current animation-aided visualization (AV) applications. Secondly, the taxonomy can be used as a tool for comparing different AV applications with each other and understanding their strengths and limitations.

Another motivation has been to study the evolution of Program and Algorithm visualization tools and related research over the years.

ABSTRACT

Program visualization has been highly effective in pedagogical context. This has led to rapid increase in development of PV tools in the last two decades, with focus on enhancing the teaching/learning experience. What are the features that should be kept in mind while developing a visualization tool or choosing one that increases educational effectiveness? This project is an effort to establish a taxonomy that can be used as a means for evaluating visualization tools and understanding their strengths and limitations. The taxonomy can be used as a tool to compare visualization systems with each other as well as for designing and implementing new systems and features. The project is based on a study of animation-aided visualization applications and a resultant list of attributes which are desirable and increase the effectiveness of such applications. The overall efficacy of these applications will increase when educators can easily and effectively use visual representations while teaching, thus enabling learners to have better understanding and clarity about the concept. The project also concisely captures the evolution and progress of research in the field of program visualization and development of PV tools, over the years.

INTRODUCTION

Visualization systems are becoming increasingly important as an aid for teachers to effectively explain concepts (hardware or software). Educators can use visualizations to help students understand and learn new concepts. Animation acts as a key tool to enhance the potency of visualization applications. There are quite a few such applications available which can be used by educators to explain concepts. Animal Algorithm Animation system, VisuAlgo, Explain Git with D3, Loupe for JavaScript, Alice 2, jGRASP are examples of such applications. These applications provide animated visualizations for specific concepts. There are certain attributes that, if present in such applications, will increase their usefulness for educators and learners. For example, in case of an algorithm animation system, the ability of the application to highlight the current step of pseudo code being depicted by the animation concurrently on screen. Another feature, if present in such an application, that would be extremely beneficial for teachers – the ability of the application to provide educators with the option to create visual representations and animations using an easy to use language. If the teacher needs some specific type of interaction between the user and the animation – this feature would help serve that purpose without too much time or effort being involved. This project describes more such attributes which add to the overall effectiveness of animation-aided visualization applications. The attributes have been identified after studying some specific applications from two points of view – 1) that of student who aims to learn a concept 2) that of an educator who aims to effectively explain a concept.

The work in this project has been divided into two sections:

Chapter I – This section is dedicated to the identification of desirable features and creation of a taxonomy. The taxonomy will serve as a means to evaluate and compare AV tools. This section contains the taxonomic evaluations of some specific AV tools.

Chapter II – This section is dedicated to representing how research in the field of program visualization has progressed and evolved through the years. The development of various PV tools over the last three decades has also been captured in a timeline-based representation.

CHAPTER I

SURVEY AND TAXONOMY

1.1 AV APPLICATIONS STUDIED

Five Animation-aided visualization (AV) applications were studied. The key criteria that was used in choosing the tools was their availability - easily available tools which used animation techniques to visualize a program/concept/algorithm.

A. VisuAlgo

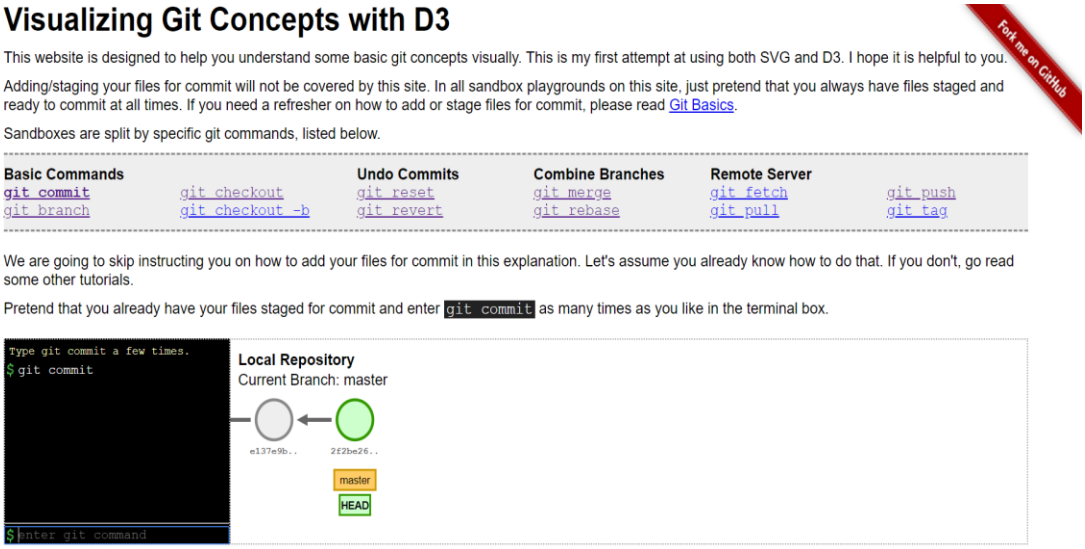
VisuAlgo is a web-based algorithm visualization tool without the need to install any additional software. It uses the latest web technology: HTML5, CSS3, JavaScript. The tool allows users to specify their own algorithm inputs and the visualization will work with those inputs. The application offers visualizations for various algorithms. It is a collection of algorithm visualizations with unified interface. As of 2015, this application recorded 2000 sessions daily from worldwide visitors; it has grown more popular since then.

url: https://visualgo.net/en

B. Explain git with D3

This is a web-based application which can be used to understand basic git concepts visually. This tool can be used without installing any additional software. SVG and D3 were used to develop the animations.

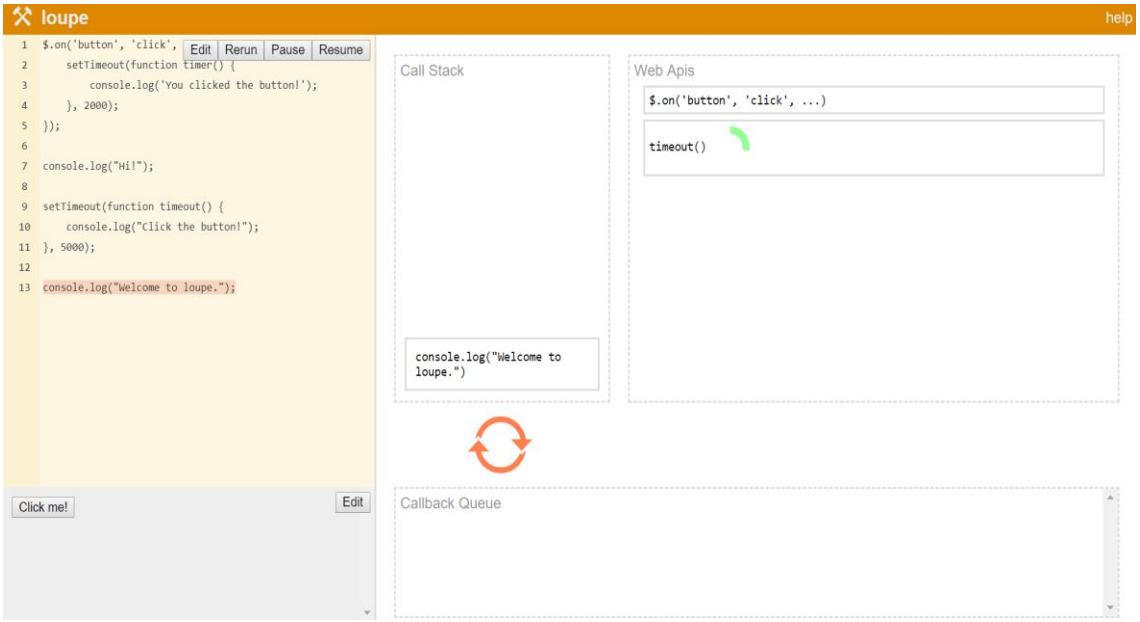
url: https://onlywei.github.io/explain-git-with-d3/



C. Loupe

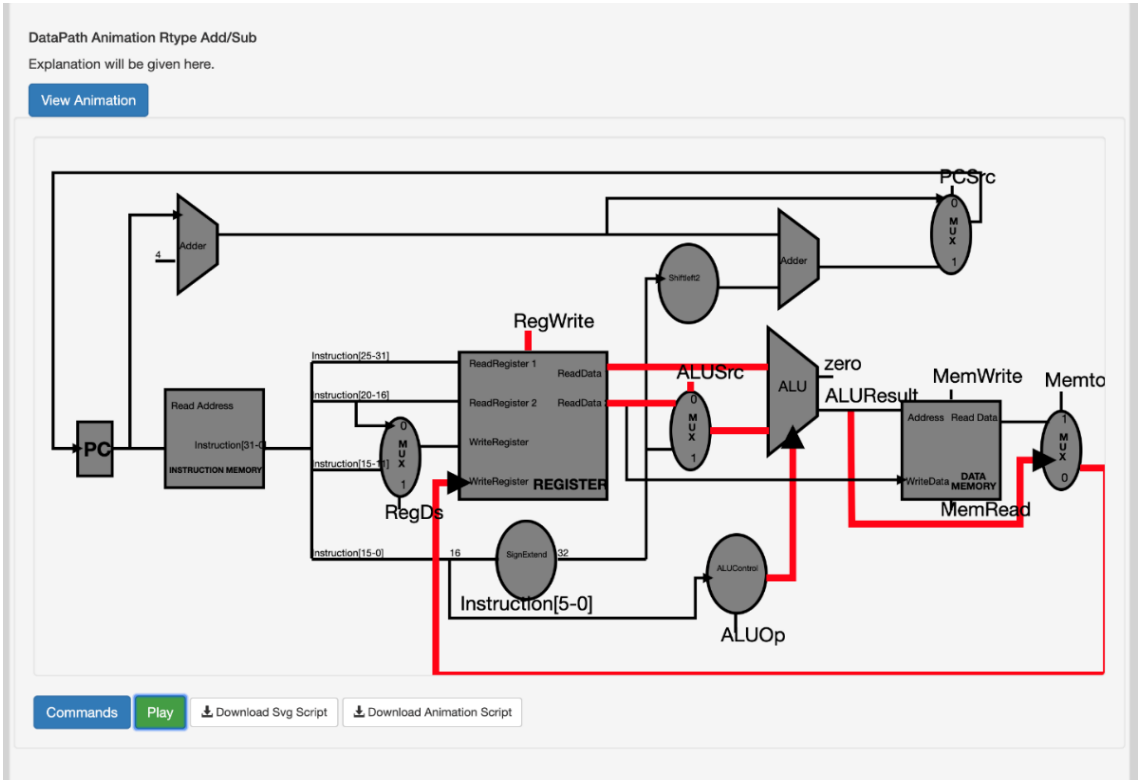
Loupe is a web-based application which runs entirely in the browser. It is visualization to help students understand how JavaScript's call stack/event loop/callback queue interact with each other. Loupe uses JavaScript, ExtendScript, CSS to create the visualizations.

url: http://tinyurl.com/ncefteb



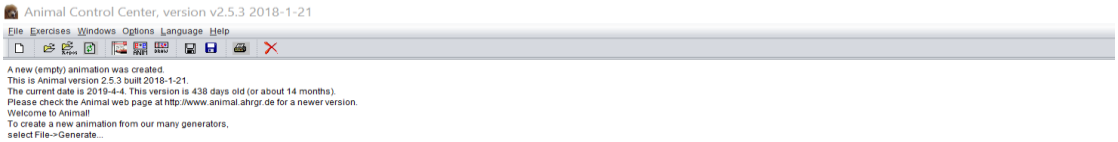
D. Datapath Animation

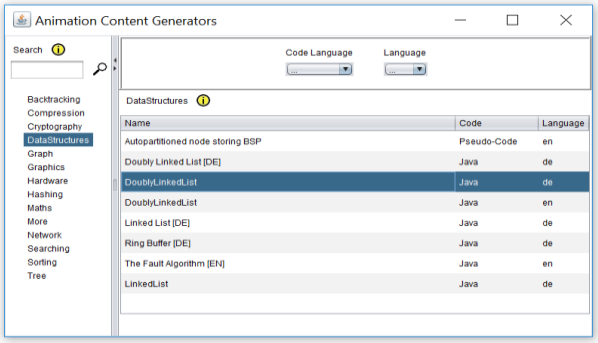
This is a general-purpose animation tool developed by a fellow student for his creative component, Spring 2019 at Iowa State University. The tool helps to visualize the workings of an R Type Adder/Subtractor, Branching and Load word. This application can also be used to create animations. Javascript and Jquery were used to develop the tool.

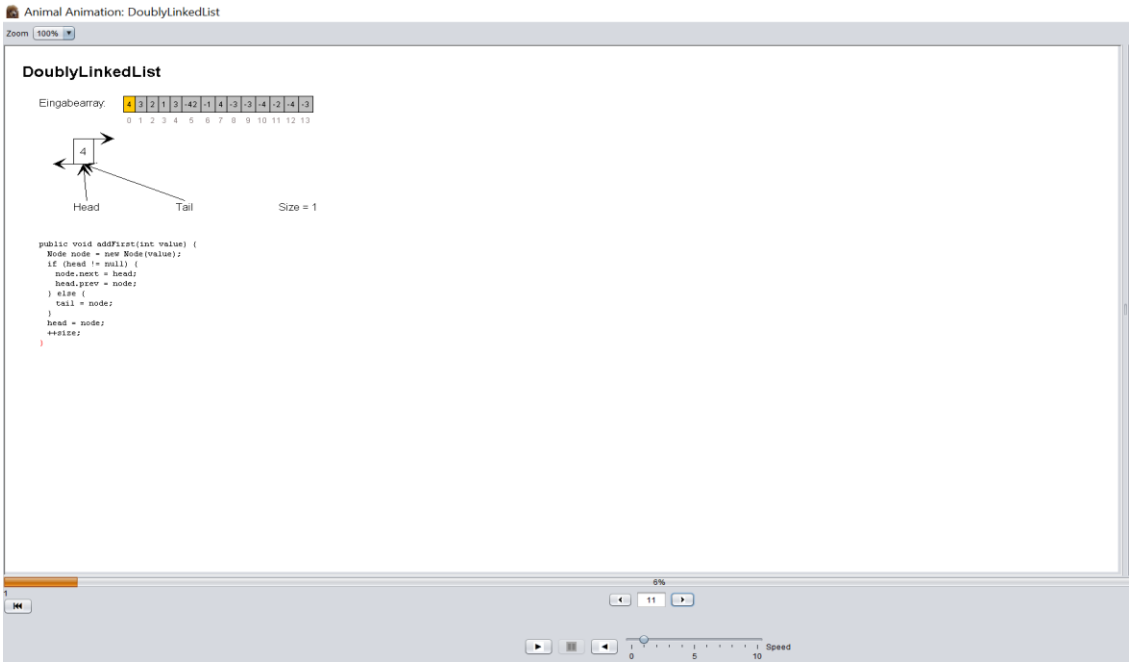


E. ANIMAL Algorithm Animation System

ANIMAL is a general-purpose animation tool with a current focus on algorithm animation. This application offers animated visualizations for a varied range of algorithms. ANIMAL is similar to VisuAlgo in the respect that both the tools offer animated visualizations for numerous algorithms. ANIMAL can also be used for developing animations to be used in lectures. ANIMAL is not a web-based application; one must download and install the application to be able to use it. The software is freely available for download at http://www.algoanim.net/.







1.2 ATTRIBUTE IDENTIFICATION

The above-listed AV applications has been studied from two points of view – 1) that of student who aims to learn a concept 2) that of an educator who wants to effectively explain a concept.

The study was aimed to identify features which make applications such as the above-mentioned, more educationally effective and pedagogically relevant. The feature that is most likely to enhance effectiveness of such applications is user interaction with the animation. More than passive animation, algorithm visualizations must require users to interact with the animation, for these tools to be educationally effectual.

Below is a list of identified attributes that are preferable to have in AV applications and which enhance the overall educational efficacy of such systems. The term "pseudo code" in all occurrences in this report, refers to outline of steps for a concept or process/program or algorithm.

Table 1
Identified features

1. Application ability to show pseudo code on screen concurrently with the animation.
2. Ability of application to highlight step of pseudo code currently being depicted by animation and/or explain in words concurrently on the screen.
3. Ability of application to provide option on screen to explore summary of concept and/or urls/multimedia to know in detail about depicted concept.
4. User ability to control speed of animation(fast/slow/medium). User may choose to view animation at default speed.
5. User ability to pause and resume animation.
6. User ability to forward and rewind through animation.
7. User ability to rerun animation.
8. Ability of application to show animation, step by step, controlled by user click. User may choose to see default animation in batch mode.
9. Ability of application to take user data as input, where applicable and show animation with the corresponding user input. User may choose to view animation with application-provided default data.
10. Application ability to provide more than one representation of input/output data for depicted concept, if at all multiple input/output data representations is possible.
11. Ability of application to provide users with the option to create visual representations and animations using an easy-to-use language. The users who create the animations using the application will potentially be teachers attempting to explain a concept to students.

1.3 TAXONOMY

Using the study of the applications mentioned in the previous section and the identified features as a background, I will now define a taxonomy of features for AV applications. This taxonomy can be used to compare and evaluate AV tools, understand their strengths/shortcomings and identify features that are desirable in such applications for them to be effective.

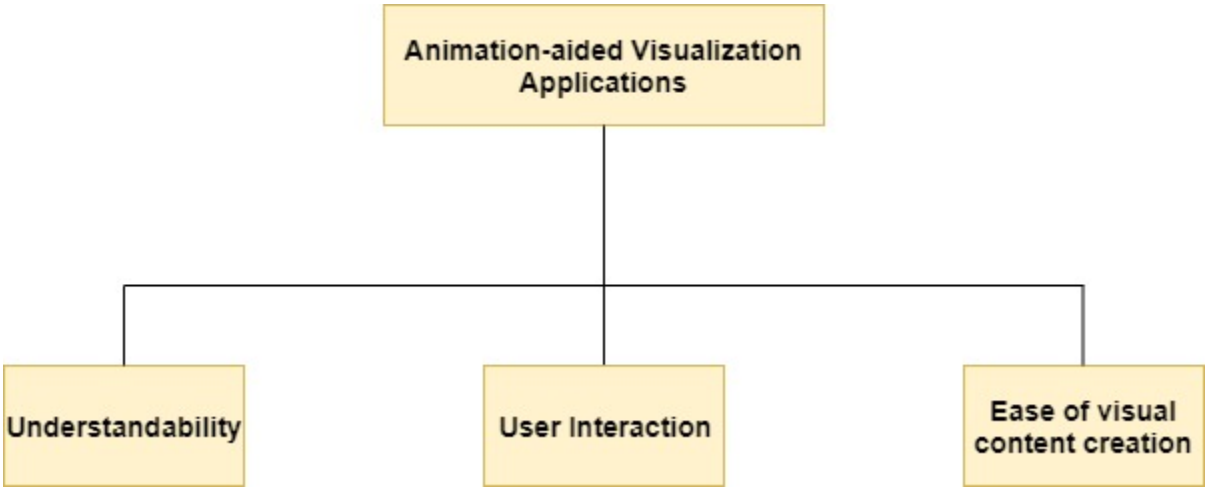


Figure 1. Top level of taxonomy

At the top level of the taxonomy, the categories are:

A. **Understandability**: Understandability considers the ease of understanding provided by the application for the users. For example, features like having the outline of steps for the concept/process on screen concurrently with the animation and highlighting step(s) currently being animated. These features make it easier for users to understand the concept being depicted.

B. **User Interaction:** The category User Interaction describes the level and type of interaction and control provided for the end user of animations.

C. **Ease of visual content creation:** This category depicts whether or not the application supports user creation of visual representations and animations using an easy-to-use language.

1.3.1 Category A: Understandability

Understandability depicts the ease of understanding the application provides for the users.

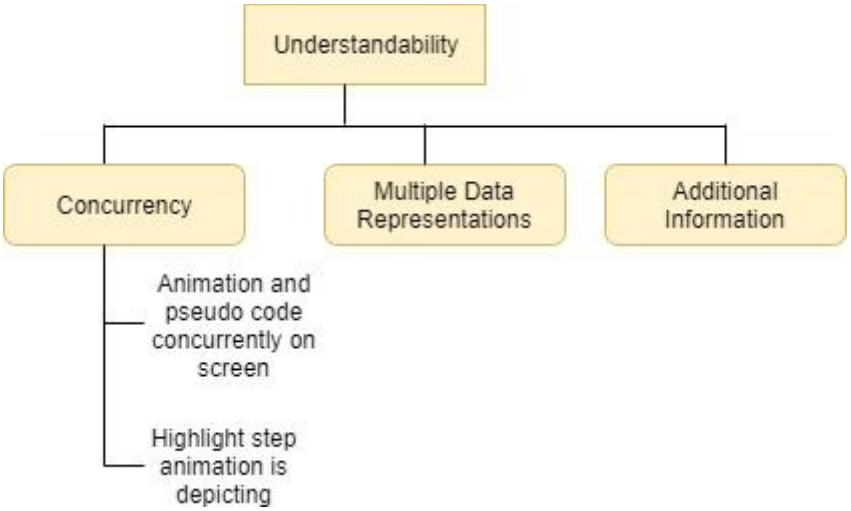


Figure 2. Category Understandability

A1. **Concurrency** considers the concurrent actions being shown on screen along with the animation, which aids understanding of the concept being visualized.

A1.1 Animation and pseudo code concurrently on screen – This denotes application ability to show pseudo code on screen concurrently with the animation. User can choose to enable or disable this feature.

A1.2 Highlight step animation is depicting – This denotes ability of application to highlight step of pseudo code currently being depicted by animation and/or explain in words concurrently on the screen. User can choose to enable or disable this feature.

Evaluation: Table 2 represents the evaluation of the five studied AV applications in the Concurrency category.

Table 2
Evaluation of applications in category Concurrency

Attribute Classification	Attributes	Visu Algo	Explain Git with D3	Loupe for JS	Datapath animation	Animal
A1. Concurrency	**A1.1** - show pseudo code on screen concurrently	**True**	**NA**	**True**	**True**	**True**
	A1.2 - highlight step of pseudo code currently being depicted	**True**	**NA**	**True**	**True**	**True**

A2. Multiple Data Representations considers application ability to provide more than one representation of input/output data for depicted concept, if at all multiple input/output data representations is possible.

Evaluation: Table 3 represents the evaluation of the applications in category Multiple Data Representations.

Table 3
Evaluation of applications in category Multiple Data Representations

Attribute Classification	Attribute	Visu Algo	Explain Git with D3	Loupe for JS	Datapath animation	Animal
A2. Multiple Data Representations	**A2**- provide more than one representation of input/output data for the concept, if at all multiple data representations is possible	**False**	**NA**	**NA**	**NA**	**False**

A3. Additional Information denotes ability of application to provide option on screen to explore summary of concept and/or urls or multimedia like sound clip, video etc. to know in detail about depicted concept.

Evaluation: Table 4 represents the evaluation of the applications in category Multiple Data Representations.

Table 4
Evaluation of applications in category Additional Information

Attribute Classification	**Attribute**	**Visu Algo**	**Explain Git with D3**	**Loupe for JS**	**Datapath animation**	**Animal**
A3. Additional information	**A3**- Ability of application to provide option on screen to explore summary of concept and/or url/multimedia to know in detail about depicted concept.	**True**	**True**	**True**	**False**	**True**

1.3.2 Category B: User Interaction

The category User Interaction describes how the application supports interaction between the user and the animation.

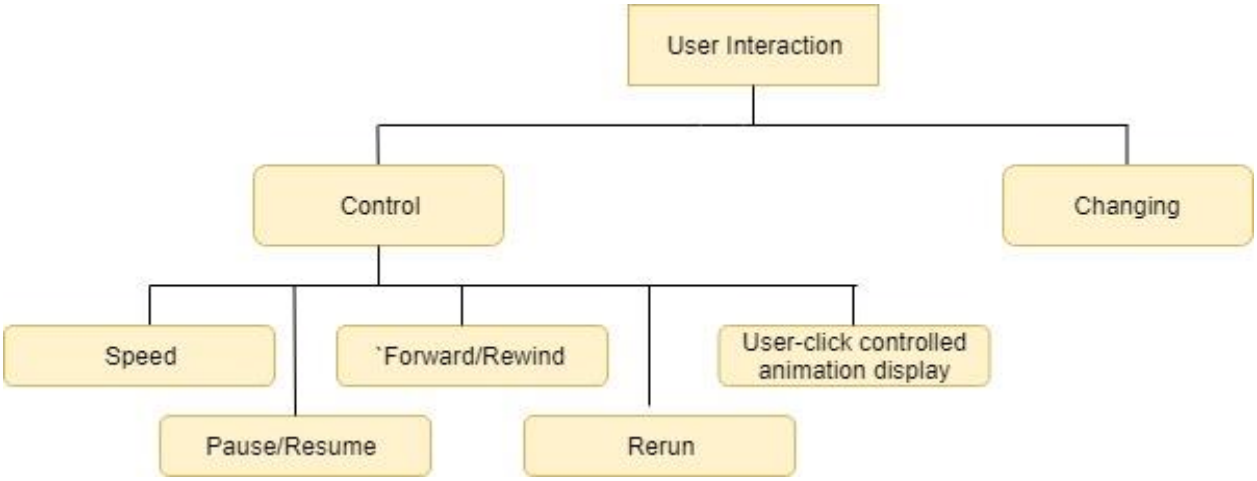


Figure 3. Category User Interaction

B1. Control denotes the application ability to support user interaction which controls the way the animation is executed. Such features include, for instance, pausing the animation, and browsing it stepwise or continuously.

B1.1 Speed - User ability to control speed of animation(fast/slow/medium). User may choose to view animation at default speed.

B1.2 Pause/Resume - User ability to pause and resume animation.

B1.3 Forward/Rewind - User ability to forward and rewind through animation.

B1.4 Rerun - User ability to rerun animation.

B1.5 Batch mode or user click controlled - Ability of application to show animation, step by step, controlled by user click. User may choose to see default animation in batch mode.

Evaluation: Table 5 represents the evaluation of the applications in category Control.

Table 5
Evaluation of applications in category Control

Attribute Classification	Attributes	Visu Algo	Explain Git with D3	Loupe for JS	Datapath animation	Animal
B1. **Control**	**B1.1**- control speed of animation(fast/slow/medium)	**True**	**False**	**False**	**False**	**True**
	B1.2 - pause and resume animation	**True**	**False**	**True**	**True**	**True**
	B1.3 - forward and rewind through animation	**True**	**False**	**False**	**True**	**True**
	B1.4 - rerun animation	**True**	**True**	**True**	**True**	**True**
	B1.5 - show animation, step by step, controlled by user click	**True**	**False**	**False**	**True**	**True**

B2. Changing denotes ability of application to take user data as input, where applicable and show animation with the corresponding user input. User may choose to view animation with application-provided default data.

Evaluation: Table 6 represents the evaluation of the applications in category Changing.

Table 6
Evaluation of applications in category Changing

Attribute Classification	Attribute	Visu Algo	Explain Git with D3	Loupe for JS	Datapath animation	Animal
B2. **Changing**	**B2**- application ability to take user data as input, where applicable and change animation according to the corresponding user input	**True**	**True**	**True**	**True**	**False**

1.3.3 Category C: Ease of visual content creation

This category considers ability of application to provide users with the option to create visual representations and animations using an easy-to-use language. The users who create the animations using the application will potentially be teachers attempting to explain a concept to students.

Evaluation: Table 7 represents the evaluation of the applications in category Ease of visual content creation.

Table 7
Evaluation of applications in category Ease of visual content creation

Attribute Classification	**Attribute**	**Visu Algo**	**Explain Git with D3**	**Loupe for JS**	**Datapath animation**	**Animal**
C. Ease of visual content creation	**C**- provide users with the option to create visual representations and animations	**False**	**False**	**False**	**True**	**True**

1.4 SUMMARY OF ATTRIBUTES

Table 8 represents a consolidated form of all the identified attributes and their classification with respect to the taxonomy.

Table 8
Identified attributes and their classification with respect to the taxonomy

Attribute Classification		Attribute Description
A. Understandability	**A1. Concurrency**	**A1.1 Animation and pseudo code concurrently on screen -** Application ability to show pseudo code on screen concurrently with the animation.
		A1.2 Highlight step animation is depicting - Ability of application to highlight step of pseudo code currently being depicted by animation and/or explain in words concurrently on the screen.
	A2. Multiple Data Representations	**A2 -** Application ability to provide more than one representation of input/output data for depicted concept, if at all multiple input/output data representations is possible.
	A3. Additional Information	**A3 -** Ability of application to provide option on screen to explore summary of concept and/or urls/multimedia to know in detail about depicted concept.
B. User Interaction	**B1. Control**	**B1.1 Speed -** User ability to control speed of animation(fast/slow/medium). User may choose to view animation at default speed.
		B1.2 Pause/Resume - User ability to pause and resume animation.
		B1.3 Forward/Rewind - User ability to forward and rewind through animation.
		B1.4 Rerun - User ability to rerun animation.
		B1.5 User-click controlled animation display - Ability of application to show animation, step by step, controlled by user click. User may choose to see default animation in batch mode.
	B2. Changing	**B2 -** Ability of application to take user data as input, where applicable and show animation with the corresponding user input. User may choose to view animation with application-provided default data.
C. Ease of visual content creation		**C -** Ability of application to provide users with the option to create visual representations and animations using an easy-to-use language. The users who create the animations using the application will potentially be teachers attempting to explain a concept to students.

1.5 TAXONOMIC EVALUATION OF APPLICATIONS

Table 9 presents a consolidated version of evaluation of the five applications in all categories of the taxonomy.

Table 9
Evaluation of Applications in all categories of the taxonomy (consolidated)

Attributes			VisuAlgo	Explain Git with D3	Loupe for JS	Datapath animation	Animal
A. Understandability	A1. Concurrency	A1.1 Animation and pseudo code concurrently on screen	True	NA	True	True	True
		A1.2 Highlight step animation is depicting	True	NA	True	True	True
	A2. Multiple Data Representations		False	NA	NA	NA	False
	A3. Additional Information		True	True	True	False	True
B. User Interaction	B1. Control	B1.1 Speed	True	False	False	False	True
		B1.2 Pause/ Resume	True	False	True	True	True
		B1.3 Forward/ Rewind	True	False	False	True	True
		B1.4 Rerun	True	True	True	True	True
		B1.5 User-click controlled animation display	True	False	False	True	True
	B2. Changing		True	True	True	True	False
C. Ease of visual content creation			False	False	False	True	True

1.6 COMPLETE TAXONOMY

Figure 4 represents the complete taxonomy described in this project.

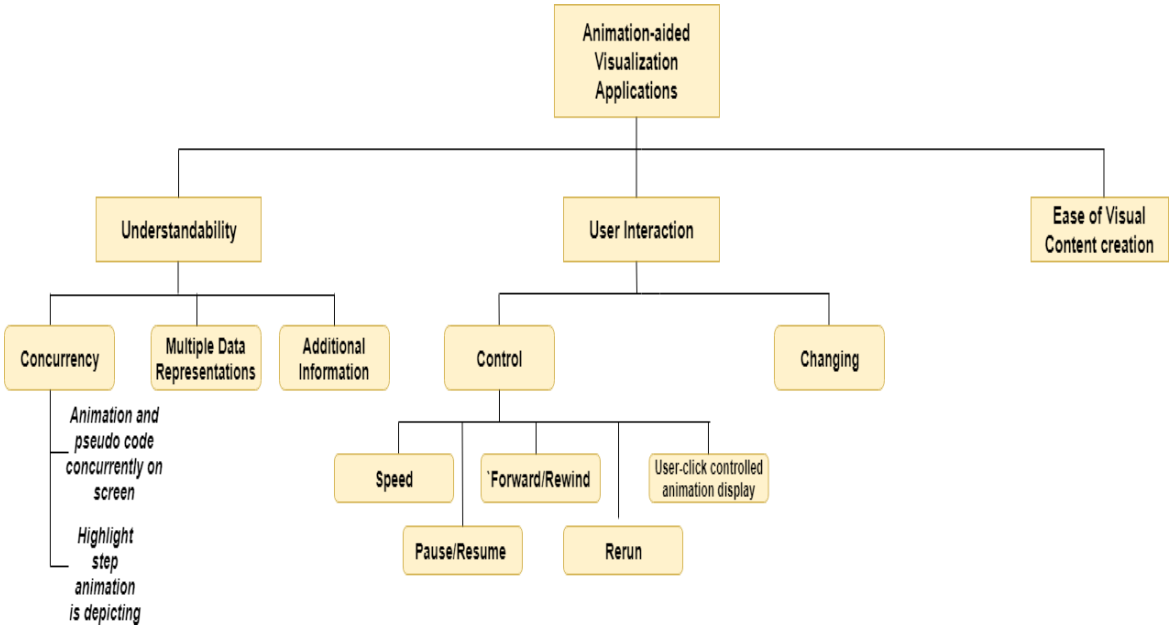


Figure 4. Complete Taxonomy

CHAPTER II

Evolution of Research in the field of Program Visualization

2.1 Progress of research in the field of animation-aided Program visualization through the years (1980s – 2017)

The need for visualization as a means to effectively understand algorithms or some piece of data or even a program or concept was being realized in the early 1980s.

Numerous papers and articles were published which stressed on how allowing a user to interact with dynamically changing visual representations of algorithms or a concept or process/program may help in teaching, research, or systems programming. In the early 1990s, researchers published ideas of visualization systems and algorithm animation systems like Tango, Zeus were created.

From the mid-1990s, extensive research was being conducted on program and algorithm visualization in educational context. Studies researching the need of visualizations in learning and education were carried out. The studies established that visualization helped students learn faster and educators could more effectively explain concepts.

In 2000, the ANIMAL algorithm animation system was created. This tool was specifically designed for educational purposes. The application has the feature for developing animations to be used in lectures.

Since the 2000s, more work has been conducted on program and algorithm visualizations with respect to education and pedagogy. Studies have been carried out to evaluate the impact of program visualizations on education and learning. And they have established that visualization tools are the need of the day in Computer Science education.

2.1.1 Chronological representation of research in program visualization

Figure 5 depicts a chronological representation of research in the field of animation-aided visualization through the years 1985-2017.

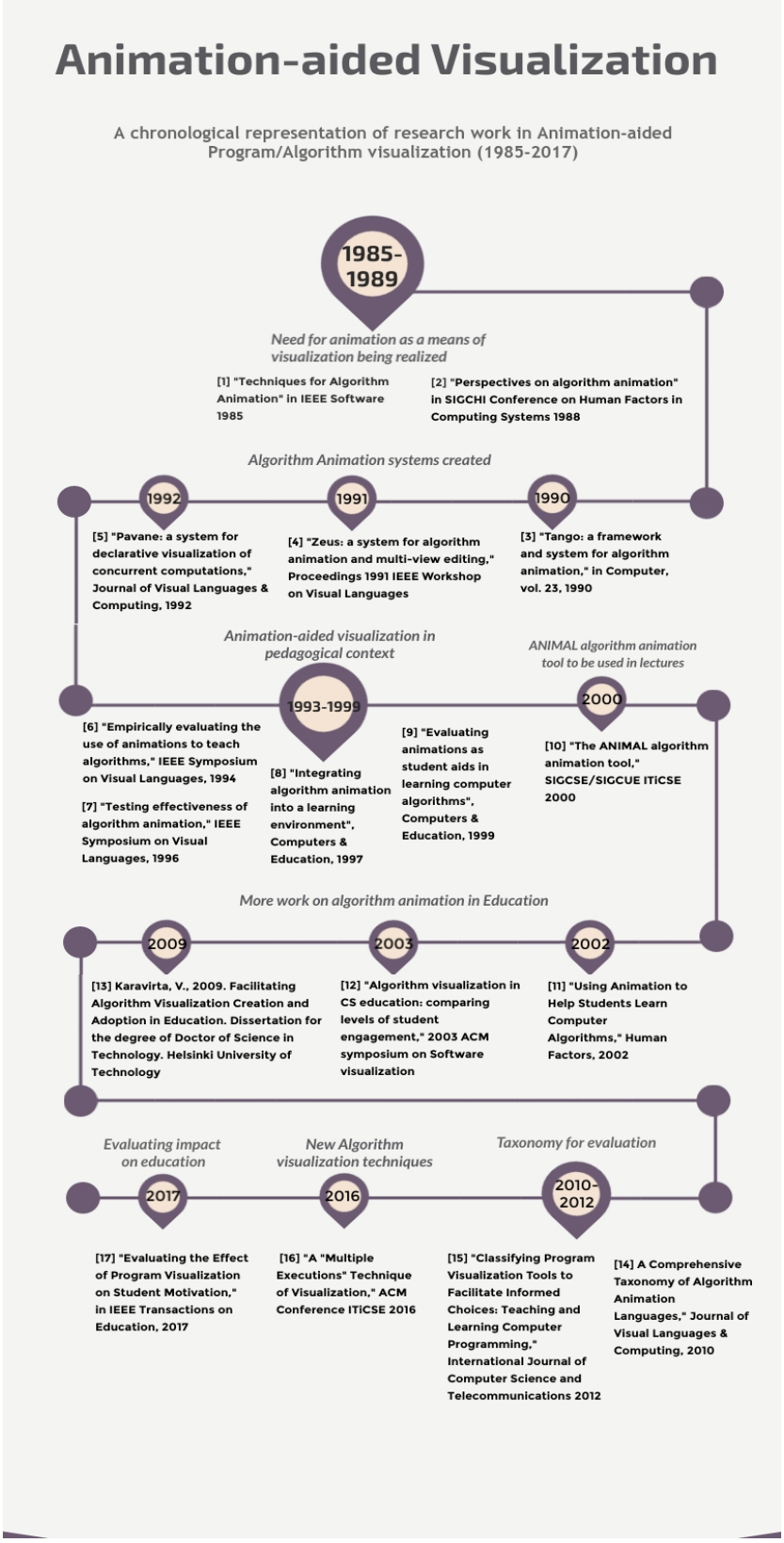


Figure 5. A chronological representation of research in the field of animation-aided visualization

2.2 Development of Education-aiding Program Visualization Tools through the years (1990s – 2015)

This section presents a brief history of Program and Algorithm Visualization systems over the last three decades. The field has evolved a lot since the first visualization systems were introduced.

Since the early 1990s, quite a few algorithm animation and visualization tools were being developed. Work in this area has been going on in different areas of the US and Europe. The use of visualization applications in education has increased over the last two decades. Many of the tools mentioned in this paper are extensively used in educational institutions around the world to effectively explain concepts, mostly algorithms in Computer Science education.

Programming is an important subject in computing and engineering fields. To ensure proper realization of the pedagogical objectives, program visualization tools are needed.

Figure 6 shows a timeline of some education-aiding program and algorithm visualization systems developed from 1990 to 2015. Plenty more systems exist, but the selected systems will be briefly described in this chapter.

TANGO

TANGO [18] was developed in 1990. It is a framework and system for algorithm animation organized as a set of cooperating components as opposed to one large self-contained system. The components include the program being animated, the code controlling the animation, and the Tango executable itself. Using Tango, programmers could create new animations in a few hours or days rather than many days or weeks.

ZEUS

Zeus [19] is an algorithm animation system that was created in 1991. The system could also be used for building multi-view editors. Zeus was noteworthy for its use of objects, strong-typing, parallelism and graphical development of views.

JAWAA

JAWAA [20] is a command language for creating animations of data structures and displaying them with a Web browser. Commands are stored in a script file that is retrieved and run by the JAWAA applet when the applet's Web page is accessed through the Web.

ANIMAL

ANIMAL [21] is an algorithm animation tool that was developed in 2000. It offers animated visualizations of numerous algorithms. The tool can also be used to create animations to be used in lectures.

SRec

SRec [22] was created in 2003. It is a visualization system specifically designed to animate recursion in Java programs. It is intended to assist in algorithm courses to better understand and analyze algorithm behavior.

ALICE2

ALICE is an open-source object-based educational programming language with an integrated development environment (IDE). It is a block-based programming environment that makes it easy to create animations and build interactive narratives. ALICE was first introduced in 1998. ALICE2 was released in 2004.

jGRASP

jGRASP [24] was introduced in 2004. It is an integrated development environment with visualizations for teaching and learning Java. The visualizations include Control Structure Diagrams, UML Class Diagrams, and new dynamic object views.

Jeliot3

Jeliot3 [25] was created in 2005. It is a program visualization tool designed to aid students to learn object-oriented programming. It visualizes how a Java program is interpreted. Method calls, variables, operation are displayed on a screen as the animation goes on, allowing the student to follow step by step the execution of a program.

AlViE

AlViE [26] is a post-mortem algorithm visualization Java environment. It was introduced in 2007. It is an algorithm visualization tool with an XML-based scripting language.

VISUALGO (2012)

VisuAlgo [27] was first developed in 2012. It is a web-based algorithm visualization tool without the need to install any additional software. The application offers visualizations for various algorithms. As of 2015, this application recorded 2000 sessions daily from worldwide visitors; it has grown more popular since then.

Explain Git with D3

Explain Git with D3 [28] is a web-based application which can be used to understand basic git concepts visually. This tool can be used without installing any additional software. Explain Git with D3 was developed in 2014 and the code is freely available on Github.

Loupe

Loupe [29] was introduced in 2015. It is a web-based application which runs entirely in the browser. It is a visualization to help students understand how JavaScript's call stack/event loop/callback queue interact with each other. The code is freely available on Github.

2.2.1 Timeline of program visualization systems (1990s – 2015)

Figure 6 depicts a timeline of the above-mentioned algorithm and program visualization applications developed over the last three decades.

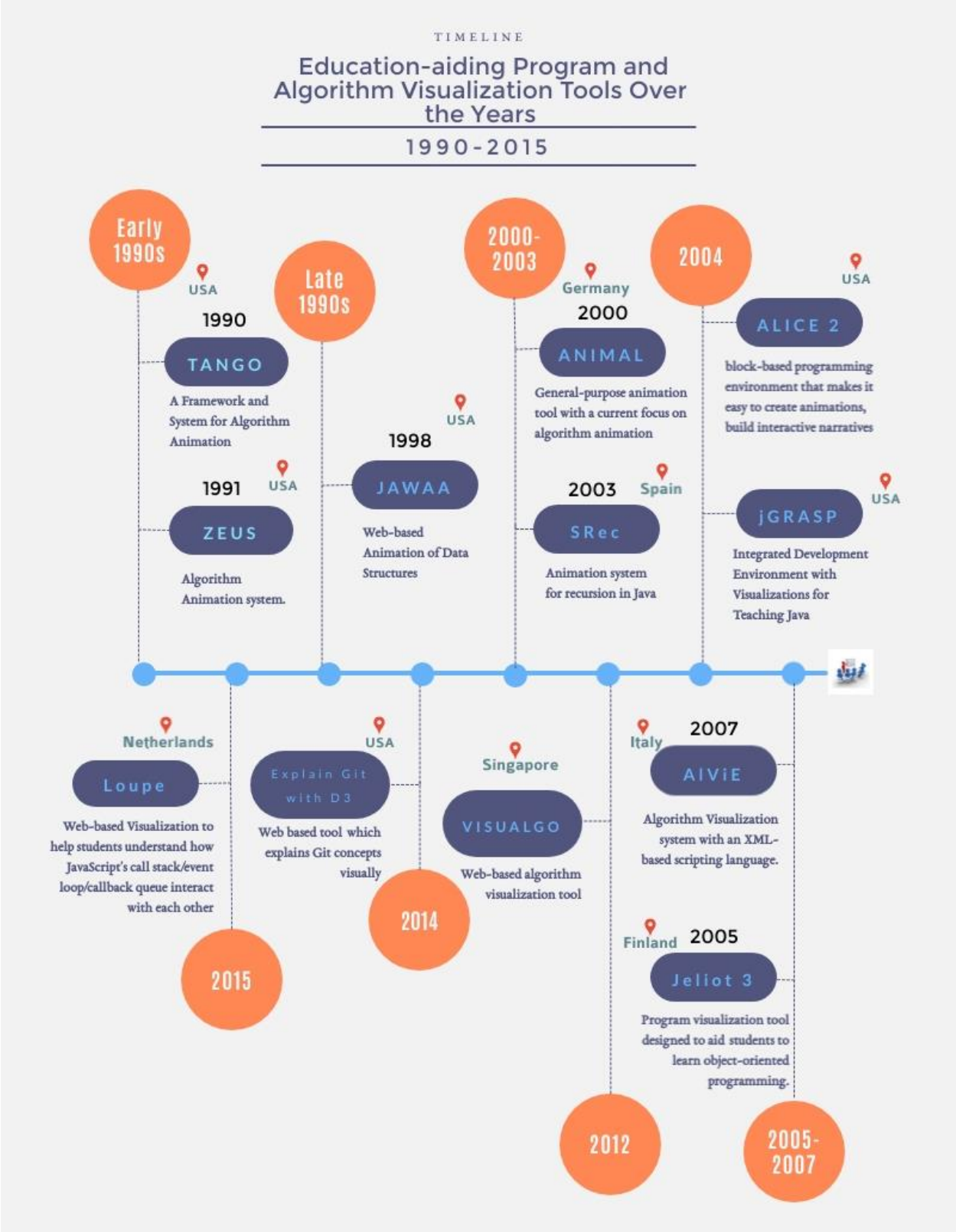


Figure 6. History of some Algorithm and Program Visualization Systems. The location symbol denotes the country in which the tool was first developed

CONCLUSION AND FUTURE WORK

Since the 1980s, many algorithm visualization tools have been developed to support teaching and learning core computer science topics such as data structures and algorithms. The success of these tools depends largely on the features they support. For example, increasing student engagement via user interaction with the visualization improves learning effectiveness.

The taxonomy developed for this project and the survey of the five AV applications based on it has revealed the strengths and limitations of these applications.

- ***Table 2*** evaluates the five applications in category Concurrency. Concurrency denotes features like animation and pseudo code are concurrently on screen and step of pseudo code currently being depicted by animation is highlighted. Table 2 indicates that majority of the applications include the features.

- ***Table 3*** summarizes evaluation of the applications in category Multiple Data Representations. It considers application ability to provide more than one representation of input/output data for depicted concept, if at all multiple input/output data representations is possible. Table 3 indicates that multiple data representations are not implemented in these tools.

- ***Table 4*** provides evaluation of applications in category Additional Information. Table 4 shows that most applications do contain the feature where there is an explanation of the concept being visualized or the tool provides urls or other materials like sound clip, video etc. that effectively explain the concept.

- ***Table 5*** evaluates the applications in subcategory Control under parent category User Interaction. Control denotes features which enable user to control the execution of animation like pause, resume, animation speed etc. Increasing student engagement via user interaction significantly improves learning effectiveness, so features enabling user interaction with the animation is essential to have in AV applications. Table 5 depicts that while some applications have these features, some don't.

- ***Table 6*** summarizes evaluation of the applications in subcategory Changing under parent category User Interaction. Changing denotes application feature to show animated visualization with user data as input. Table 6 indicates majority of the applications support this feature.

- ***Table 7*** evaluates the applications in the category Ease of Visual content creation. The ability to create animations and visual representations using an easy-to-use language is a need many educators feel, especially in the field of Computer Science. Table 7 depicts that majority of the applications do not provide this feature.

This taxonomy can be used as a tool to evaluate and compare animation-aided visualization applications and identify their strengths and limitations. It can be utilized to identify desirable features while developing an AV application.

For future work, this taxonomy can be further expanded. More rigorous study of various animation-aided visualization tools can lead to identification of more attributes that improve learning efficiency. New attributes can be added to the taxonomy to make it more comprehensive. The taxonomy can also be modified to include features focusing on aiding teachers and students in choosing the most appropriate tool for a better experience in the classroom.

BIBLIOGRAPHY

[1] M. Brown and R. Sedgewick, "Techniques for Algorithm Animation" in IEEE Software, vol. 2, no. 01, pp. 28-39, 1985.

[2] M. H. Brown. 1988. Perspectives on algorithm animation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88), J. J. O'Hare (Ed.). ACM, New York, NY, USA, 33-38.

[3] J. T. Stasko, "Tango: a framework and system for algorithm animation," in Computer, vol. 23, no. 9, pp. 27-39, Sept. 1990.

[4] M. H. Brown, "Zeus: a system for algorithm animation and multi-view editing," Proceedings 1991 IEEE Workshop on Visual Languages, Kobe, Japan, 1991, pp. 4-9

[5] Gruia-Catalin Roman, Kenneth C Cox, C.Donald Wilcox, Jerome Y Plun, "Pavane: a system for declarative visualization of concurrent computations," Journal of Visual Languages & Computing, Volume 3, Issue 2, June 1992, Pages 161-193

[6] A. W. Lawrence, A. M. Badre and J. T. Stasko, "Empirically evaluating the use of animations to teach algorithms," Proceedings of 1994 IEEE Symposium on Visual Languages, St. Louis, MO, USA, 1994, pp. 48-54.

[7] J. S. Gurka and W. Citrin, "Testing effectiveness of algorithm animation," Proceedings 1996 IEEE Symposium on Visual Languages, Boulder, CO, USA, 1996, pp. 182-189.

[8] Charles Kann, Robert W.Lindeman, Rachelle Heller, "Integrating algorithm animation into a learning environment," Computers & Education, Volume 28, Issue 4, May 1997, Pages 223-228

[9] Michael D.Byrne, Richard Catrambone, John T.Stasko, "Evaluating animations as student aids in learning computer algorithms," Computers & Education, Volume 33, Issue 4, 1 December 1999, Pages 253-278

[10] Guido Rößling, Markus Schüer, and Bernd Freisleben. 2000. The ANIMAL algorithm animation tool. In Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education (ITiCSE '00). ACM, New York, NY, USA, 37-40. , 87-94

[11] Catrambone, R., & Seay, A. F. (2002). Using Animation to Help Students Learn Computer Algorithms. Human Factors, 44(3), 495–511.

[12] Scott Grissom, Myles F. McNally, and Tom Naps. 2003. Algorithm visualization in CS education: comparing levels of student engagement. In Proceedings of the 2003 ACM symposium on Software visualization (SoftVis '03). ACM, New York, NY, USA, 87-94.

[13] Karavirta, V., 2009. Facilitating Algorithm Visualization Creation and Adoption in Education. Dissertation for the degree of Doctor of Science in Technology. Helsinki University of Technology

[14] Karavirta, V., Korhonen, A., Malmi, L., and Naps, T. A comprehensive taxonomy of algorithm animation languages. Journal of Visual Languages & Computing 21, 1 (2010), 1-22

[15] S. Mutua, F. Wabwoba, P. Ogao, P. Anselmo, and E. Abenga, “Classifying program visualization tools to facilitate informed choices: teaching and learning computer programming,” Int. J. Comput. Sci. Telecommun., vol. 3(2), February 2012.

[16] J. Ángel Velázquez-Iturbide, Isidoro Hernán-Losada, and Antonio Pérez-Carrasco. 2016. A "Multiple Executions" Technique of Visualization. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '16). ACM, New York, NY, USA, 59-64.

[17] J. Á. Velázquez-Iturbide, I. Hernán-Losada and M. Paredes-Velasco, "Evaluating the Effect of Program Visualization on Student Motivation," in IEEE Transactions on Education, vol. 60, no. 3, pp. 238-245, Aug. 2017.

[18] J. T. Stasko, "Tango: a framework and system for algorithm animation," in Computer, vol. 23, no. 9, pp. 27-39, Sept. 1990

[19] M. H. Brown, "Zeus: a system for algorithm animation and multi-view editing," Proceedings 1991 IEEE Workshop on Visual Languages, Kobe, Japan, 1991, pp. 4-9.

[20] Willard C. Pierson and Susan H. Rodger. 1998. Web-based animation of data structures using JAWAA. SIGCSE Bull. 30, 1 (March 1998), 267-271

[21] Guido Rößling, Markus Schüer, and Bernd Freisleben. 2000. The ANIMAL algorithm animation tool. In Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education (ITiCSE '00). ACM, New York, NY, USA, 37-40

[22] J. Ángel Velázquez-Iturbide, Antonio Pérez-Carrasco, and Jaime Urquiza-Fuentes. 2008. SRec: an animation system of recursion for algorithm courses. SIGCSE Bull. 40, 3 (June 2008), 225-229.

[23] Cooper, S., Dann, W., & Pausch, R. (2000). Alice: a 3-D tool for introductory programming concepts. Paper

presented at the Journal of Computing Sciences in Colleges

[24] J. H. Cross, D. Hendrix and D. A. Umphress, "JGRASP: an integrated development environment with visualizations for teaching java in CS1, CS2, and beyond," 34th Annual Frontiers in Education, 2004. FIE 2004., Savannah, GA, 2004, pp. 1466-1467

[25] Andrés Moreno, Niko Myller, Erkki Sutinen, and Mordechai Ben-Ari. 2004. Visualizing programs with Jeliot 3. In Proceedings of the working conference on Advanced visual interfaces (AVI '04). ACM, New York, NY, USA, 373-376

[26] https://www.pilucrescenzi.it/wp/software/alvie/

[27] https://ioinformatics.org/files/volume9.pdf#page=245

[28] https://github.com/onlywei/explain-git-with-d3

[29] https://github.com/latentflip/loupe