An Overview of Visualization in Mathematics, Programming and Big Data

Ilja Golland
ilja.golland@rwth-aachen.de

Desared Osmanllari
desared.osmanllari@rwth-aachen.de

Figure 1. Source: `www.sas.com/en_sg/insights/big-data/data-visualization.html`

ABSTRACT

Visualization is a descriptive way to ensure the audience attention and to make people better understand the content of a given topic. Nowadays, in the world of science and technology, visualization has become a necessity. However, it is a huge challenge to visualize varying amounts of data in a static or dynamic form. In this paper we describe the role, value and importance of visualization in maths and science. In particular, we are going to explain in details the benefits and shortages of visualization in three main domains: Mathematics, Programming and Big Data. Moreover, we will show the future challenges of visualization and our perspective how to better approach and face with the recent problems through technical solutions.

Author Keywords

Visualization, Statistics, CodeCity, Profiler, Statistical Analysis, Blockly, Scratch, Scalability, Big Data, Data Reduction, Data Latency, Spatiotemporal dimension

INTRODUCTION

The main purpose of visualization is data communication. This is done through data transformation from an abstract form into meaningful information. Visualization offers a way from which people can learn something useful from ambiguous data. New techniques, mainly dependent from the domain they cover, are used to translate raw numbers and statistics into visible images. At first we will take a look at an approach to make statistical analysis easier to grasp. As statistics in general is an area which is hard to approach for those without enough technical background, visualization can help to introduce them to this field. We discuss a tool proposed by Kandel et al. which visualizes different data types in a helpful way and briefly discuss it with our own view. Our second approach in this paper is to show the role of visualization in education. Studies have shown that visualization has significantly improved the skills of novice students [2]. Students tend to get a better essence of content through visualization. These experiments have their own limitations and require extra training time in visualization techniques. However, our focus lies in the effects of visualizations for computer science students and overall there is evidence that visualizing images have a positive impact on students performance. Consequently, we thought to analyze in more details the power of visualization. Does it have such a meaningful impact in specific scientific domains? What is the role of visualization nowadays and how challenging is this process? To better define these questions, we cover in details the importance of

visualization in three main domains: Mathematics, Programming and Big Data.

- Visualization in Mathematics is a broad domain. In this section, we will focus mainly in visualization of statistical analysis and business mathematics. In the first field, we will deeply present quality assessment of integrated statistical analysis visualization. How to present statistical analysis clearly and what tools are necessary for managing data quality? Sometimes duplicate data values undermine the process of visualization and exhibit wrong results. These failures instead of helping the researcher, prevent her from retrieving useful information from statistical analysis.
  In business mathematics section, we will present some visual materials such as Marshallian demand function [6] and Black-Sholes Model [21]. Not only will we describe such an approach, but also will we show the benefits, shortages and challenges of future visualization processes for business people. Business Mathematics is strongly related to mathematics education and statistics, so we decided to cover these topics as a single domain.
- Aside from mathematics, programming and algorithms are also theoretical concepts which can be illustrated and visualized in order to be conveyed more easily to students. At the same time developers who code and design applications benefit from visual feedback and visualizing tools. Concerning this domain we discuss different kinds and aspects of visualization for both target groups. We also take a look at different tools for visualization and what kind of help and impact those have.
- Big Data visualization techniques are new. Until lately, a single pie chart or table might have been enough to visualize a given data set. This technique can not be implied in Big Data, because scientists have to deal with an enormous amount of data, coming from different sources and having lots of hidden insights. Traditional tools couldn't support the "big scale" of such ambiguous data. Consequently, there is a need to offer new creative opportunities in order to build unique interfaces. In the section of Big Data, we will cover not only the latest problems, but also the future challenges: How to improve this approach by integrating technical tools? Also, we will show some techniques, examples and properties how big data visualization process works. Moreover, we will describe briefly the basic visualization tools, just to attract the reader attention by visualizing these techniques. What is needed to mention in this section is the importance of computing power. GPUs developed today help the process of data manipulation, reduction and randomization. Visualizing *Big Data* is as exciting as it is challenging. The final aim is to lighten users' cognitive load by presenting graphical images of data.

Visualization has improved not only the perception of science, but also the techniques of approaching it. In each of the sections we describe below, we will show some attributes, properties and tools of visualization. Improving these elements by cleaning noisy data and retrieving valuable information is one of the biggest challenges of visualization.

VISUALIZATION IN MATHEMATICS

Visual materials explain what numbers can not. In Business Mathematics, visualization is a necessity not only for new students to better understand the economic concepts, but also for professors and scientists to clearly explain problems and mathematical concepts. In this section, we will analyze how visual materials help the process of learning by explaining several concepts in economics and statistics.

Statistical Analysis

Applied statistics can be split in 2 major areas. Descriptive statistics is a method to order the raw data and depict them more clearly. While statistics is a tool to observe events and find new trends or connections, in most cases only sample sizes of the target group are taken. In order to derive analysis from this sample group to the whole target group or population, inferential statistics can help to achieve that. However, novice users oftentimes struggle with this process of statistical analysis because they either are not aware of it or do not know how to do it properly. Microsoft Excel and other spreadsheet programs are very popular when it comes to working with managing raw data. They also help with creating diagrams and therefore visualize the numbers in order to see some impacts or trends. The issue here is the scalability as with data sets or samples sizes in the amount of several hundreds it starts to become less efficient. Furthermore tools such as Excel do offer inferential statistics but without some knowledge the user will not know what to gain from the results as there is no guidance. Raw data also exposes the issue of "missing, erroneous, extreme and duplicate values [which] undermine analysis and are time-consuming to find and fix" as has been stated by Kandel et al. [9] in their work.

In their work, they present a visual analysis tool named Profiler which helps researchers and analysts to tackle these issues and also "assessing quality issues in tabular data". Profiler uses inference and data mining methods to suggest and display varying summary visualizations depending on the context of the data. These methods also tackle the above mentioned issues by identifying data quality issues automatically. Furthermore it alleviates the scalability issue by allowing to work with "millions of data points". While their work provides several contributions, we will focus on the main aspects of their provided tool.

The authors describe an explicit usage scenario in order to explain the functionality. We will briefly summarize this scenario in the following way. A user wants to skim through movie data and crawls data from several online movie databases and loads them in Profiler. Using this data she can choose different anomalies in order to find missing data, erroneous data, inconsistent data, extreme values, and key violations. She chooses missing data for *MPAA Ratings* and Profiler opens up a bar chart because it seems to be best fitted for this data types (see Figure2). In this bar chart she can see the grey bar indicating that that there are missing values so she clicks on it. In the paper it is unclear if the other charts open up after clicking on the grey bar or if they already appeared after clicking on the *MPAA Rating* and then changed. Regardless, in the *Release Date* chart she then sees missing

data points corresponding to earlier release dates.

Another possible scenario is when the user wants to investigate the extreme values anomaly. For this she clicks *Worldwide Gross* in the same anomaly list and then sees corresponding charts (see Figure 3). Again, the originally described text offers ambiguous explanations by describing the charts and what is displayed. According to them the user can see that the *Worldwide Gross* is correlated to the *US Gross* and *Production Budget*. Further she can also see that summer and winter seasons have movies with high gross. However, it is unclear what the colors represent in this explicit scenario so we assume it is U.S. gross and worldwide gross.

As we have mentioned previously Profiler is able to automatically choose the best visualization of a data set depending on its type. As shown in Figure 4, a user can also choose e.g. *Release Location* and combine it with another data set to see the corresponding geographical location.

These scenarios and examples show both advantages and also disadvantages of Profiler. Hereby we want to note that the tool is still under work and according to the authors it is planned to be usable by end users in the future. Consequently, our opinion is based solely on the screenshots and usage scenarios found in the paper. The automated and suggested visualization of the data types can help those who normally would have to choose the best fitted visualization on their own, which means choosing from scatterplots, histograms, bar charts, or other types. Although this helps by reducing the effort to choose a type, it seems that the user is not able to choose a different type afterwards in order to gain a better or different insight and therefore has to rely on the recommendation. Although in its current state the tool is target towards analysts (and possibly researchers), the usability still shows ambiguous elements. For example in Figure 2, as described, the grey bar indicates missing values. Without guidance or help, the user would be unable to know this. The same holds for the 2 different colors depicted in both Figure 3 and Figure 2 as it is unclear which color represents what data or attribute. Another general issue is that although Profiler seems to help make connections and also possible correlations between different data sets, there are no visual cues or reports highlighting indicating those, meaning the user still has to find those on her own. This is slightly similar to the output of statistical testing methods in Excel, in which the user still needs to form his own conclusion.

In their paper the authors state that they "plan to evaluate Profiler through both controlled studies" and "intend to develop a tool for end users to define custom types". This shows that these issues might be addressed in future versions.

Nonetheless in our opinion this might be a good approach to allow end users to find anomalies in their data easily and also find correlations between different data among fitting visualizations.

Business Mathematics

Visual approach to business mathematics materials leads to a better understanding of mathematical processes. First of all, we will focus on business teaching materials and their

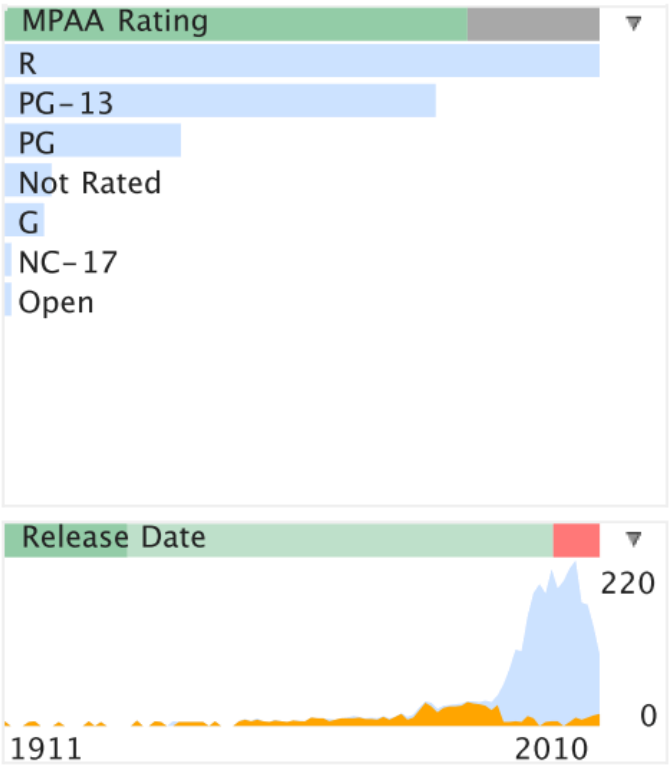

Figure 2. ***MPAA Ratings*** **combined with the** ***Release Date*** **(Image modified from Kandel et al. [9]**

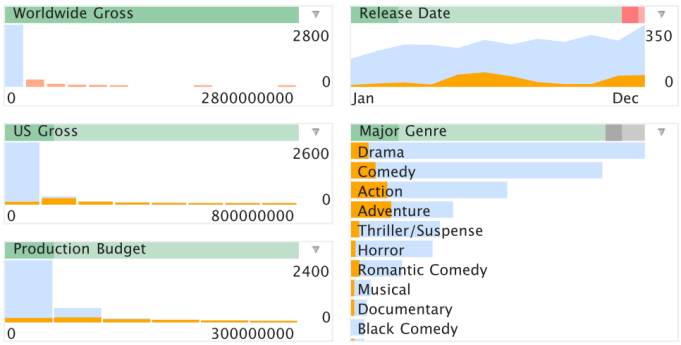

Figure 3. Automatically generated views to help assess Worlwide Gross (Image taken from Kandel et al. [9])

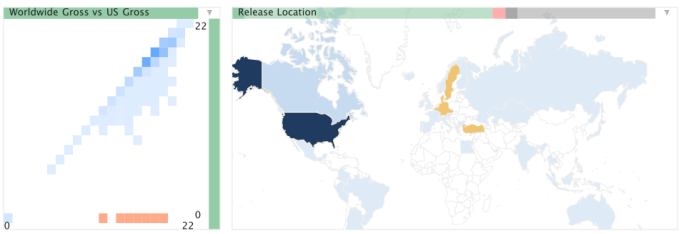

Figure 4. Example of a context aware visualization of geographical data (Image taken from Kandel et al. [9])

mathematical content. Then we will show how visualization help students to get familiarized with mathematical theorems.

Studies constructed through experimental research by Hashimoto et al. [6] proved our hypothesis that visualization helps business students into a better performance and understanding of mathematical processes. How did Hashimoto conclude in such a result and why is it important for us?
Business students were introduced to algebraic and visual approach teaching. Even though some students did not understand the algebraic calculation process, they could intuitively understand the essence only by seeing the visual effects. During the experiment, Hashimoto utilized a three-dimensional graphics approach for a typical economics problem called Marshallian demand function [6]. Marshallian demand function is a constrained optimization problem which expresses the amount of commodity that a consumer will buy as a function of commodity prices and an available budget or income. By changing the income amount, the students could better understand the fluctuation of the maximum point which leads to an easier understanding of the Lagrange multiplier. Lagrange Multiplier is the ratio of the changes in optimum values. Lagrange multiplier is explained in many mathematical textbooks [3, 7]. We are not going in depth of the Marshallian demand function, but we are analyzing the effects and results of this experiment. We used such an example to show that students are tended to understand the content and the mathematical processes much more easily through visualization rather than through algebraic explanation. Lagrange multiplier is an abstract concept and it can not be easily explained through formulas. Visualization helps not only students to learn new concepts, but also professors who teach business mathematics.

Usually experiments might be misleading if they are not followed by comparable works which lead to similar results. Consequently, we will explain another example from the finance field that proves our hypothesis. The second study comes from Shirota et al [21] , who presents several cases in finance and statistic fields. In his paper, Shirota presents visual teaching materials for the following topics: covariance in regression, central limit theorem, principal component analysis, price-yield surface and black-sholes model. We will mention the black-sholes model and make a correlation between both experiments mentioned in this passage. Black-Sholes model is a mathematical model of a financial market. This formula is widely-used in global financial markets by traders and investors to calculate the theoretical price of european options (a type of financial security). The formula has been demonstrated to yield prices very close to the observed market prices. Black-Sholes model requires complex mathematical calculations and this makes it hard to be explained by formulas. Consequently, Shirota has visualized the model and test the users if they understand the topic better. One component of the model is illustrated in figure 5. In this figure, we can see the fluctuation movement of many stock prices. The distribution of the probability for stock price is illustrated vertically. The probability distribution function is a logarithmic normal distribution. As

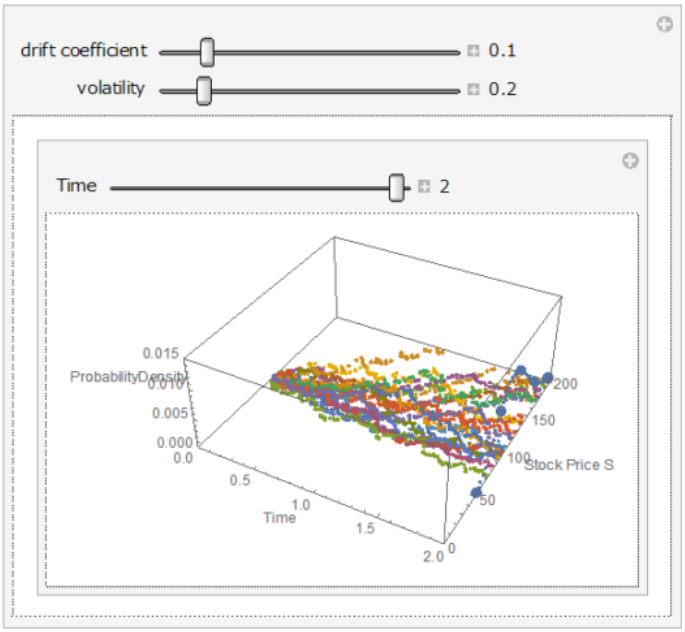

Figure 5. Visualization of Black-Sholes Model (Image taken from Shirota et al [21])

it is easily-perceived, while the time passes, a curve will be generated. With the slide bar, the user can change the current time and see how the probability distribution changes over time. This model helps the users to interact with the stock prices and deeply understand the Black-Sholes.
To sum up, the results retrieved by Black-Sholes visualization model and not only, help the students to intuitively understand the core elements of the mathematics and statistics. Moreover, business persons, researchers and professors should be grateful to the latest visualization tools available nowadays. The power of visualization should be spread and used more in practical business fields.

PROGRAMMING

As a subset of mathematics, computer science also provides vast and abstract topics with theoretical aspects. Many of these are fundamental, especially for undergraduate students or even high school students who are not familiar with formal and theoretical concepts. Earlier work [12, 5] provided evidence that courses which deal with programming and algorithms benefit from conveying a visual representation along with the theoretical background. Additionally, [10] tried to increase the motivation and reduce the drop-out rate of students by providing visual aid in the learning process. Although they only achieved an increase of 10%, they saw potential for future work. Algorithmic procedures and several data structures can be represented easily using simple graphical objects. The same holds true for concepts of different programming languages and paradigms, as can be seen in material and books from different computer science courses and topics.

Considering the practical side of computer science, developers who program applications can benefit from being visually aided in their developing process.

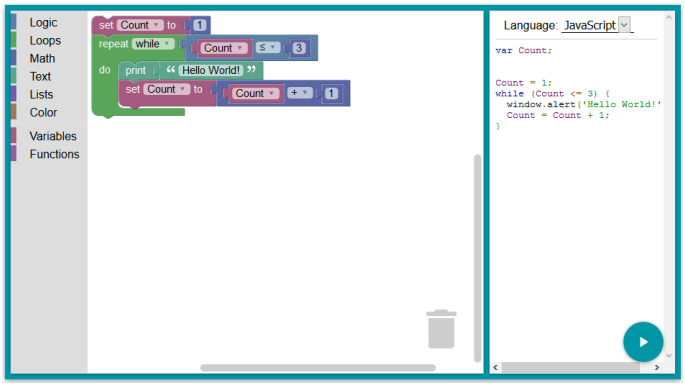

Figure 6. Default "Hello World" example using Blockly

In our present work, we focus on properties of different types and kinds of visualization in certain domains. Using visualization for students in order to help them better understand material, it becomes clear that the main purpose and goal should be improving this understanding and making it easier for them compared to a method which would go without visual aid of this type. Sensalire et al. analyzed different tools and showed results from their experience in order to evaluate these tools [14]. Combined with results and research of previous work they define "a SoftVis tool *effective* if it enables its users to achieve results in an easier and faster way compared to the traditional method of doing the same task". This definition is also applicable for tools targeting developers as they want to benefit from support and aid in order to be more effective in their workflow.

In the following section we will discuss and analyze some approaches, which both students and developers better understand the material using visualization tools.

Students

Programming has reached more popularity in the recent years due to initiatives like "Hour of Code"[1], which promotes and appeals to everyone that they can and should try to code. As we mentioned above, computer science itself provides many different theoretical fields and even programming might seem abstract from a novices perspective. Hour of Code asks beginners to solve little tasks and assignments by programming using Blockly, which is "library for building visual programming editors" by Google[2].

This approach does not focus on the theoretical background of programming (data structures or different paradigms) and also omits the syntactical aspect. Instead it provides an editor that shows building blocks are similar to puzzle pieces which then can be put together to construct an actual program, see Figure 6.

However, the basic concept of using blocks is not novel as Google explains on the same website "Blockly was influenced by App Inventor, which in turn was influenced by Scratch, which in turn was influenced by StarLogo". The idea

[1]https://hourofcode.com/
[2]https://developers.google.com/blockly/

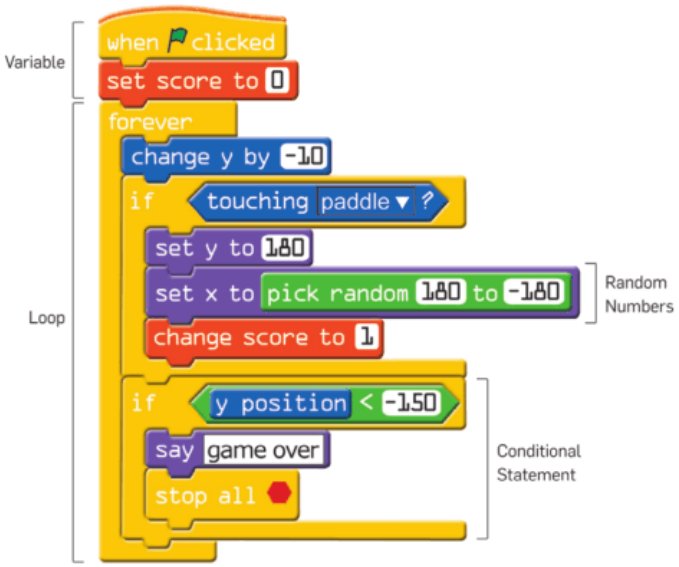

Figure 7. An example application made in Scratch with computation and mathematical concepts highlighted (Image modified from Resnick et al. [13])

behind Scratch was "to develop an approach to programming that would appeal to people who had not previously imagined themselves as programmers" and "to make it easy for everyone, of all ages, backgrounds, and interests, to program their own interactive stories, games, animations, and simulations, and share their creations with one another" [13]. An example application can be seen in Figure 7.

Ferrer-Mico et al. [4] conducted studies with 12-13 years old students in order to find the impact on the learning experience using Scratch. According to their results, "the majority of the students [were] able to realize that they [could] increase their knowledge construction if they [spent] longer time using the particular tool" and 90% of the probands thought that if they kept on working they would also increase their understanding and confidence. While they provide evidence for a positive effect with their participants, their studies lack more concrete observational data showing that students using Scratch or any other visual aid had a better experience compared to those without.

We think that using this real-world metaphor is a good approach in order to broaden the target group consisting of people who want to start with computer programming. The connection between the more abstract source code and blocks similar to puzzle pieces ,which people are more likely to be familar with, might bridge the gap which holds them back from being introduced to this field. Which is also in line with Resnick's idea of making it available to every person, regardless of all factors.

Developers

At the same time there are professional software engineers and developers who develop applications using integrated development environments (IDEs). These play a crucial part in the coding process as they support the developer with a

wide spectrum of different tools to understand the code itself and find problems easier. However, with time the general amount of code for software projects increased. Car software projects have about 100 million LOC (Lines Of Code) and all of Google's web services amount to over 2 billion LOC [8]. In order to keep the code organized and know what is happening at what point and in which file, developers often rely on these features of IDEs. According to Sensalire et al. [14] the problem occurs "[w]hen trying to understand large programs, [as] it is not easy to get the complete picture just by browsing through the code, as a lot of information can be easily missed" which is also supported by [11]. These issues for programmers can be supported by software visualization (SoftVis) tools, whereas Sensalire et al. [14] define software visualization itself as "the use of interactive computer graphics, typography, graphic design, animation, and cinematography to enhance the interface between the software engineer or the computer science student and their programs" in the same context. Similarly Stasko et al. [15] define it as "[t]he use of the crafts of typography, graphic design, animation, and cinematography with modern human-computer interaction and com- puter graphics technology to facilitate both the human understanding and effective use of computer software"

In their work, Wettel et al. [17, 18] created a framework to visualize software projects as 3D cities. Their goal was "to give the viewer a sense of locality to ease program comprehension", which supports developers by allowing them to view their software projects not only via more abstract code but as something more familiar from the real world. In this regard it is similar to Blockly's approach, making a connection between abstract program code and blocks which are similar to physical puzzle pieces. In order to accomplish this, they mapped properties of Object Oriented Programming (OOP) such as "packages, classes, methods, attributes, and all their explicit and implicit relationships" to properties similar to buildings in a city. Figure 8 is an example which illustrates what a class might look like when being visualized as a city. While a single building represents a class, its height and width as well as length represent number of methods respectively number of attributes. Surrounding rectangles which encompass multiple buildings and appear as districts represent packages.

In a later work Wettel et al. [20] evaluated their approach in order to "provide experimental evidence of the viability of our approach in the context of program comprehension". Their study design involved several tasks which needed to be solved by 41 developers (21 from academia and 20 from industry). The dependent variables measured were the task completion time and the correctness with the independent variables *tools* and *object size*. The participants were either using CodeCity or Eclipse together with Excel (a sheet containing all the necessary metrics for the software project) as tools to solve the tasks (a between-groups approach). According to their review of related works, the dichotomy of participants from both academia and industry as well as different sizes of the projects (measured in classes and kLOC) are a distinctive feature (see Figure 9).

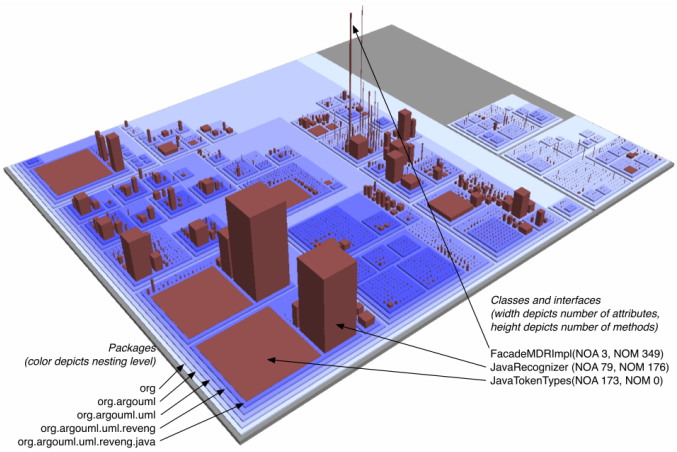

Figure 8. An overview of the city of ArgoUML v.0.24 (taken from [19])

Experiment	Subjects		Object system(s)	
	Academia	Industry	Classes	kLOC
Storey et al. [14]	30	0	17	2
Marcus et al. [6]	24	0	27	42
Lange et al. [3]	100	0	38 39	? ?
Quante [10]	25	0	475 1,725	43 160
Cornelissen et al. [1]	20	1	310	57
Wettel et al. [20]	21	20	1,320 4,656	93 454

Figure 9. Comparison of related work (taken from Wettel et al. [20])

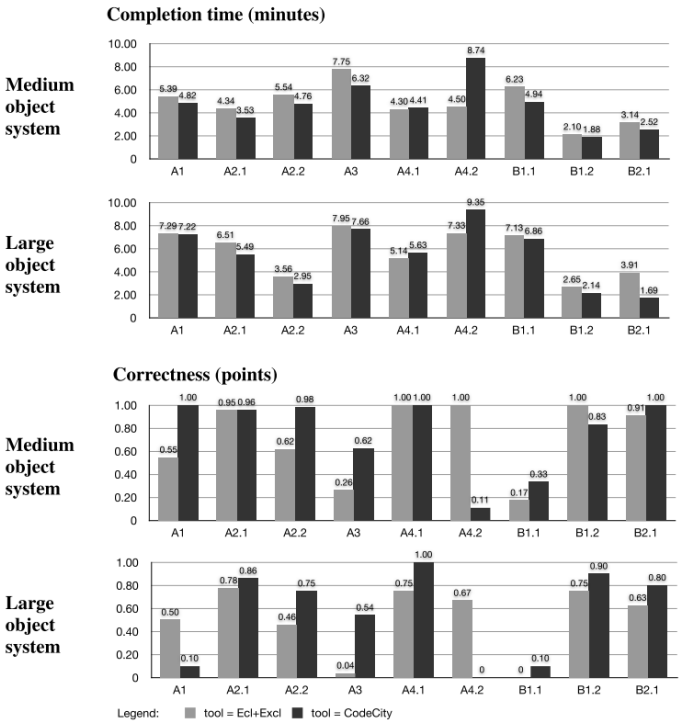

Figure 10. Average correctness and completion time per task (Image modified from Wettel et al. [20])

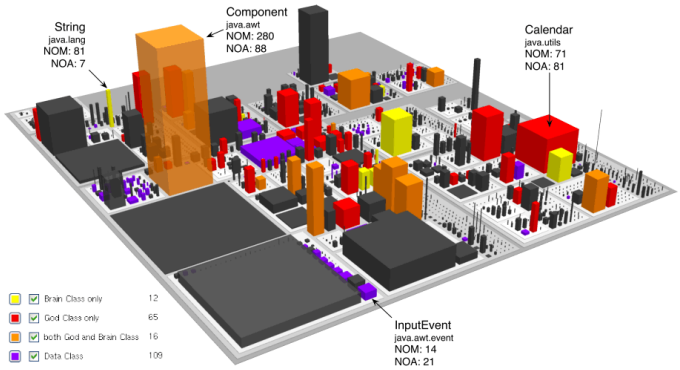

Figure 11. Example of disharmony maps (Image taken from Wettel et al. [20])

In previous work the participants consisted almost only of students in academia and the systems used (in most cases) were not big enough and therefore "not representative for a realistic work setting". Their results showed that in terms of correctness, using CodeCity lead to an increase of 24% and regarding task completion time 12% faster than the Eclipse+Excel method (the average completion time and correctness can be seen in Figure10). Although they included sections about threats to internal and external validity, in our opinion there are some crucial points missing which are additional threats or issues the authors should have considered.

Representative set of tools: The tasks were supposed to be solved by using either CodeCity or Eclipse together with Excel. According to the authors the combination of the latters were chosen because Eclipse itself provided source code exploration and Excel added the possibility of "exploring meta-data, such as software metrics and design problem data, since they were not available in Eclipse". The design problem data mentioned here refers to so called design flaws, which emphasize "bad design" decisions in terms of code. An example would be that a class should not have more than 6 sub-classes, which then (in case it has) can be visualized using so called disharmony maps as can be seen in Figure 11.

However, this is not representative for an actual working environment for a significant reason. The Excel sheet given to the participants seemed to be already filled with the needed meta-data for the metrics. A developer who works with Eclipse would not have such Excel sheet and would need to fill it first, which requires time and effort. The authors stated that they were unable to find a plugin for Eclipse which fitted the requirements. We think that if a developer is really interested in meta-data such as these metrics, she might write a plugin herself or use an IDE which supports this feature or has plugins which allow this.

Comparison of CodeCity with other visualization: Using these two tools (CodeCity and Eclipse+Excel), the authors compared a tool which used visualization with a source code exploration tool. Although the tasks were suited and solvable with both of them, it would have been also interesting to see how it compares with other visualization tools. This is also mentioned briefly in the conclusion, as the authors state to believe that the reasons for the results are "due to both the visualization as such, as well as the metaphor used by CodeCity, but we can not measure the exact contribution of each factor". This further emphasizes the fact, that with this study, it is not clear if CodeCity helped the developers or if it was the visualization aspect.

Willingness of the developers to use CodeCity: As mentioned before, developers tend to have their own workplace they are already familiar with. Among other factors this workplace also encompasses the software they use to develop, more specifically the editors. Since they are used to this environment due to experience and their workflow, they do not only need to know and be sure that if they change this setting they will improve their work, but also be comfortable switching to a different setting. The authors provide evidence that using CodeCity, at least compared to Eclipse+Excel, can improve the work. However, from our point of view, they should have also surveyed the aspect of acceptance. More specifically, they should have asked the developers if they could imagine integrating CodeCity into their current working environment.

In addition to these, CodeCity aims to provide a better overview for the current software project and classes as well as the corresponding attributes and methods. Despite its ability to visualize meta-data, these data are mostly represented in the amount of it and does not show information about the attributes and methods themselves. Hence, considering the size and dimensions of a building, a developer is only able to see (or estimate) the amount of attributes and methods but unable to derive such he might need or want to inspect. Furthermore, an issue also addressed by the authors, with the current version of CodeCity is difficult to see the relations between different classes "such as inheritances, method invocations and attribute accesses" [18]. As of now, relationships can be set to be shown but it can lead to "visual occlusion and decreased realism".

BIG DATA

Big Data is sometimes an abstract concept, especially for non-scientific people. Nowadays, plenty of information is necessary to be analyzed for various purposes: science, medicine, statistics, manufacturing, sports etc. To make all this information meaningful, many visualization techniques are used. However, this process is as challenging as it is thriving. Visualization of big data is not an easy process as scalability is an important issue [1]. In this section, we will review some visualization tools, the visualization process, opportunities offered by the approach of big data visualization and some challenges which make this process complicated.

- Challenges of Big Data Visualization
  Visualizing *Big Data* is a demanding task. It requires a lot of effort to identify duplicate, wrong, or missing values. Presenting valid information by using traditional techniques is almost impossible because of the vast amount

of data it supports and due to some of the following challenges.

  - Real-time Scalability
    Visualizing real-time information is crucial, especially if users need to make real-time decisions based on given data. How can big data be processed in real-time without causing latency and spoiling information? It is very difficult, because many data sets can not fit in actual storage systems and also such huge queries can not be processed in real-time. Consequently, visualizing data will cause high latency. Overcoming such problems like limited memories and data processing capabilities is challenging in big data visualization. To understand what amount of data we are analyzing in this section, we will take some simple examples from Facebook statistics [3]. In the third quarter of 2012, the number of active Facebook users had surpassed 1 billion. As of the first quarter of 2016, Facebook had 1.65 billion monthly active users, where 83 millions are thought to be fake accounts. There are 300 million photos uploaded per day. Every 60 seconds on Facebook there are 510 comments posted, 293,000 statuses updated, and 136,000 photos uploaded. If our purpose is to visualize in real-time only the mentioned statistics above, then we will need powerful machines processing complex queries and containing big storage. Otherwise, this process will be impossible.
  - Perceptual Scalability
    Human perception is one element of perceptual scalability. When data becomes extremely large, human eyes have difficulties to extract the information. To visualize big data accordingly to human perception, visualization systems must have scale accuracy to make the information meaningful. Secondly, limited screen displays do not help the visualization process. Visualizing big amounts of data, which may contain millions of entry points is challenging. There is no effective way to visualize information retrieved from big data into conventional displays (1-3 million pixels). Visualizing every data point might lead to overplotting, overlapping and may overwhelm users' perceptual and cognitive capabilities [16]. How can we escape from such a barrier? There are many novel and abstract techniques which are tested lately to deal with the visualization in limited screen displays. We are not going to explain them in details, since our purpose is just to mention why perceptual scalability is challenging in big data visualization. Moreover, this is a new research area, where many scientists are working on inventing new technologies to display big data in modern and capable screens.

- Data and Latency Reduction
  Challenges can not be neglected. In this section we will briefly describe what techniques are followed to cover the problems explained above.
  Data Reduction is the most effective strategy to address the above-mentioned challenges of big data visualization. Data reduction mainly consists of resolution reduction on results by displaying all the information in a limited display. This process must save the credibility of information without violating the results. Data reduction techniques include sampling, filtering, and binned aggregation [1]. All of them work towards the same aim: reduce big data to small amenable data before visualization.
  Another concerning challenge on big data visualization is latency. As explained above, because of vast amount of data we are dealing with, it is almost impossible to work on visualization process without reducing the system latency. What techniques are used to reduce latency? Pre-computed data is a very popular strategy in visualization process. It is used to improve interactive scalability such as dragging, panning, zooming, and dropping. It can support quick exploration rather than generate image tiles intended for direct displays. Such a strategy is widely used in Google Maps. Another technique is Parallelize Data Processing and Rendering. Kaddadi et al. [1] explain it briefly: "Data tiles can be very large in the process of aggregation, because it depends on the process of resolution. If data tiles have more than millions values, it will increase latency of aggregation. To speed up this process, visualization system can use a dense indexing scheme that simplifies parallel query processing." In web browsers, the web application could use WebGL to leverage parallel processing on the GPU. For example: immense system [16]. Moreover, to improve real-time scalability, a predictive middleware [1] is tested for residing between frontend visualization interface and backend data store. Predictive middleware will predict pre-fetched and cache data for future purposes. To conclude this section, it is important to mention that many other techniques are under research. Dealing with big data visualization problems is as much challenging as the visualization process itself.

- Visualization Tools
  In this section we introduce a couple of visualizing tools. They include services, platforms, widgets, and libraries. These tools help the visualization process, while most of them are specific for given domain. Some of them focus on frontend visualization components, the others on backend data management and query processing.

  - Data-Driven Document(D3)
    Data-Driven Documents(D3) is an interactive big data visualization tool. It uses JavaScript libraries to manipulate HTML documents based on data. Why should we use Data Driven Document? First of all, D3 makes the data interactive through the use of transformations and transitions such as zooming and panning. It helps attach data to DOM (Document Object Model) elements. Moreover, it is an easy way to bring data to life using well-known web technologies such as HTML, SVG, and CSS. D3 offers the opportunity to create complex graphs and

[3] http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

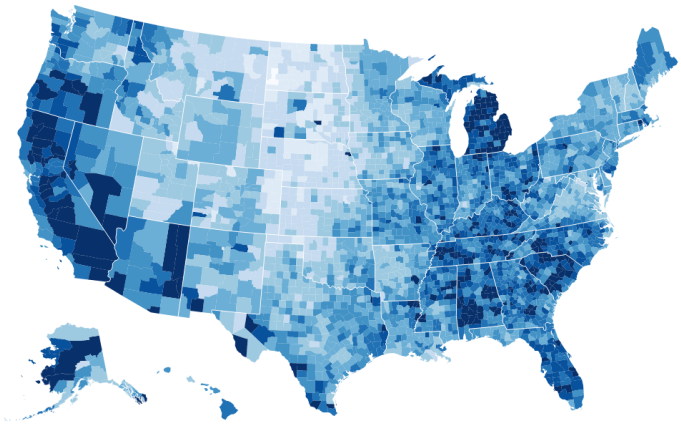

Figure 12. USA unemployment rate using choropleth layout (Source: https://d3js.org/)

Figure 13. Graph generated by Gephi visualizing the number of tweets (Source: DH2014 day by day conference; http://www.martingrandjean.ch/dataviz-digital-humanities-twitter-dh2014/)

creates inline and large charts with ease. Overall, D3 is a tool which helps users to build data visualization frameworks too.

D3 is a big data visualization tool, which has its own advantages and disadvantages. Since all the tools are applied in specific domains, we can not expect perfection. In our case, D3 uses SVG instead of canvas. This is a technique which prevents the support of billions of points. But on the other hand it has more advantages such as unlimited power and control. It is supported on all advanced browsers. Data-Driven Document offers a good API documentation too. Users can find examples and tutorials via forums where you can find support for many questions and learn more about the visualization techniques in this field. Last but not least, D3 helps the user to operate complicated math operations.

Below, we will give one simple example and show how visualization of big data works on D3. Choropleth is a project illustrated in Figure 12. It encodes unemployment rates from 2008 with a scale ranging from 0 to 15 percent. A threshold scale is a useful alternative for coloring arbitrary ranges. The illustration shows that visualization has the power of helping understand the content easily. By this map visualization technique using D3, everyone can realize the regions and states where unemployment rate is relatively high and compare it with other regions. We decided to analyze USA unemployment rate by D3, knowing that most students are conscious that central states of USA have a lower employment rate than west or east regions. This is clearly visualized in the map. Analyzing huge amount of data which are correlated to each other, we can conclude that unemployment might lead to higher rate of crime, worse education or less happiness. Visualization is meaningful not only when it narrates issues, but also when it shows correlation and possible causation among events and topics.

– Gephi
Gephi is the second visualization tool we will mention in this section. It is designed for visualizing complex and huge datasets. It covers specific domains such as networks, graphs and systems. Gephi provides interactive visualization and is open source.
We will briefly explain the graph definition and give some examples how Gephi helps on graph visualization. A graph consists of two types of primitive elements: nodes (vertices) and links (edges). A subgraph of a graph G is a graph with nodes and links which are subsets of G. The size of a node depends on the value of its “degree centrality” (its number of connections). Some other attributes of graphs are centrality measures, which are essential metrics to analyze the position of an actor in a network. They come in many definitions such as degree centrality (number of connections), closeness centrality (closeness to the entire network), betweenness centrality (bridges nodes), eigenvector centrality (connection to well-connected nodes). Most of computer science students know the definition of these elements, since social network analysis is studied widely in every university nowadays. We are going to show how Gephi visualizes big datasets as graphs.

In the following we will analyze an example from Twitter. The graph in Figure 13 represents all the mentions contained in tweets (a tie connects two users when one mentions the other at least once in a message). The size of the circles indicates the number of tweets sent. The intensity of the color depends on the number of incoming mentions (in degree): the more a user is mentioned, the clearer the color (from blue to white between 0-100 mentions, and white for more). This simple network is reduced for visualization purposes into some specific users.

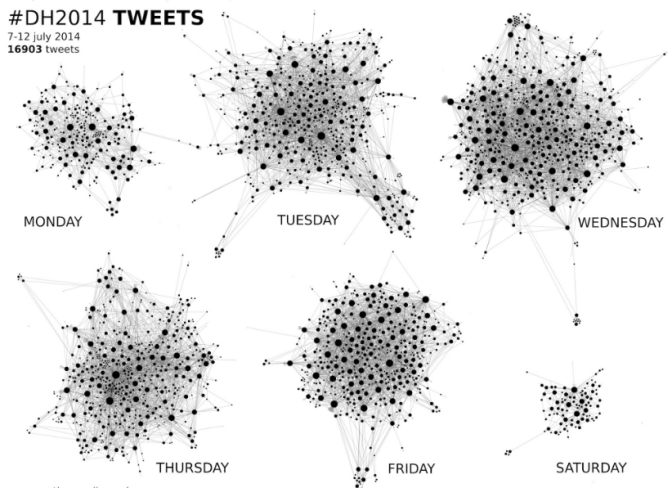

Figure 14. Graph generated by Gephi visualizing number of tweet global signatures during a week (Source: DH2014 day by day conference; http://www.martingrandjean.ch/dataviz-digital-humanities-twitter-dh2014/)

In real world, the number of tweets and mentions is extraordinary huge, and visualization becomes even more challenging. In our case we have visualized a relatively small number of tweets. Still, the networks seems very dense, even though a strong force-directed algorithm is applied. In this case, the communities are not separated from the main cluster. This happens because probably all accounts that follow a given discipline with hashtag, also responded to all participants rather than to a particular number of accounts. The output is displayed as a single cluster. But what we can analyze from visualization of tweets in the graph is that the most mentioned people are not necessarily the most active ones. There is a correlation between them, but it does not imply causation. If a member live-tweets from a conference or meeting, it is not necessary that all the followers respond to all tweets. Overall, we can observe that the list of users who have been mentioned more than 70 times does not correspond to number of users who are most active.

The mass of tweets is unevenly distributed in time, something that the complete graph does not reveal. For these features, Gephi is used by Yannick Rochat in DH2014 day by day conference[4] to produce a more visual and instinctive approach that shows global signatures of six daily graph. It easily distinguished that the number of signatures drastically decreases on Saturday, while its peak is on Wednesday. The graph is visualized in Figure 14.

FUTURE WORK

Next generations of data visualization are related to new technologies, starting from emerging sources of intelligence to evolving cognitive frameworks. The visualization will evolve from art to science, by playing a key role in opening new horizons. Data visualization will play a big impact in society and organizations. The internet has transformed the way we can visualize information through a better understanding of networks.

The future is related to the term Internet of Things (IoT). Tens of billions of devices and data will be connected to the internet in the next years. The IoT will provide unprecedented insight into what's happening around us. Interconnected data streams will help us to better understand consumer demand, improve safety elements, increase operational efficiency, and further develop science.

Data Visualization is expected to have an important role in medicine too. The vast majority of data visualization nowadays is two-dimensional. Until now, spatiotemporal problems were analyzed either from the time-based approach (for instance, evolution of oil prices over the time), or from a spatial perspective (for instance, a map of oil prices in one region). These dimensions were never studied simultaneously from both perspectives. However, research is working on fast rhythms especially in creative use of colors and size. The integration of space and time in computer graphics can create the future models. Some results are already successfully implemented. Neuroscientists Emmanuelle Tognoli and Scott Kelso were awarded a patent called the five dimensional (5D) cholorimetric technique [5]. The method is designed to interpret large data sets with complex spatiotemporal patterns. In Figure 15, there is a illustration of the model explained above, which provides a dynamic view of brain activity.

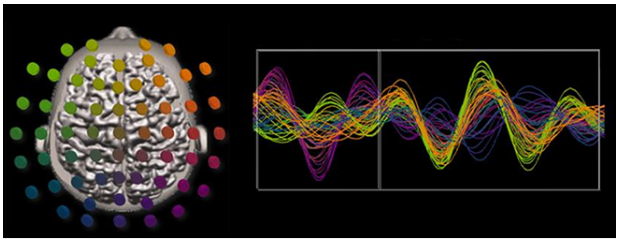

Figure 15. The 5-D cholorimetric technique (Source: Florida Atlantic University, Center for Complex Systems and Brain Sciences; http://www.ccs.fau.edu/hbbl3/?p=1013)

As shown in Figure 16, Microsoft's holograph is another interactive 3D platform which can render static and dynamic statistics and manipulate complex images above or below a plane for more natural exploration and manipulation of complex data. The future will help users to actually interact with it as commented by team members Curtis Wong and David Brown posted on Microsoft News[6].

Soon users will expect every visualizing graph, map or chart to be interactive. The future must offer users the opportunity to switch between different types of visualization and watch

[4]http://www.martingrandjean.ch/dataviz-digital-humanities-twitter-dh2014/

[5]http://www.analytics-magazine.org/january-february-2015/1196-data-visualization-the-future-of-data-visualization

[6]http://mspoweruser.com/microsoft-research-talks-about-holograph-an-interactive-3-d-data-visualization-research-platform/

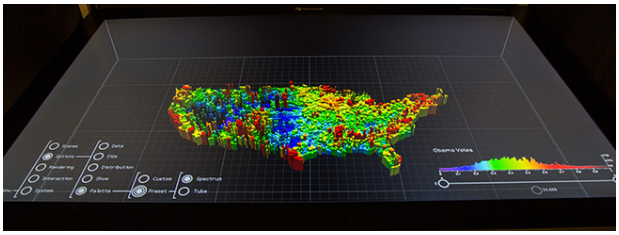

Figure 16. Future USA rendered using Microsoft's holograph (Source: Microsoft; http://microsoft-news.com/microsoft-research-talks-about-holograph-an-interactive-3-d-data-visualization-research-platform/)

animations during the entire time. The challenge is to provide interactivity without complicating the user interface of data visualization.

Another feature developed lately for data visualization is storytelling. Data visualization is empty of meaning without a story. Stories are used as complement for data visualization. Eventually, tools will be deigned to simultaneously automate data interpretation and data visualization. On the other side, the future of storytelling is virtual reality. We hypothesize that in the future visual perception systems of humans will be optimized to interact in three dimensions. Data visualization will no longer be constrained in flat screens.
As the world becomes increasingly interconnected and interdependent, opportunities to generate information through data visualization will only increase. Advanced technologies in multidimensional visualization will allow us to more effectively synthesize and explore spatiotemporal conditions. It is important to create real-time visualization and dynamic systems containing structural dependencies.

CONCLUSION

In this paper we showed various domains where visualization is mainly used and explained why it is crucial nowadays. Furthermore we have analyzed different approaches and provided our own views on how they succeed or what they lack. We presented our motivation why our interest lies beyond data analysis, statistics, or mathematics. Our aim is to present the future of science which is deeply related to visualization. Visualization has the power to make people of different backgrounds understand and appreciate the meaning of the content at a glance.

Overall, visualization needs to present conclusions rather than artistic masterpieces. We have to stay conscious that not all the people understand the data as much as a scientist does. Our focus was to show how visualization helps a broad set of people: starting from students who can use it to understand their study materials, to scientists and professors who might need visualization for experimental and research purposes. This is the power of visualization: making all kind of people with different knowledge and backgrounds understand science and moreover benefit from it. Consequently, our intention was to present visualization as a "tool" which presents clearly valuable information. Visualizing numbers, data, statistics, theorems, and many scientific topics must be considered as an art. It is a difficult field to invest in because it must not represent an interface of abstract puzzles containing "scary" numbers and purely data, but valid information, knowledge and wisdom.
As we explained in our paper, there are many components, attributes, techniques and skills that must interact with each other to visualize valid information. Visualization can harm further studies if information retrieved from it is incomplete or lacks precision. This is another strong point why we need abundant budget to visualize especially big data. Powerful machines and technically expert users must pay attention to the final result.

From our findings, we concluded that data visualization is a field where researchers can test their ideas. We mentioned the tools that are mainly used today to visualize images in specific domains. We concluded that even visualization tools are various, dependent on the field of study. Some of them focus on the interface, some other on data retrieving process or reduction. As data visualization is likely to grow in importance, it is in our interest to continue monitoring data visualization trends. The world is becoming increasingly interconnected and interdependent, consequently the need to generate value through data visualization will only increase.

REFERENCES

1. A. Kadadi, X. Dai, A. Challenges and Opportunities with Big Data Visualization. *MEDES '15: Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems* (2015), 169 – 173.

2. Al-Barakati, N. M., and Al-Aama, A. Y. The effect of visualizing roles of variables on student performance in an introductory programming course. *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education - ITiCSE '09 41*, 3 (2009), 228.

3. Bittinger, M. Calculus and its Applications(Eighth Edition).

4. Ferrer-Mico, T., Prats-Fernàndez, M. À., and Redo-Sanchez, A. Impact of Scratch Programming on Students' Understanding of Their Own Learning Process. *4th WORLD CONFERENCE ON EDUCATIONAL SCIENCES (WCES-2012) 02-05 46* (2012), 1219–1223.

5. Hansen, S., Narayanan, N. H., and Hegarty, M. Designing educationally effective algorithm visualizations. *Journal of Visual Languages & Computing 13*, 3 (2002), 291–317.

6. Hashimoto. Teaching Materials for Business Mathematics. *Web Publication of Visual Teaching Materials for Business Mathematics - Marshallian Demand Function as a Constrained Optimizaiton Problem* (2012), 1 – 4.

7. Hughes-Hallett, A. G. Applied Calculus (Second Edition).

8. information is beautiful. `http://www.informationisbeautiful.net/visualizations/million-lines-of-code/`, 2015 (accessed 14 June 2016).

9. Kandel, S., Parikh, R., Paepcke, A., Hellerstein, J. M., and Heer, J. Profiler : Integrated Statistical Analysis and Visualization for Data Quality Assessment. *Proceedings of Advanced Visual Interfaces, AVI* (2012), 547–554.

10. Kasurinen, J., Purmonen, M., and Nikula, U. A Study of Visualization in Introductory Programming. *Ppig '08*, Winslow 1996 (2008), 181–194.

11. Keown, L. Virtual 3d worlds for enhanced software visualization. *Master's thesis, University of Canterbury, Department of Computer Science* (2000).

12. Lawrence, A. W., Badre, A. M., and Stasko, J. T. Empirically evaluating the use of animations to teach algorithms. In *Visual Languages, 1994. Proceedings., IEEE Symposium on* (Oct 1994), 48–54.

13. Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., and Silver, J. Scratch. *Communications of the ACM 52*, 11 (2009), 60.

14. Sensalire, M., Ogao, P., and Telea, A. Evaluation of software visualization tools: Lessons learned. *2009 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis* (2009), 19–26.

15. Stasko, J. T., Brown, M. H., and Price, B. A., Eds. *Software Visualization*. MIT Press, Cambridge, MA, USA, 1997.

16. Tavel. Modeling and Simulation Design. *AK Peters Ltd., Natick, MA* (2007).

17. Wettel, R., and Lanza, M. Program Comprehension through Software Habitability. In *15th IEEE International Conference on Program Comprehension (ICPC '07)*, IEEE (jun 2007), 231–240.

18. Wettel, R., and Lanza, M. Visualizing software systems as cities. In *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis* (June 2007), 92–99.

19. Wettel, R., and Lanza, M. CodeCity: 3D visualization of large-scale software. *Companion of the 13th international conference on Software engineering - ICSE Companion '08* (2008), 921.

20. Wettel, R., Lanza, M., and Robbes, R. Software systems as cities: a controlled experiment. *2011 33rd International Conference on Software Engineering (ICSE)* (2011), 551–560.

21. Y. Shirota, T. Yutaka, T. N. M. M. Visually Do Statistics for Business Persons: Visual Materials from Regression to Black-Sholes Model. *Proceedings of the 8th International Symposium on Visual Information Communication and Interaction* (2015).