On the Impact of the Medium in the Effectiveness of 3D Software Visualizations

Leonel Merino*, Johannes Fuchs†, Michael Hund†, Craig Anslow‡, Mohammad Ghafari*, Oscar Nierstrasz*
Michael Behrisch†, Daniel A. Keim†

*Software Composition Group University of Bern, Switzerland
†Data Analysis and Visualization Group University of Konstanz, Germany
‡School of Engineering and Computer Science Victoria University of Wellington, New Zealand

***Abstract*—Many visualizations have proven to be effective in supporting various software related tasks. Although multiple media can be used to display a visualization, the standard computer screen is used the most. We hypothesize that the medium has a role in their effectiveness. We investigate our hypothesis by conducting a controlled user experiment. In the experiment we focus on the 3D city visualization technique used for software comprehension tasks. We deploy 3D city visualizations across a standard computer screen (SCS), an immersive 3D environment (I3D), and a physical 3D printed model (P3D). We asked twenty-seven participants (whom we divided in three groups for each medium) to visualize software systems of various sizes, solve a set of uniform comprehension tasks, and complete a questionnaire. We measured the effectiveness of visualizations in terms of performance, recollection, and user experience. We found that I3D has the most positive impact in the effectiveness of visualizations. I3D not only boosts their performance, but also promotes the best recollection and a good user experience. Developers who visualize systems based on P3D exhibit a good performance, the least recollection and a moderate experience. Finally, developers who use SCS exhibit the best performance, but they remember fewer details of visualizations and undergo the best user experience.**

I. Introduction

Many software visualizations have been proposed to support developers in tasks related to various software concerns [1]. When designing visualizations, multiple attributes must be taken into account such as the supported *task* (*e.g.*, software comprehension) and the visualization *technique* (*e.g.*, 3D software cities). Amongst these attributes there is also the display *medium* (*e.g.*, computer screen) on which visualizations are designed to be rendered.

The medium has been considered as an attribute in foundational software visualization taxonomies. Roman and Cox [2] identified new capabilities offered by emerging computer-based visualizations as opposed to traditional visualizations in paper. Price *et al.* [3] observed that while computer-based visualizations can be designed for one medium, they can often be transfered to another. A decade later, Maletic *et al.* [4] envisioned a future in which software visualizations would take advantage of multiple media.

In a previous study [5], we characterized software visualizations using the medium amongst other attributes. Amongst other insights we found that the standard computer screen **(SCS)** remains the most frequently used medium to render software visualizations [5]. Other media used in a few software visualizations were immersive 3D environments **(I3D)** [6], physical 3D printed models **(P3D)** [7], large multi-touch tables [8], and wall-displays [9].

Figure 1: Participants visualize software cities for software comprehension tasks using various media. We evaluate how the effectiveness is affected by the medium: an immersive 3D environment (top), a physical 3D printed model (middle), and a standard computer screen (bottom).

Nevertheless, the impact of the medium amongst these visualizations is not clear. In the past, the effectiveness of the medium is studied either in isolation or from a single perspective.

We investigate to what degree the choice of a medium affects the effectiveness of visualizations. We consider effective visualizations to be those that excel at: (1) performance (*i.e.*, completion time and correctness), (2) recollection (*i.e.*, recollection of recent events), and (3) user experience (*i.e.*, feelings and difficulties). Consequently, we formulated the following research questions:

RQ.1) How does using different media for software visualization affect *completion time and correctness?*

RQ.2) How does using different media for software visualization affect *recollection of recent events?*

RQ.3) How does using different media for software visualization affect *user experience?*

We investigated these questions via a controlled user experiment. In the experiment we focused on *software comprehension*. That is, the cognitive process in which developers learn about a software artifact to accomplish a task [10], and the *3D city visualization technique*, which *i*) has proven to be effective to support software comprehension tasks [11], *ii*) is available for various media [7], [12], and *iii*) is easily transferable from one medium to another.

We selected media used in software visualizations that take different approaches to interaction (*i.e.*, SCS, I3D, P3D) (shown in Figure 1). We formulated a set of nine software comprehension tasks inspired by those used in previous studies [9], [11], [13], and we selected a set of open-source software subject systems of various sizes. For each medium we conducted interviews with between-subject groups of nine developers to collect data that helped us to answer our research questions. We found that I3D has the most positive impact in the effectiveness of visualizations. I3D not only boosts their performance, but also promotes the best recollection and a good user experience. Developers who visualize systems based on P3D exhibit a good performance, the poorest recollection and a moderate experience. Finally, developers who use SCS exhibit the best performance, but they remember fewer details of visualizations and undergo the best user experience.

The main contribution of the paper is an evaluation and discussion of the impact of the medium in the effectiveness of 3D software visualizations. We discuss the need of the explicit inclusion of the medium and technique as properties for benchmarks that evaluate software visualizations. Finally, we also contribute to the reproducibility of our research by making the design and data set of the study available on request.

II. Related Work

The medium has been identified as an important characteristic in the software visualization community. Price *et al.* [3] proposed a software visualization taxonomy that includes the medium as a dimension. They observed that a primary target medium must be identified for visualizations that eventually could be transfered across multiple media. Maletic *et al.* [4] proposed a complementary taxonomy that also includes the medium as one of the five dimensions that characterize software visualizations. Although these foundational taxonomies have been present in the software visualization community, the medium has not been a main concern among most proposed visualizations.

We now elaborate on related work of the 3D software visualization technique that we use in our experiment.

A review of 3D software visualization was presented by Teyseyre and Campo [14]. They classified twenty-two visualization tools based on their expected audience, data source, presentation, interaction, evaluation, and framework used. They observed that the medium plays a key role in the effectiveness of software visualizations. However, all tools included in the overview were designed for one medium (*i.e.*, SCS), and consequently they did not include it as a classification criterion.

3D city visualizations have been proposed extensively to support software comprehension. Wettel and Lanza [15] stated that software cities provide developers a physical space with strong orientation points. Knight and Munro [16], proposed a visualization that implements the city metaphor to support program comprehension. They observed that virtual reality provides developers orientation when exploring code artifacts. Panas *et al.* [17] proposed visualization to support multiple comprehension tasks using a single-view. However, none of them elaborated on why they decided to use the SCS medium.

Software visualization based on I3D is not new. Maletic *et al.* [18] proposed an immersive software visualization object-oriented system for comprehension using a *CAVE* setup. Recently, Fittkau *et al.* [12] evaluated the visualization of software cities using the *Oculus Rift* for software comprehension tasks. However, none of them elaborate on the grounds that supported their selected medium.

A few visualizations have proposed P3D as their medium. Huron *et al.* [19] proposed constructive visualization as a paradigm for simple creation of flexible and dynamic visualizations (*e.g.*, using Lego bricks). Fittkau *et al.* [7] used a physical 3D printed model of a software city that they compared to visualization in a computer screen. Their evaluation showed little differences between the performance of visualizations displayed on SCS versus P3D. In this work, we study two systems of different size. We not only compare P3D versus SCS, but include I3D. Finally, besides evaluating performance, we also evaluate recollection and user experience, since we believe that software comprehension can benefit from both.

In summary, we observe that even though research in software visualization has spanned various media, little has been done to support developers to choose the most effective medium. Therefore, our interest is to study the impact of the medium in the effectiveness of 3D software visualizations.

III. Controlled User Experiment

We performed a controlled user experiment that evaluates the impact of the medium in the effectiveness of 3D software cities for comprehension tasks. Now we elaborate on the design of our experiment.

A. Experiment Design

The purpose of our experiment is to evaluate the impact of the *medium* (independent variable) in the effectiveness of software visualizations by comparing *performance*, *recollection* and *experience* (dependent variables). The *performance* of participants was measured in terms of completion time and accuracy. To measure *recollection*, we asked participants in the last part of the session to draw what they remembered of the visualization of the second system (approximately twenty minutes after). Finally, to measure *user experience* *i*) during the visualization of each system participants were asked to score the difficulty of the tasks, and *ii*) at the end of the visualization of each system participants were asked to identify their top ten experienced feelings (sorted by intensity).

We decided to use between-subject groups of nine participants. That is, the participants of each group visualize the three systems (listed in Table II) one-by-one solving the nine tasks (listed in Table I) in one medium. We ran four pilot studies and analyzed their outcome. We tried various configurations of the parameters of the visualization technique and selected the one that performed better for navigation and comparison. We fine-tuned the tasks, so the experiment would last around one hour (to avoid fatigue).

When designing our experiment, we noticed that there is a need for a standard protocol to compare evaluations of software visualizations. We observed that Maletic and Marcus [20] issued a call-for-benchmarks towards standardizing the evaluation of software visualizations. They proposed four properties that characterize visualizations for benchmarks: *task*, *data set*, *evaluation* and *interaction*. We observe that a developer willing to adopt a visualization technique that is available in various media cannot compare the results of isolated evaluations of visualizations that not only differ in the technique but also in the display medium, thus possibly leading to misleading results. Thus, the need of a standard protocol to compare evaluations of software visualizations that includes the medium explicitly. Consequently, we propose to add explicitly two properties to these benchmarks: *medium* and visualization *technique*. In this way, benchmarks support not only researchers who compare new visualization techniques, but also those who evaluate visualizations across multiple media (as is the goal of this paper).

Extended Benchmark Properties: We first describe our proposal for the two new added properties (*i.e.*, medium and technique) and then for each of the four original properties (*i.e.*, interaction, task, and data set) of benchmarks.

Medium. Amongst the media used in software visualizations we find immersive 3D environments, physical 3D models, wall-displays, multi-touch tables, and standard computer screens [5]. Since the last three use the screen to display visualizations, we propose the media used in the following setups to conduct the experiment:

- *i*) *Standard Computer Screen (SCS)*. We used an Apple MacBook Pro with a resolution of 1440 x 900 pixels. The visualizations were provided by the CodeCity [1] implementation for Moose 5 on OSX.[2]
- *ii*) *Immersive 3D environment (I3D)*. We used an HTC Vive VR Headset with a 2160 x 1200 combined resolution, 90 Hz refresh rate and 110° field of view. We implemented a custom visualization using Unity 5.5 based on models of the cities exported from CodeCity.
- *iii*) *Physical 3D model (P3D)*. We used a Form 2 3D printer by formlabs[3] based on stereolithography (SLA) technology. To implement the visualizations, we exported them from their implementation in Unity (used for I3D) to the Stereo Lithography (STL) format required by the printer using the *pb_Stl* [4] library.

Technique. In a previous study [5] we identified sixty-four visualization tools that implement various visualization techniques. We selected from them a visualization technique based on the following criteria: (C1) proven effective for software comprehension tasks, (C2) suitable for the capabilities of the selected media, and (C3) implementations or source code are available. We focused on the most restrictive criterion, namely C2. In the process of selecting a suitable technique we rejected visualizations that: *(i)* support tasks that do not focus on software comprehension, such as Vizz3D [21], or *(ii)* neither provide implementations for all media, such as TraceCrawler [22], nor make their source code publicly available, such as MetricView [23]. Instead, we observed that 3D city visualizations fulfill all these criteria. Firstly, software cities have proven effective to solve software comprehension tasks in terms of performance [11], recollection [24], and user experience [13] (C1). Secondly, they have proven to be suitable for SCS [25], I3D [12], and P3D [7] (C2). Finally, even though we did not find implementations available for all media, the simple design of software cities based on colored cubes and the availability of source code enables their implementation to be easily transferred from one medium to another (C3).

Figure 2 shows *CodeCity* [11], a well-known implementation of 3D software cities for SCS. In this visualization metaphor, buildings in the city represent classes in the software. Contiguous buildings in a district represent the classes that belong to a package. The visualization allows developers to analyze software metrics and identify potential design problems such as *god classes*. We configure the visualization in such a way that the height of a building encodes the number of attributes (NOA) of the represented

[1] http://smalltalkhub.com/#!/~RichardWettel/CodeCity
[2] http://www.moosetechnology.org/
[3] https://formlabs.com/3d-printers/form-2/
[4] https://github.com/karl-/pb_Stl

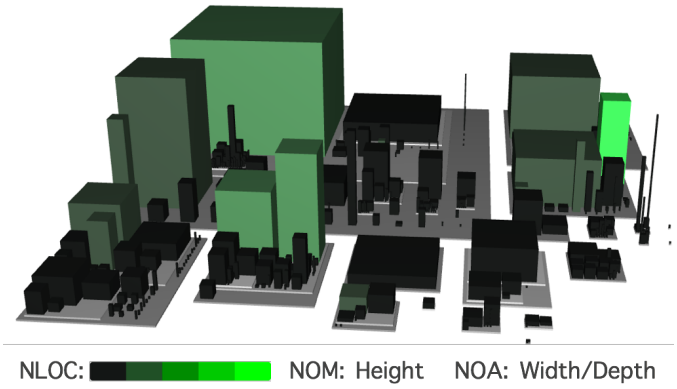


Figure 2: Freemind 2.0.9 is the medium size system used in the experiment. The system is visualized as a software city where buildings represent the classes of the system, so as districts do to packages. Three software metrics are mapped to attributes of buildings: number of lines of code to the color, number of methods to the hight, and number of attributes to the width/depth.

class, the size of the square base of a building represents the number of methods (NOM), and the color encodes the number of lines of code (NLOC). We use a linear scale of five different tones of green as proposed by the ColorCAT [26] tool for visualizations that support comparison tasks on continuous data. The brighter the color, the higher the value of the metric.

Interaction. We confined the interaction to those which are in common in all media. Consequently, since P3D does not support selection, the interaction provided to participants in all media was limited to navigation (*e.g.*, rotate, pan, zoom).

Tasks. We assume developers who want to contribute to an open-source object-oriented software system need to collect class candidates for potential refactoring. To accomplish this high-level task, they usually define nine specific sub-tasks (listed in Table I) that they have to solve. The visualization helps developers to obtain an overview of the whole software system and spot refactoring candidates.

When developers obtain an overview of a software system, they are able to (1) spot outliers, (2) detect patterns, and (3) quantify elements [27]. Although some of these tasks can be addressed faster and eventually with more accuracy by other approaches, visualizations enable developers to combine all of them at once. We were inspired by a previous evaluation of CodeCity [11] to design our tasks. We focused on two criteria to select the tasks: (i) they can be solved in a reasonable amount of time (*e.g.*, < 5 minutes), and (ii) the only interaction needed to solve them is navigation. For each medium (*i.e.*, SCS, I3D, P3D) a different group of participants visualize one at a time the systems (shown in Table II) and solve the tasks (shown in Table I). The tasks are grouped by themes. Tasks *T1-T3* require metric analysis to *find outliers*. Tasks *T4-T6* concern the detection of potential design problems by *identifying visual patterns*. Finally, Tasks *T7-T9* concern *location and quantification*.

Data set. We looked for a collection of real-world open source software systems of diverse size. We observed that the *Qualitas Corpus* [28] fulfills these criteria. We selected three systems (from the Qualitas Corpus) of various sizes that have been used extensively in other studies (shown in Table II).

B. Hypothesis

We hypothesized that the most common medium used in software visualizations, the standard computer screen, is an effective medium. Since the computer display is the main medium used during development, we envisage that interacting with visualizations displayed on the computer screen with a mouse and keyboard will not pose difficulties. We therefore conjecture that visualizations using this medium will excel in performance (RQ.1) and user experience (RQ.3), but is not clear to us how this medium encourages user recollection (RQ.2). We want to know whether media may hinder the performance of visualizations, and if so, to what degree. We ask whether participants who use I3D or P3D might remember more details of the visualized software than participants who use a more conventional medium such as the computer display. We observe that P3D as opposed to I3D and SCS involves two senses: sight and touch. We conjecture that this characteristic promotes recollection. We also hypothesize that non-traditional media such as I3D and P3D might boost user experience. We consequently define the following null hypotheses:

[H1] When visualizing software as cities for comprehension, the *time to complete* tasks and the *accuracy* of developers is equal across various media (RQ.1).

[H2] When visualizing software as cities for a software comprehension task, the *recollection* of developers is equal across various media (RQ.2).

[H3] When visualizing software as cities for a software comprehension task, the user *experience* of developers is equal across various media (RQ.3).

C. Participants

One important goal for between-subjects groups of participants (*i.e.*, each participant visualizes all systems using a single medium) is that groups have to be as similar as possible [29]. We selected participants of the groups to have a similar distribution of gender and education level. Each group was formed of one post-doc researcher, five PhD students and three bachelor/master students in computer science. The average age was 28.72 ± 1.43 years, and the average experience as a developer was 8.08 ± 0.77 years. Participants were not paid. They were invited and freely opted to participate in the study. Thirteen out of the twenty-seven participants were recruited from the University of Konstanz in Germany. The rest were recruited from the University of Bern in Switzerland. The interviews were conducted from February 2017 to March 2017.

Table I: Software comprehension tasks that participants have to solve.

Theme	Rationale	Id	Task
Find Outliers	Classes that exhibit extrema values of software metrics might lead to problem detection and might represent a good candidate for refactoring	T1	Find the three classes with the highest NOM
		T2	Find the three classes with the highest NOA
		T3	Find the three classes with the highest NLOC. If two are in the same range select the one with the lowest NLOC
Identify Patterns	The relationship among values of software metrics help developers to identify design problems. The ratio among the metric's values produce a pattern among the visual representation of entities	T4	Locate the best candidate for the *god class* smell (hint: god classes contain many methods with many lines of code)
		T5	Locate the best candidate for the *data* class pattern (hint: a data class has high NOA, and low NOM and NLOC)
		T6	Locate the longest *facade* class (hint: facade classes have high NOM, and low NOA and NLOC)
Locate and Quantify	Help developers to prioritize what is the most critical, *e.g.*, a package that contains many *god* classes might be a good candidate for refactoring	T7	Locate the package with the highest number of classes such that NLOC in the classes are the least
		T8	Determine the total number of packages this system has
		T9	Estimate the total number of classes this system has

Table II: Systems used in the experiment. Participants visualized *Axion* for the training session. *Freemind* and *Azureus* were used for evaluation.

System	Version	# KLOC	# Classes	# Pkgs.	Size
Axion	1.0-M2	23	223	27	Small
Freemind	2.0.9	56	881	108	Medium
Azureus	4.8.1.2	646	6619	560	Large

D. Procedure

The experiment was conducted in two locations: one at the University of Konstanz and the other at the University of Bern. The rooms at both locations were of similar size (*i.e.*, 5 m x 5 m approximately) and lighting. During the study only the participant and the experimenter were in the room. The same experimenter conducted the experiment at both locations. A different setup was defined for each medium: for I3D, participants wore a headset and held a controller. Participants interacted with the visualization by walking and crouching. The tasks were displayed within the visualization. A legend with the encoding of the visualization was visible at all times. Participants used the controller to specify their answers to the tasks; SCS participants sat in a chair in front of the computer screen. They interacted with the visualization through the mouse and keyboard. The tasks were handed to them printed on paper. A legend with the encoding of the visualization was visible on a separate screen at all times. Finally, P3D participants sat in front of a desk on which the model was placed. They interacted with the model by holding, rotating and moving it with their hands. The tasks were also handed to them printed on paper. A legend with the encoding of the visualization was visible on a computer screen at all times. Participants had a wooden stick to point in the model to their answers.

We started the experiment by reading an introduction to explain participants the problem domain, the encoding of the visualization, and what they were expected to perform during the experiment. Firstly, participants had a training session where they viewed a visualization of the *Axion* system. They were asked to read one-by-one the tasks aloud, then they had to describe the visual pattern to solve the task, and finally they pointed to the element that corresponded to their answer. Secondly, participants visualized *Freemind* and solved the tasks one at a time as they did during the training. This time, when they gave their answer to each of the tasks, we asked them how difficult they found the task. We asked them to score their answer on a 5-step Likert scale [30]. When they finished all the tasks we asked them to approach a table where we previously placed 270 labels. Each label contained a word that represents a feeling. We placed positive feelings on the left side of the table and negative ones on the right. Labels were organized into eight groups of positive feelings and also eight of negative ones. Participants were asked to collect ten feelings, experienced during the previous visualization, from the table (without any restriction) and to sort them according to their intensity. Thirdly, participants visualized *Azureus* and repeated the same steps: solve the tasks, score their difficulty and identify the feelings experienced during the visualization. Lastly, to evaluate the recollection of near-time memories, participants were asked to approach a whiteboard and to draw what they remembered from the visualization of *Freemind* (approximately twenty minutes after they finished with the visualization).

E. Data Collection

We collected several data points during the experiment. We *(i)* video recorded participants as they navigated visualizations (*e.g.*, moving across the room in I3D) as well as the view they obtained of the visualization itself (*e.g.*, screen record in SCS), *(ii)* video recorded participants drawing the recollected memories of Freemind, and *(iii)* took pictures of the selected labels that described their experienced feelings during visualizations. We edited the videos to produce single records that contain the whole interview of each participant. We watched each of these records to measure and double-check completion time and accuracy, as well to identify recurrent concepts for qualitative analysis (observed emergent codes).

IV. Results

We performed a statistical analysis of the collected data. To analyze performance, we observed that the results of accuracy did not follow a normal distribution. We then analyzed accuracy using Kruskal-Wallis' test [31]. We also observed that the rest of the dependent variables (*i.e.*, *completion time*, *recollection* and *experience*) satisfy *(i)* independent observations of between-subject groups of participants, *(ii)* homogeneous variances of dependent variables (validated using Lavene's test [32]), and *(iii)* normal distribution of dependent variables (validated using Shapiro-Wilk's test [33]). Accordingly, we used the one-way Analysis Of Variance (ANOVA) to test these hypothesis, followed by Tukey's HSD for comparing differences between groups using a different medium. In either case, we chose a 95% confidence interval (α = .05) to evaluate whether there are statistically significant differences in *H1* performance (shown in Figures 3a and 3b), *H2* recollection (shown in Figure 4), and *H3* experience (shown in Figures 5a and 5b) between visualizations used to solve comprehension tasks among different media.

A. Performance (RQ.1)

We study performance by analyzing: *completion time* and *accuracy*.

1) Completion Time: Firstly, independent of the size of the system, the variation of the time to *identify outliers* (T1-T3) among media was much larger than the variation of the time within each medium (Freemind: $F_{2,78}$ = 8.006, p = .00069; Azureus: $F_{2,78}$ = 4.69, p = .012). Thus, we *reject H1* for tasks T1-T3. We found significant differences between P3D-I3D (Freemind: p = .00089; Azureus: p = .011) and SCS-I3D (Freemind: p = .0092; Azureus: p = .096) but not between SCS-P3D (Freemind: p = .74; Azureus: p = .65). Secondly, in both software systems the variation of the time to *detect patterns* (T4-T6) among media was less than the variation of the time within each medium (Freemind: $F_{2,78}$ = 1.23, p = .30; Azureus: $F_{2,78}$ = 2.27, p = .11). Thus, we cannot *reject H1* for tasks T4-T6. Finally, in Freemind, the variation of the time to *locate and quantify* classes (T7-T9) among media was much larger than the variation of the time within each medium ($F_{2,78}$ = 6.19, p = .0032). Thus, we *reject H1* for tasks T7-T9. We also found significant differences between SCS-I3D (p = .0053) and SCS-P3D (p = .019) but not between P3D-I3D (p = .92). However, in Azureus, the variation of the time among media was less than the variation of the time within each medium ($F_{2,78}$ = 1.65, p = .2). Thus, we cannot *reject H1* for tasks T7-T9.

> Developers who visualize software cities for comprehension require the least time using **P3D** for *all tasks indiscriminately.*

2) Accuracy: Firstly, the variation of the accuracy to *find outliers* (T1-T3) among media was less than the variation of the accuracy within each medium (Freemind: χ^2 = .73, p = .69; Azureus: χ^2 = 5.80, p = .055). Thus, we cannot *reject H1* for tasks T1-T3. Secondly, We also found that the variation of the accuracy to *find patterns* (T4-T6) among media was less than the variation of the accuracy within each medium (Freemind: χ^2 = .95, p = .62; Azureus: χ^2 = 1.77, p = .41). Thus, we also cannot *reject H1* for tasks T4-T6. Finally, we found that independent of the size of the system, the variation of the accuracy to *locate and quantify* classes (T7-T9) among media was less than the variation of the accuracy within each medium (Freemind: χ^2 = 9.20, p = .010; Azureus: χ^2 = 7.79, p = .020). Thus, we cannot *reject H1* for tasks T7-T9.

> Developers who visualize software cities for comprehension are equally accurate to *identify outliers* and *detect patterns* using **SCS**, **I3D**, and **P3D** indiscriminately; and obtain the highest accuracy to *locate and estimate* when using **I3D**.

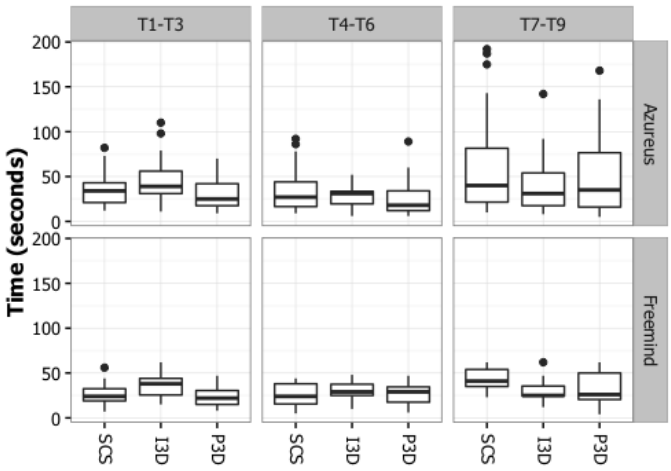
B. Recollection (RQ.2)

During software comprehension developers do not know what information might become relevant to remember. We therefore did not ask participants to remember details of the visualization. Instead, at the end of the interview we asked them to draw on a whiteboard what they remembered from the *Freemind* system (approximately twenty minutes after they finished with the visualization). Most participants said that they did not remember anything. However, after a few seconds they started to remember some details and drew some aspects of the visualizations on the board. We quantitatively analyzed the drawings by measuring two aspects of them (i) amount of used ink, and (ii) number of identified design problems.

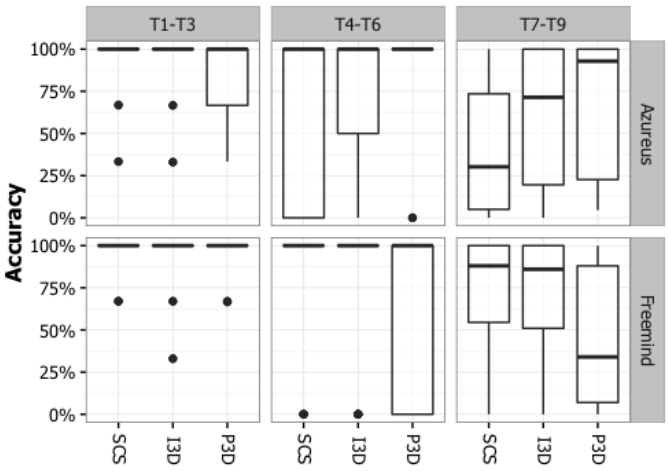
The data-ink ratio of the drawings is a measure of the ink used to depict the recollected data. [34] We analyzed the color statistics of pictures of the drawings using an online color summarizer[5]. We observed that the variation of the recollection among media was much larger than the variation of recollection within each medium ($F_{2,24}$ = 4.82, p = .017). Thus, we *reject H2*. We found significant differences between P3D-I3D (p = .014) but not between SCS-P3D (p = .47) and SCS-I3D (p = .16). We also noticed that most drawings depicted the classes candidates of design problems (*e.g.*, god class, data class, longest facade) that they had to find to solve the tasks. We measured their frequency and report the results in Figure 4. We observed that I3D has the highest recollection, followed by SCS and P3D, and that recollection decreases when visualizing larger systems (*i.e.*, Azureus). We did not find significant variances in the recollection of design problems (p = .25).

> Developers who visualize software cities for comprehension recollect the highest number of memories using **I3D**

[5] http://mkweb.bcgsc.ca/color-summarizer/



(a) Completion time of the participants in the experiment. Box plots are grouped by the theme of tasks (vertically). Rows contain the results that correspond to a different system.



(b) Accuracy of the participants in the experiment. Box plots are grouped by the theme of tasks (vertically). The results of each system are split into rows.

Figure 3: Performance

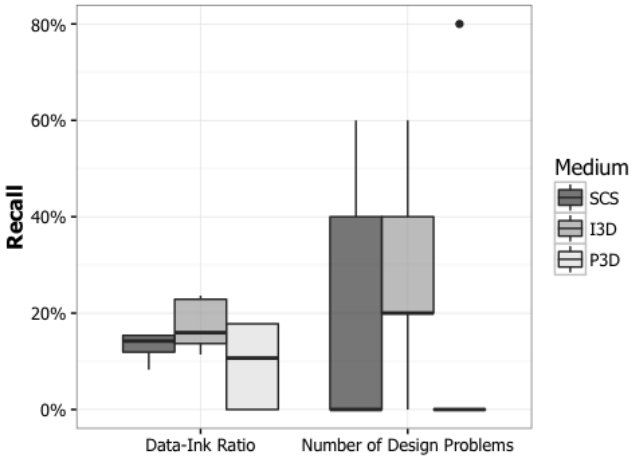


Figure 4: The mean recollection of the five most frequent candidates of design problems found in Freemind (skewers show the standard deviation). One means the five candidates recollected, while zero means none.

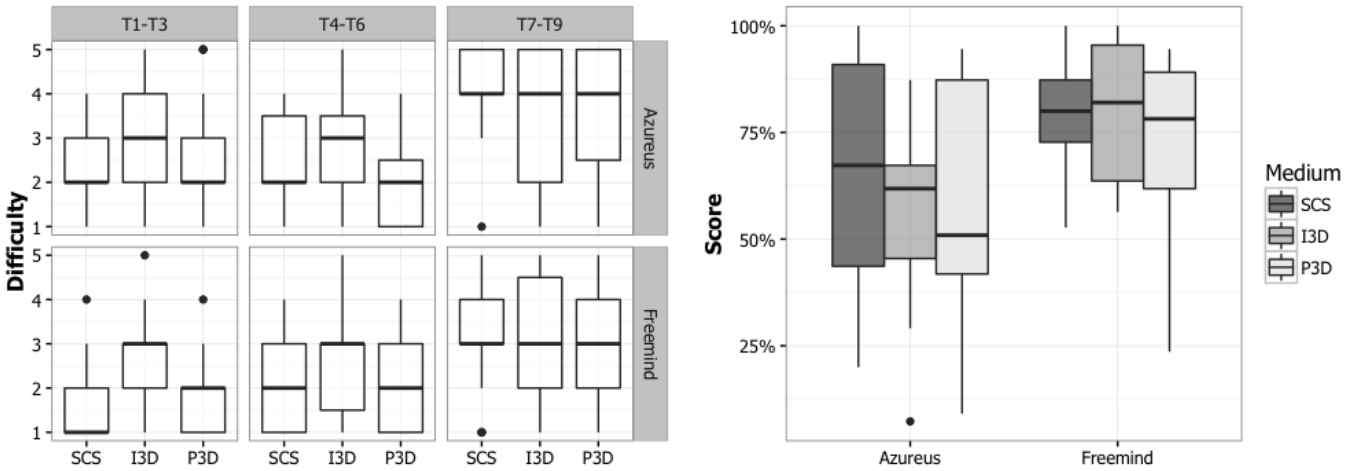
C. User Experience (RQ.3)

We measured two attributes that contribute to user experience: *difficulty* and experienced *feelings*. During the experiment (i) after each task we asked participants to rank the experienced difficulty using a 5-step Likert scale, and (ii) when participants finished all the tasks of one of the systems we asked them to identify their top ten strongest feelings experienced during the visualization.

1) Difficulty: Firstly, independent of the size of the system, the variation of the experienced difficulty to *finding outliers* (T1-T3) among media was much larger than the variation of the difficulty within each medium (Freemind: $F_{2,78}$ = 10.43, p = 9.69e-05; Azureus: $F_{2,78}$ = 3.91, p = .02). Thus, we *reject H3* for tasks T1-T3. In Freemind we also found significant differences between SCS-I3D (p = .00023) and P3D-I3D (p = .0011) but not between SCS-P3D (p = .89). In Azureus we found significant differences only between SCS-I3D (p = .018) but not between SCS-P3D (p = .29) and P3D-I3D (p = .42). Secondly, in Freemind the variation of the experienced difficulty to *finding patterns* (T4-T6) among media was less than the variation of the difficulty within each medium ($F_{2,78}$ = 1.61, p = .21). Thus, we cannot *reject H3* for tasks T4-T6. In Azureus the variation of the experienced difficulty *finding patterns* (T4-T6) among media was much larger than the variation of the difficulty within each medium ($F_{2,78}$ = 3.99, p = .022). Thus, we *reject H3* for tasks T4-T6. In Azureus we found significant differences only between SCS-P3D (p = .037) but not between SCS-I3D (p = .99) and P3D-I3D (p = .051). Finally, independent of the size of the system, the variation of the experienced difficulty to *locate and quantify* classes (T7-T9) among media was less than the variation of the difficulty within each medium (Freemind: $F_{2,78}$ = .31, p = .74; Azureus: $F_{2,78}$ = 1.99, p = .14). Thus, we cannot *reject H3* for tasks T7-T9.

> Developers who visualize software cities for comprehension perceive tasks the least difficult to *identify outliers* using **SCS**; and to *detect patterns* and *locate and estimate* using **P3D**.

2) Feelings: We defined the *score* metric shown in Equation 1 to rank the experience of participants. The score is a weighted sum of the top ten strongest feelings that



(a) Difficulty experienced by participants. Box plots are vertically grouped by the theme of tasks. The overall difficulty is higher in Azureus than in Freemind.

(b) Feelings' score experienced by participants. Bars show the mean results, and skewers show the standard deviation.

Figure 5: User Experience

participants experienced during the visualization of the systems. The score takes into account the intensity of the feeling (*i.e.*, position) and the type of feeling (*i.e.*, positive and negative).

$$score = \sum_{i=1}^{10} i \times type(feeling_i)$$
$$type(feeling) = \begin{cases} 1 & \text{if feeling is positive} \\ -1 & \text{if feeling is negative} \end{cases} \quad (1)$$

Independent of the size of the system, the variation of the score of the experienced feelings of participants among media was less than the variation of the score within each medium (Freemind: $F_{2,78} = .75$, $p = .49$; Azureus: $F_{2,78} = .58$, $p = .57$). Thus, we cannot *reject H3*.

We observed that the highest frequency of positive feelings is offered by SCS, in which users feel *confident*, *certain* and *satisfied* and a few times *frustrated*, *unsure*, and *overwhelmed*. Participants of I3D experienced balanced feelings. Sometimes they felt *interested*, *fascinated* and *optimistic*, and in some others cases they felt *doubtful*, *hesitant*, and *uncertain*. Participants of P3D reported the largest number of negative feelings of which the most frequent words were *hesitant*, *frustrated* and *impulsive*.

Curious and *challenge* are the two most frequent feelings identified among all media. After visualizing *Freemind* (*i.e.*, the medium size system) 67% of participants selected *curious* and 48% *challenge* (41% selected both simultaneously). Then, after participants visualized *Azureus* (*i.e.*, the largest system in the study) 41% of them selected *curious* and 37% *challenge* (19% selected both simultaneously).

> Developers who visualize software cities for comprehension experience the most positive feelings using **SCS** for *large systems*, and using **I3D** for *medium sized systems*.

D. Summary

Firstly, we found significant differences in completion time (*performance*) between SCS-I3D for tasks that involve identifying outliers (*i.e.*, T1-T3) and that require to locate and quantify classes (*i.e.*, T7-T9). We also found significant differences between I3D-P3D (but only for tasks T1-T3) and SCS-P3D (only for tasks T7-T9). Consequently, we reject *H1* and accept the alternative hypothesis. For tasks that involve finding patterns (*i.e.*, T4-T6) we cannot reject the null hypothesis *H1*.

Secondly, the analysis of the data-ink ratio of the recollected memories drawn by participants (*recollection*) shows significant differences of results only between P3D-I3D. We thus reject *H2* in favor of the alternative hypothesis. That is, when visualizing software cities for comprehension tasks the medium affects recollection of recent events.

Finally, we found significant differences between SCS-I3D in the difficulty reported by participants (*user experience*) for tasks that involve identifying outliers (*i.e.*, T1-T3). For tasks that concern finding patterns (*i.e.*, T4-T6) we found significant differences only between SCS-P3D in the visualization of Azureus. Accordingly, we reject *H3* and accept the alternative hypothesis. In the rest of the cases we cannot reject hypothesis *H1*.

V. Discussion

We now present a qualitative analysis of the results. We split the analysis by the concerns that we investigated through our research questions.

A. Performance (RQ.1)

We discuss the completion time and accuracy of participants based on the *theme* of tasks, the *size* of systems and the *medium* used. We also elaborate on the strategies and reflections made by participants.

1) Completion Time: we noticed simple tasks that required little navigation (*i.e.*, finding outliers T1-T3) required the least time when visualizing a system using P3D, followed by SCS and I3D. For tasks that required more navigation (*i.e.*, finding design problems T4-T6) the results were mixed. The least time to solve the tasks in the medium sized system is obtained visualizing by SCS. In contrast, SCS performed badly for the large system, for which the least time to solve the tasks is provided by P3D. One participant who was locating a package (tasks T7) observed that navigation in SCS makes it *"difficult to get to that part of the city"* On the contrary, a participant who used P3D to find the longest facade class (tasks T6) reflected that *"it is very easy to find these [types] of classes"* We observe that not all participants using SCS, who spent a longer time navigating a system, achieved a higher accuracy than participants using other media. A good balance is offered by I3D, for which one participant observes that *"depth helps a lot to identify packages"*.

2) Accuracy: we observed few differences among tasks related to finding outliers and patterns. Answers across all the media used in the experiment were highly accurate. Instead, the results of the tasks related to location and quantification show that participants are more accurate to quantify elements in medium sized systems than in larger ones. In summary, independent of the size of the system, the best accuracy was provided by I3D, followed by SCS and P3D. We observed that the highest accuracy to assess the size of systems (location and quantification tasks T7-T9) was obtained by participants who compared a current visualized system to the one visualized during the training session (*i.e.*, Axion). Besides, participants who spent a longer time analyzing a system provided highly accurate estimations. One of them developed an algorithm that consisted of mentally dividing the city visualization into a number of sections with an approximately similar number of buildings and then multiplying the number of sections by the number of buildings. Interestingly, the result was the most accurate among participants using SCS and the top three across all media.

B. Recollection (RQ.2)

Participants were asked to draw on the whiteboard only what they freely remember from the second visualized system. We did not force them to guess an answer. In fact, a few of them did not draw anything. Among the majority that remembered, their strongest memories were about the classes spotted when solving design problems tasks (*i.e.*, T4-T6). Most participants were unable to build an overview of the whole system, but had scattered memories of parts of it. Sometimes recollected memories were placed in a wrong location. Surprisingly, some of them remembered unexpected aspects of visualizations such as a thin line crossing the top of a building in P3D, a tiny crowded package in I3D. It suggests that recollection of memories might be boosted by allowing users to individualize the visualized systems. A few participants mentioned that they would *"expect a better recollection of memories if the tasks would encourage them to reason about the system as a whole"* Although the amount of details and accuracy of the memories of participants varied, we can observe a trend. Visualization using I3D produced the most detailed and accurate memories, followed by SCS and P3D. Only three participants were unable to draw their recollection of the system who all used P3D.

C. User Experience (RQ.3)

We observed that even before the experiment participants who visualized systems using I3D were very motivated. Participants who used P3D were less interested. Participants who visualized using SCS showed the least interest. A participant who ran the experiment using SCS asked to try the visualization in I3D just for fun. Participants perceived that the difficulty of tasks increased when they moved from the visualization of Freemind (medium size system) to Azureus (large system). Similarly, the same occurred with the number of negative feelings that also increased. We also analyzed distinct feelings that emerged in only one medium but not in the others. We think those feelings represent advantages and disadvantages that a medium impose. Feeling *quiet* is most distinctive advantage of the I3D medium, and feeling *sure* (*i.e.*, certain) does so for SCS. The former might relate to the unique characteristics of being immersive in the visualization, while the latter might reveal the certainty felt by users of traditional computer interfaces. Several distinctive feelings arise when using P3D that also might relate to the nature of the medium. Participants who used P3D positively felt *sensitive* and *touched*, and negatively felt *dissatisfied* and *powerless*. We noticed differences in the reported difficulty of tasks in terms of system's (i) **Size**. Tasks were perceived less difficult in the medium size system (*i.e.*, Freemind) than in the large system (*i.e.*, Azureus); task's (ii) **Theme**. Tasks sorted by themes were perceived as increasing in difficulty. That is, tasks that concern (a) to identify outliers (T1-T3) were the least difficult, (b) to detect patterns (T4-T6) were of moderate difficulty, and (c) to locate and quantify (T7-T9) classes were the most difficult; and, (iii) **Medium**. Participants who used I3D consistently perceived tasks more difficult than participants who used other media. Between SCS and P3D participants had mixed perceptions depending on the type of task. Tasks concerned with identifying outliers (T1-T3)

were perceived more difficult when using P3D, while tasks to detect patterns (T4-T6) were considered more difficult when using SCS.

We observed that even though participants who used I3D found most tasks consistently more difficult than participants using other media, they reported the most positive feelings and seemed happier during the visualization than participants who used other media. In summary, we consider that I3D provided the best overall experience to participants, closely followed by SCS and P3D.

Surprisingly, two of the three main concepts that influence engagement in computer games are the two most frequently selected by participants: curious and challenge [35]. The third concept, which is *fantasy*, defined as "an illusory appearance",[6] is also inherent to visualizations. We observe that software visualizations could benefit from computer game techniques to increase the effectiveness of visualizations.

VI. Limitations

There are five main threats to the validity of our experiment. The first is (i) bias in the selection of groups. To mitigate this we formed similar groups in terms of education level, gender, age and experience in software development. The second threat is (ii) tasks might not be realistic. We reduced this threat by defining types of tasks that have been previously used in other experiments and studies [7], [9], [11]. The third threat is (iii) visualizations in different media might differ. To mitigate this, visualizations were transfered to all media by automatic procedures. Consequently, the position, size of buildings was the same. Although color was automatically transfered to visualizations in I3D and SCS, we manually colored (*i.e.*, painted) visualizations in P3D. We achieved the same color scale by using a dropper. The fourth threat concerns (iv) environmental aspects such as the room, light and experiment length might be different. Although we interviewed participants in two different locations, we chose rooms with similar characteristics (*i.e.*, size, light, level of noise), conducted the experiment following the same checklist, read the same introduction during the tutorial, displayed the same legend of the encoding used in the visualization in a second screen during the whole experiment, and offered to have a break, drinks and snacks to avoid fatigue to all participants. The same experimenter also conducted a pilot experiment with four participants to identify a suitable length for the experiment (approximately one hour), and fine-tune the tasks. The final thread is (v) any given participant did not have the opportunity to compare two or more media. We considered that the *learning effect* would hindered the quality of the results. Instead we opted for between-subject groups of participants. That is, each participant visualized systems using a single medium.

[6] "fantasy | phantasy, n." OED Online. Oxford University Press, March 2017. Web. 6 April 2017.

VII. Conclusion

Many visualizations have proven to be effective in supporting various software related tasks. Although multiple media can be used to display visualizations, most of software visualizations use a standard computer screen. We hypothesize that the medium used to present visualizations has a role in their effectiveness.

We investigated our hypothesis by conducting a controlled user experiment. In the experiment we focused on the 3D city visualization technique that has proved to be effective for software comprehension tasks. We deployed 3D city visualizations across a standard computer screen (SCS), an immersive 3D environment (I3D), and a physical 3D printed model (P3D). For each medium we asked a different group of nine participants to perform a set of nine comprehension tasks and complete a questionnaire. We measured the effectiveness of visualizations in terms of performance (*i.e.*, completion time and correctness), recollection (*i.e.*, recollection of recent events), and user experience (*i.e.*, feelings and difficulties). We found that (i) I3D has the most positive impact in the effectiveness of visualizations. I3D not only boosts their performance, but also promotes the best recollection and a good user experience; (ii) developers who visualize systems based on P3D exhibit a good performance, the least recollection and a moderate experience; and (iii) developers who use SCS exhibit the best performance, but they remember fewer details of visualizations and undergo the best user experience.

The main contribution of the paper is an evaluation and discussion of the impact of the medium in the effectiveness of 3D software visualizations. We discuss the need of the explicit inclusion of the medium and technique as properties for benchmarks that evaluate software visualizations. Finally, we also contribute to the reproducibility of our research by making the design and data set of the study available on request.

In the future we plan to expand this work in several ways. First, we want to investigate the impact of the medium in the effectiveness of visualizations that use other techniques (possibly based in 2D), and secondly, to investigate the impact of media used for collaborative visualization (*e.g.*, wall-displays, multi-touch-tables) in the effectiveness of software visualizations.

Acknowledgments

We gratefully acknowledge the financial support of the Swiss National Science Foundation for the project "Agile Software Analysis" (SNSF project No. 200020-162352, Jan 1, 2016 - Dec. 30, 2018). The authors thank the German Research Foundation (DFG) for financial support within project A03 "Quantification of Visual Analytics Transformations and Mappings" of SFB/Transregio 161. Merino has been partially funded by CONICYT BCH/Doctorado Extranjero 72140330. We thank our participants: the DBVIS in Konstanz, and the SCG in Bern.

REFERENCES

[1] S. Diehl, *Software Visualization*. Berlin Heidelberg: Springer-Verlag, 2007.

[2] G.-C. Roman and K. C. Cox, "A taxonomy of program visualization systems," *Computer*, vol. 26, no. 12, pp. 11–24, 1993.

[3] B. A. Price, R. M. Baecker, and I. S. Small, "A principled taxonomy of software visualization," *Journal of Visual Languages and Computing*, vol. 4, no. 3, pp. 211–266, 1993.

[4] J. I. Maletic, A. Marcus, and M. Collard, "A task oriented view of software visualization," in *Proceedings of the 1st Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2002)*. IEEE, Jun. 2002, pp. 32–40.

[5] L. Merino, M. Ghafari, and O. Nierstrasz, "Towards actionable visualisation in software development," in *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*. IEEE, 2016. [Online]. Available: http://scg.unibe.ch/archive/papers/Meri16a.pdf

[6] B. Ens, D. Rea, R. Shpaner, H. Hemmati, J. E. Young, and P. Irani, "ChronoTwigger: A visual analytics tool for understanding source and test co-evolution," in *Working Conference on Software Visualization (VISSOFT)*. IEEE, 2014, pp. 117–126.

[7] F. Fittkau, E. Koppenhagen, and W. Hasselbring, "Research perspective on supporting software engineering via physical 3D models," in *Working Conference on Software Visualization (VISSOFT)*. IEEE, 2015, pp. 125–129.

[8] C. Anslow, S. Marshall, J. Noble, and R. Biddle, "Sourcevis: Collaborative software visualization for co-located environments," in *Software Visualization (VISSOFT), 2013 First IEEE Working Conference on*, Sep. 2013, pp. 1–10.

[9] C. Anslow, S. Marshall, J. Noble, E. Tempero, and R. Biddle, "User evaluation of polymetric views using a large visualization wall," in *Proceedings of the 5th International Symposium on Software Visualization*, ser. SOFTVIS '10. New York, NY, USA: ACM, 2010, pp. 25–34. [Online]. Available: http://doi.acm.org/10.1145/1879211.1879218

[10] G. Canfora, M. Di Penta, and L. Cerulo, "Achievements and challenges in software reverse engineering," *Commun. ACM*, vol. 54, no. 4, pp. 142–151, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1924421.1924451

[11] R. Wettel, M. Lanza, and R. Robbes, "Software systems as cities: a controlled experiment," in *Proceedings of the 33rd International Conference on Software Engineering*, ser. ICSE '11. New York, NY, USA: ACM, 2011, pp. 551–560. [Online]. Available: http://doi.acm.org/10.1145/1985793.1985868

[12] F. Fittkau, A. Krause, and W. Hasselbring, "Exploring software cities in virtual reality," in *Working Conference on Software Visualization (VISSOFT),*. IEEE, 2015, pp. 130–134.

[13] ——, "Hierarchical software landscape visualization for system comprehension: a controlled experiment," in *Working Conference on Software Visualization (VISSOFT)*. IEEE, 2015, pp. 36–45.

[14] A. R. Teyseyre and M. R. Campo, "An overview of 3D software visualization," *Transactions on visualization and computer graphics*, vol. 15, no. 1, pp. 87–105, 2009.

[15] R. Wettel and M. Lanza, "Visualizing software systems as cities," in *Proceedings of VISSOFT 2007 (4th IEEE International Workshop on Visualizing Software For Understanding and Analysis)*, 2007, pp. 92–99. [Online]. Available: http://dx.doi.org/10.1109/VISSOF.2007.4290706

[16] C. Knight and M. Munro, "Virtual but visible software," in *Information Visualization, 2000. Proceedings. IEEE International Conference on*. IEEE, 2000, pp. 198–205.

[17] T. Panas, T. Epperly, D. Quinlan, A. Saebjornsen, and R. Vuduc, "Communicating software architecture using a unified single-view visualization," in *Conference on Engineering Complex Computer Systems*. IEEE, 2007, pp. 217–228.

[18] J. I. Maletic, J. Leigh, A. Marcus, and G. Dunlap, "Visualizing object-oriented software in virtual reality," in *International Workshop on Program Comprehension (IWPC)*. IEEE, 2001, pp. 26–35.

[19] S. Huron, S. Carpendale, A. Thudt, A. Tang, and M. Mauerer, "Constructive visualization," in *Conference on Designing interactive systems*. ACM, 2014, pp. 433–442.

[20] J. I. Maletic and A. Marcus, "CFB: A call for benchmarks-for software visualization," in *Workshop on Visualizing Software for Understanding and Analysis (VISSOFT)*. Citeseer, 2003, pp. 113–116.

[21] T. Panas, R. Lincke, and W. Löwe, "Online-configuration of software visualization with Vizz3D," in *Proceedings of ACM Symposium on Software Visualization (SOFTVIS 2005)*, 2005, pp. 173–182.

[22] O. Greevy, S. Ducasse, and T. Gîrba, "Analyzing software evolution through feature views," *Journal of Software Maintenance and Evolution: Research and Practice (JSME)*, vol. 18, no. 6, pp. 425–456, 2006. [Online]. Available: http://scg.unibe.ch/archive/papers/Gree06bTraceScraperJSME-SCG.pdf

[23] M. Termeer, C. F. Lange, A. Telea, and M. R. Chaudron, "Visual exploration of combined architectural and metric information," *VISSOFT 2005. 3rd IEEE International Workshop on Volume*, vol. 0, p. 11, 2005.

[24] P. Irani, M. Tingley, and C. Ware, "Using perceptual syntax to enhance semantic content in diagrams," *Computer Graphics and Applications*, vol. 21, no. 5, pp. 76–84, 2001.

[25] R. Wettel, "Software systems as cities," Ph.D. dissertation, University of Lugano, Switzerland, Sep. 2010.

[26] S. Mittelstädt, D. Jäckle, F. Stoffel, and D. A. Keim, "ColorCAT: Guided design of colormaps for combined analysis tasks," in *Eurographics Conference on Visualization*, vol. 2, 2015.

[27] C. Anslow, J. Noble, S. Marshall, and E. Tempero, "Towards visual software analytics," *Australasian computing doctoral consortium (ACDC)*, 2009.

[28] E. Tempero, C. Anslow, J. Dietrich, T. Han, J. Li, M. Lumpe, H. Melton, and J. Noble, "The Qualitas Corpus: A curated collection of Java code for empirical studies," in *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*, Dec. 2010, pp. 336 –345.

[29] J. Nielsen, *Usability Engineering*, 1st ed. Morgan Kaufmann, Sep. 1993. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0125184069

[30] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 22, no. 140, pp. 1–55, 1932.

[31] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[32] H. Levene *et al.*, "Robust tests for equality of variances," *Contributions to probability and statistics*, vol. 1, pp. 278–292, 1960.

[33] S. Shaphiro and M. Wilk, "An analysis of variance test for normality," *Biometrika*, vol. 52, no. 3, pp. 591–611, 1965.

[34] E. R. Tufte, *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press, 2001.

[35] T. W. Malone, "What makes things fun to learn? heuristics for designing instructional computer games," in *Symposium on Small systems (SIGSMALL)*. ACM, 1980, pp. 162–169.