Overcoming Issues of 3D Software Visualization through Immersive Augmented Reality

Leonel Merino*, Alexandre Bergel†, Oscar Nierstrasz*
*Software Composition Group, University of Bern, Switzerland
†PLEIAD, University of Chile, Chile

***Abstract*—Several usability issues (*i.e.*, navigation, occlusion, selection, and text readability) affect the few 3D visualizations proposed to support developers on software engineering tasks. We observe that most 3D software visualizations are displayed on a standard computer screen, and hypothesize that displaying them in *immersive augmented reality* can help to *(i)* overcome usability issues of 3D visualizations, and *(ii)* increase their effectiveness to support software concerns. We investigate our hypothesis via a controlled experiment. In it, nine participants use 3D city visualizations displayed on a Microsoft HoloLens device to complete a set of software comprehension tasks. We further investigate our conjectures through an observational user study, in which the same participants of the experiment use a space-time cube visualization to analyze program executions. We collect data to *(1)* quantitatively analyze the effectiveness of visualizations in terms of user performance (*i.e.*, completion time, correctness, and recollection), and user experience (*i.e.*, difficulty, and emotions); and *(2)* qualitatively analyze how immersive augmented reality helps to overcome the limitations of 3D visualizations. We found that immersive augmented reality facilitates navigation and reduces occlusion, while performance is adequate, and developers obtain an outstanding experience. Selection and text readability still remain open issues.**

Companion video: https://youtu.be/1J4JQTYNgco

I. Introduction

Many software visualizations have been proposed to support developers in software engineering tasks [1]. However, only a few have included 3D visualization techniques [2], [3]; to name a few: CallGraphAnalyzer [4] supports feature location through the visualization of call graphs, Palantir [5] facilitates software project awareness and evolution, Evo-streets [6] supports the analysis of development history, Synchrovis [7] helps to monitor traces, ChronoTwigger [8] supports understanding source and test co-evolution, and ExplorViz [9] enables software comprehension through the visualizations of hierarchical software landscapes.

The main usability issues that have been raised against the use of 3D visualizations relate to: *(i) navigation*, *(ii) selection*, *(iii) occlusion*, and *(iv) text readability* [10]. As opposed to the simplicity of using 2D visualizations, navigating 3D software visualizations is perceived as an activity that is significantly more complex. Users struggle to navigate 3D visualizations using mice and keyboards when they have to select small elements. Often elements in 3D visualizations are affected by occlusion, and therefore, are hard to distinguish. Sometimes the lack of anchored points

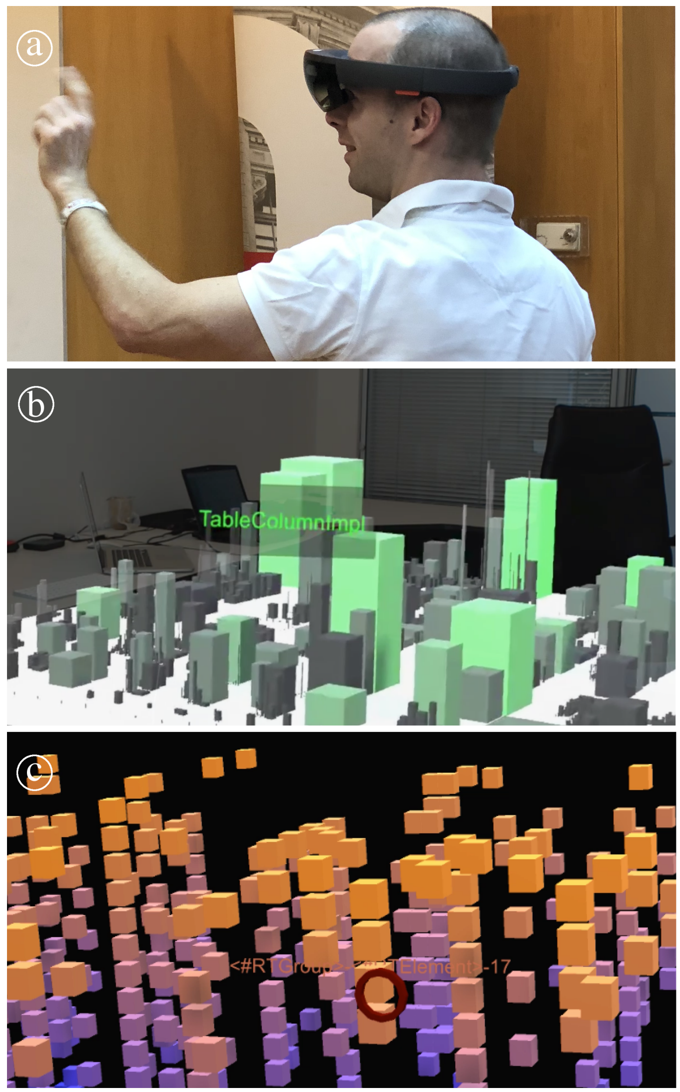


Figure 1: 3D software visualizations displayed in immersive augmented reality: (a) a developer wears an augmented reality device (*i.e.*, MS HoloLens) for visualization using, (b) the city metaphor, and (c) the space-time cube technique.

in 3D spaces promotes disorientation. Moreover, users usually complain when reading 3D text. [10], [11] These issues can impact the effectiveness of 3D software visualizations, that is, they can affect not only the performance of developers (*e.g.*, completion time, correctness, recollection), but also their experience (*e.g.*, difficulty, emotions).

We observe that most of the proposed 3D software visualizations are displayed using a standard computer screen as the medium [1]. This raises the question whether displaying 3D visualizations on a more appropriate medium can help to minimize the usability issues, and eventually boost their effectiveness to support software concerns. We hypothesize that immersive augmented reality (*e.g.*, MS HoloLens) can alleviate the main issues that reduce the effectiveness of 3D software visualizations. Accordingly, we focus on the following research questions:

RQ.1) How can immersive augmented reality help to minimize usability issues of 3D software visualizations?

- *RQ.1.1*) How well is *navigation* supported in various 3D software visualization techniques displayed in immersive augmented reality?
- *RQ.2.2*) How well is *selection* supported in various 3D software visualization techniques displayed in immersive augmented reality?
- *RQ.3.3*) How does *occlusion* affect developers who use various 3D software visualization techniques displayed in immersive augmented reality?
- *RQ.4.4*) How does *text readability* affect developers who use various 3D software visualization techniques displayed in immersive augmented reality?

RQ.2) How effective (*i.e.*, performance and experience) are 3D software visualizations displayed in immersive augmented reality to support software comprehension tasks?

We investigated these questions via a controlled experiment. In it, we selected 3D city visualizations (shown in Figure 1ⓑ), since they *(i)* have proved to be effective for software comprehension tasks [12], *(ii)* made their source code publicly available, and *(iii)* are easy to transfer to the immersive augmented reality device. We asked nine participants to complete a set of software comprehension tasks used in a previous evaluation [13]. We measured the effectiveness of visualizations in terms of user performance (*i.e.*, time, correctness, and recollection), and user experience (*i.e.*, difficulty, and emotions) for a quantitative analysis in which we compared our results to a baseline dataset of participants who used the same setup, but displayed on a standard computer screen.

To further investigate the potential of displaying software visualizations in immersive augmented reality, we conducted a user study. In it, we asked the same participants of the experiment to use the space-time cube visualization technique [14] displayed in immersive augmented reality (shown in Figure 1ⓒ) to analyze the execution of programs.

We found that an immersive augmented reality medium eases navigation and reduces occlusion, while excelling at providing developers a good experience in terms of engagement and emotions. However, we also found that the selection and text readability issues persist. The main contributions of the paper are:

(1) the design and execution of a controlled experiment, and a user study that aims at analyzing how immersive augmented reality can help to minimize the main reported issues with 3D visualization, and
(2) the discussion of the results of our evaluations, in which we elaborate on open questions that need to be addressed in the future.

We also contribute to the reproducibility of our research by making the source code of our implementations available[1].

II. Related Work

We now review software visualization approaches that have used multiple techniques and media. We elaborate on visualization approaches that have selected similar techniques, but displayed them on other media than those we selected. We describe approaches that have selected similar techniques and medium (immersive augmented reality), but that have not focused on the impact of the medium on the usability issues of 3D visualizations.

3D Software Visualizations. SeeIT3D [19] helps researchers to visualize eye gazes for understanding how developers read source code during comprehension tasks. ChronoTwigger [8] supports the analysis of source and test co-evolution. None of the described approaches focus on our selected visualization techniques: 3D space-time cube, and 3D city visualizations.

3D Space-Time Cube Visualizations. CuboidMatrix [14] proposes a matrix of cubes to explore dynamic structural relationships. Another study [22] employed a space-time cube visualization for detecting anomalies in the quality of software systems.

3D City Visualizations. CodeCity [20] provides developers an overview of software systems based on software metrics. Synchrovis [7] helps developers to monitor concurrent Java programs using the city metaphor. Even though these visualization approaches used our selected techniques, all of them are displayed on a standard computer screen.

Software Visualizations beyond the Standard Computer Screen. We found few software visualization approaches that have proposed to display visualization techniques in media different than standard computer displays. Three support the visualization of software systems displayed in virtual reality: ExplorViz [18] supports the visualization of hierarchical software landscapes, and CodePark [17] and CityVR [16] offer 3D code visualization using immersive virtual reality. Two other software visualization approaches that we consider closer to our work use augmented reality to display software visualizations. SkyscrapAR [23] supports the visualization of the city technique displayed in augmented reality for mobile devices to deal with software

[1] http://scg.unibe.ch/research/visar

Table I: Software visualizations reviewed in the related work.

Aspect [15]	Tool's Name	Software Concern	Environment	Technique	Medium
Structure	CityVR [16]	Architecture of Java systems	Unity	City metaphor	VR
Structure	CodePark [17]	Architecture of Java systems	Unity	City metaphor	VR
Structure	ExplorViz [9], [18]	Software metrics	Web	City metaphor	VR,SCS
Structure	SeeIT3D [19]	Architecture of Java systems	Eclipse	City metaphor	SCS
Structure	CodeCity [20]	Software metrics	Pharo	City metaphor	SCS
Behavior	NN [21]	Software metrics	Windows	Node-link diagrams	AR
Behavior	Synchrovis [7]	Execution traces of Java concurrent programs	Java	City metaphor	SCS
Evolution	CuboidMatrix [14]	Dynamic structural relationships	Pharo	Space-time cubes	SCS
Evolution	NN [22]	Anomalies in the quality of software systems	Pharo	Space-time cubes	SCS
Evolution	ChronoTwigger [8]	Source and test co-evolution	VR Juggler	Node-link diagrams	CAVE
Evolution	SkyscrapAR [23]	Architecture of Java systems	Processing	City metaphor	AR

evolution tasks. One study [21] proposed graph-based visualization of software metrics projected on a see-through display.

Table I presents some characteristics of the described visualization approaches in the related work. In contrast to our investigation, none of these efforts focus on evaluating how the medium selected to display visualization techniques can help to overcome usability issues of 3D visualization (*i.e.*, navigation, selection, occlusion, and text readability).

III. Controlled Experiment

We use the framework proposed by Wohlin *et al.* [24], and tailored by Merino *et al.* [25] to summarize the scope of our experiments in software visualization:

> "Analyze <*3D visualizations*> in the <*Pharo environment*> to support <*comprehension tasks*> using the <*city metaphor*> displayed on a <*standard computer screen*> for the purpose of <*comparison to visualizations displayed in immersive augmented reality*> with respect to the <*effectiveness*> and the <*prevalence of usability issues*> from the point of view of <*developers*> in the context of <*supporting comprehension tasks*>."

With the experiment, we aim to collect data to *(i)* qualitatively analyze whether the issues that arise in 3D visualizations displayed on a standard computer screen also are present when displayed in immersive augmented reality (*RQ.1*), and *(ii)* quantitatively compare the effectiveness to support software comprehension tasks of a 3D visualization technique displayed in immersive augmented reality and a standard computer screen (*RQ.2*).

A. Design

We built on the data collected in a previous experiment [13]. In particular, from that experiment we used the data collected from participants who visualized the systems using a standard computer display as a baseline, and compared those results to those that we gathered from participants who visualized the systems displayed in immersive augmented reality. We designed our experiment to use the same visualization technique (*i.e.*, city metaphor), to perform a subset of the same tasks, and to follow a similar procedure when conducting the experiment.

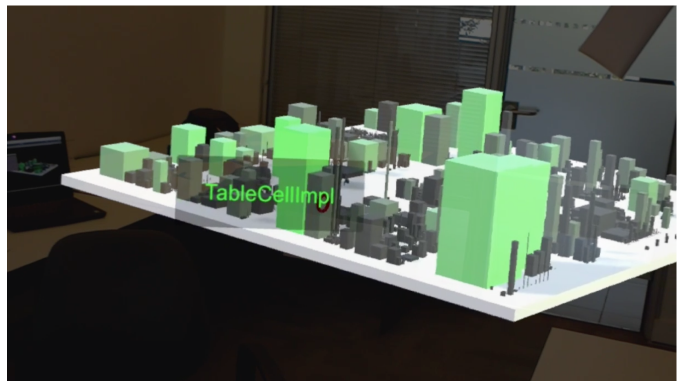


Figure 2: The city metaphor visualization technique displayed in immersive augmented reality.

Technique. *CityAR*. We selected the city metaphor visualization technique (shown in Figure 2). In it, the buildings in the city represent the classes in the system, and their size and color encode software metrics. We selected the city technique because it has proven to be effective to support software comprehension tasks [9], [12], [13].

Medium. We selected the media in the follow-up setups to conduct the experiment:

- i) *Standard Computer Screen (SCS)*. We used an Apple MacBook Pro with a resolution of 1440 x 900 pixels.
- ii) *Immersive Augmented Reality* (AR). We used a Microsoft HoloLens headset with a stereo 1268 x 720 resolution, see-through holographic lenses combined resolution, 60 Hz content refresh rate, and 30° H and 17.5° V field of view.

Interaction. To enable comparison between the two setups, we limited the interactions in the visualizations to navigation and selection. That is, participants using the computer screen setup navigate the system by rotating, panning, and zooming in the visualization, and they select classes by hovering over buildings with the mouse to read their names. Similarly, participants using the immersive augmented reality visualization navigate the system by walking, crouching, and approaching buildings in the city, and they select classes by pointing at buildings with their heads to read their names.

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

System	Version	# KLOC	# Classes	# Pkgs.	Size
Axion	1.0-M2	23	223	27	Small
Azureus	4.8.1.2	646	6619	560	Large

Dataset. We selected two open-source software systems (shown in Table II) that have been used extensively in other studies. The *Axion* database[2] that is a small software system, and the *Azureus* BitTorrent client[3], which is a large system.
Tasks. We excluded tasks that during the pilot experiment were considered ambiguous, or that required a long time to be completed. We included a set of tasks that participants can complete in approximately one hour to avoid fatigue (shown in Table III).
Implementation. The visualization displayed on the standard computer screen was provided by the CodeCity[4] implementation for Moose 5[5] on OSX. We transfered the visualization to the augmented reality device by exporting the models of the cities from CodeCity, and implementing a visualization using Unity 2017.3.0b8 (64-bit). To display the visualization in the HoloLens device and to enable interaction, we used the *MixedRealityToolkit-Unity*[6].

B. Hypotheses

Our main interest is to investigate how immersive augmented reality helps to alleviate usability issues of 3D software visualizations. However, to appreciate the potential benefits of such a medium, we also need to ask how effective are 3D software visualizations displayed in augmented reality for supporting software comprehension tasks. We consequently define the following null hypotheses:

- H_0^1 The *time* to complete comprehension tasks is not significantly different when visualizing software as cities with a standard computer screen or immersive augmented reality.
- H_0^2 The *correctness* of developer tasks is not significantly different when visualizing software as cities with a standard computer screen or immersive augmented reality.
- H_0^3 The *difficulty* perceived by developers is not significantly different when visualizing software as cities with a standard computer screen or immersive augmented reality.

C. Participants

The experiment followed a between-groups design, that is, each participant visualizes the systems using only one medium. We selected participants to have a fairly similar distribution of gender and educational level between the groups. The setup using a standard computer screen was used by group of 9 participants: 6 PhD, and 3 B/MSc students in Computer Science. Their average age was 27.67 ± 2.65 years (standard deviation), their average experience as a developer was 7.56 ± 3.36 years. The participants had little familiarity with 3D visualizations (*i.e.*, median of 2 in a 5-step Likert scale), and they were very familiar with the medium (*i.e.*, median of 5 in a 5-step Likert scale). Participants were invited and freely opted to participate in the study. They were not paid. Four participants were recruited from the University of Konstanz, and five from the University of Bern. The interviews were conducted from December 2016 to February 2017.

To evaluate the setup using an immersive augmented reality device, a group of nine participants was formed: three post-doc researchers, three PhD, and three MSc students in Computer Science. Their average age was 28.64 ± 5.26 years (standard deviation), their average experience as a developer was 8.95 ± 3.62 years. Participants reported little familiarity with 3D visualizations (*i.e.*, median of 2 in a 5-step Likert scale), and no previous experience using the medium (*i.e.*, median of 1 in a 5-step Likert scale). Participants were not paid. They were invited and freely opted to participate in the study. All of them were recruited from the University of Chile. The interviews were conducted from December 2017 to January 2018.

D. Procedure

The experiment was conducted in three locations: one at the University of Konstanz, another at the University of Bern, and one at the University of Chile. The rooms used in all locations have a similar size and lighting. The same experimenter conducted the experiments in all locations. A different setup was defined for each medium: for immersive augmented reality, participants wore a head-mounted display device (MS HoloLens). Participants interacted with the visualization by walking, and moving their heads. The tasks were read aloud by the experimenter. A legend with the encoding of the visualization was visible at all times. Participants who visualized the systems using a standard computer screen sat in a chair, and 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.

When the experiment started, we read an introduction to explain to the participants the scope of the experiment, and what they were expected to do. Then, participants had a training session where they viewed a visualization of the *Axion* system. They were asked to perform the tasks one-by-one. Once the participants felt confident with the encoding of the visualization, and with the tasks, we started the visualization of *Azureus*, and asked them to complete the tasks one at a time as they did during the training. This time, when they completed each task, we asked them to rate its perceived difficulty using a 5-step Likert scale [26].

[2] http://axion.tigris.org/
[3] https://sourceforge.net/projects/azureus/
[4] http://smalltalkhub.com/#!/~RichardWettel/CodeCity
[5] http://www.moosetechnology.org/
[6] https://github.com/Microsoft/MixedRealityToolkit-Unity

Table III: Comprehension tasks used in the experiment involving the analysis of three metrics: *Number of Methods* (NOM), *Number of Attributes* (NOA), and *Number of Lines of Code* (NLOC).

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)

To assess the emotions experienced by participants during the visualization, we used a set of emotion cards that we introduced in a previous experiment [13]. In consequence, once the participants finished all the tasks, we asked them to approach to the table where we previously placed 140 cards. Each card contained a word that describes an emotion. We placed positive emotions on the left side of the table and negative ones on the right. Cards were organized into eight groups of positive emotions and also eight of negative ones. Participants were asked to collect ten cards that describe the emotions experienced during the session, and to sort them according to their intensity. Since visualizations displayed in virtual reality promote recollection that helps developers to orientate during software comprehension tasks [27], we investigate how well is the recollection of developers who visualize systems in augmented reality. To evaluate near-time recollection, participants were asked to use an application[7] on a tablet[8] to draw their recollection of the visualization of *Azureus* (approximately twenty minutes after they finished with the visualization).

E. Data Collection

We collected several data points during the experiment. We recorded *(i)* videos of participants as they navigated visualizations, and the view they obtained of the visualization itself, *(ii)* digital drawings with the recollection of the visualization by participants, and *(iii)* the selected cards that described the experienced emotions by participants. We produced a single video recording for each participant. The video contains the complete intervention of the participant in our study. We used the video to accurately measure completion time and accuracy, as well as to identify emergent codes for qualitative analysis (*e.g.*, difficulties to read text on labels).

IV. User Study

To further investigate the potential of displaying software visualizations in immersive augmented reality, we conducted a user study. The goal of the study was to test a different visualization technique to stress immersion, and to collect the impressions of participants. We invited the same participants who visualized the systems in the controlled experiment in immersive augmented reality, and reused the same configuration. That is, the study was conducted in the same room, with the same experimenter, and they used the same device. The interview lasted around 20 minutes.

[7] Paper v.3.6.10

[8] iPad Pro 10.5"/Wifi/512GB

Figure 3: The space-time cube visualization technique displayed in immersive augmented reality.

Technique. In the interview, we introduced to the participants another visualization technique: *CuboidMatrix*. Space-time cube is a classical visualization technique in 3D where each data point is visualized as an element in a three-dimensional space. One of these dimensions is time. CuboidMatrix [14] maps color, size, and spatial location of each data point to metrics applied to the represented element. The original definition of CuboidMatrix offers a large range of interaction means, including projection, a way to select slices, and a fine camera control. In an augmented reality setting, most of the interactions are offered by movement of the visualization user. We configured the technique to use a 2 meter cubic physical 3D space. The upper cubes in the visualization are therefore close to the head of participants, bottom cubes are close to their feet, and the entire visualization can be rendered in the office space. Figure 3 shows the technique displayed in immersive augmented reality.

Tasks. Once the participants understood the encoding of the visualization technique, we asked them to complete

Table IV: Comprehension tasks used in the study.

Id	Task
T1	Which classes C1 interact with C2 only during the three first time periods? We are therefore looking for C1 and C2 for which we have a relation at (C1, C2, 1), (C1, C2, 2), and (C1, C2, 3)
T2	Which classes interact at the end of the execution, only during the last three time periods?
T3	Which are the two classes that interact during the overall execution?
T4	Which classes C1 interact with C2 only at time periods that are pair (2, 4, 6, ..., 18)?

a set of comprehension tasks. Similarly, we selected a subset of the tasks used on a previous study [14]. Table IV presents the tasks included in the study. The tasks aim to stress the abilities of the participants to identify visual patterns. Once the participants have completed the tasks, we asked them to speak freely about their experience with the visualization, and to compare that experience to the previous ones they had in the controlled experiment (where they used a different technique, and completed different tasks).

Dataset. We used the same dataset from a previous study [14]. The dataset, composed of 621 data points, shows the interaction of 101 different classes along a time frame of 19 periods. The space-time cube has a dimension of 56 × 45 × 19. We produced the dataset by monitoring the execution of a large application. A data point (c_1, c_2, t) indicates that during the time slice t, instances of the class c_1 sent messages to instances of the class c_2. The dataset is obtained by applying the Spy profiling framework [28] to the Roassal application [29].

V. Results

We quantitatively analyze the results of the controlled experiment (in Section III) based on the statistical analysis of the collected data. The Shapiro-Wilk's [30] test shows that not all measured variables follow a normal distribution, and therefore, a non-parametric test is required. We use the Kruskal-Wallis test [31]. We choose a 95% confidence interval (α = .05) to evaluate whether there are statistically significant differences in H_0^1 completion time, H_0^2 correctness, and H_0^3 difficulty (shown in Figure 4) between city visualizations used to support comprehension tasks displayed on a standard computer screen and immersive augmented reality (AR). Table V shows the results of the statistical tests that we carry out to analyze the measured variable of user performance (*i.e.*, completion time and correctness) and user experience (*i.e.*, difficulty).

A. User Performance

Completion Time. The variation of the time to *find outliers* (T1-T3) among the two media is much larger than the variation of the time within each medium. Thus, we reject H_0^1 for tasks T1-T3. However, the variation of the time to *identify patterns* (T4-T6) is not sufficiently significant, and thus, we cannot reject H_0^1 for tasks T4-T6.

Table V: Summary of the results of user performance in terms of completion time and correctness, and the results of user experience in terms of the perceived difficulty of tasks (significant values are highlighted with a gray background).

Task	Completion Time			Difficulty			Correctness		
	p-value	χ^2	m.	*p*-value	χ^2	m.	*p*-value	χ^2	m.
T1-T3	0.0043	8.1477	38.5	0.9812	6e-04	4	3.12e-06	21.7414	1
T4-T6	0.7323	0.117	27.0	0.0009723	10.8795	4	0.6416	0.2167	1

Table VI: The data-ink ratio of the recollection of participants who visualized the Azureus city in immersive augmented reality. The most detailed drawing *[P5]* is shown in Figure 8b.

	P1	P2	P3	P4	P5	P6	P7	P8	P9
%White	95.70	92.57	96.92	96.23	90.63	94.89	92.89	93.43	96.00
%Non-white	4.3	7.43	3.08	3.77	9.37	5.11	7.11	6.57	4
Ratio	0.04	0.08	0.03	0.04	0.10	0.05	0.08	0.07	0.04

> Developers who visualize software cities for comprehension displayed on a *standard computer screen* require the least *time* to *find outliers.*

Correctness. The variation of the correctness to *find outliers* (T1-T3) among the two media is much larger than the variation of the correctness within each medium. Thus, we reject H_0^1 for tasks T1-T3. However, the variation of the correctness to *identify patterns* (T4-T6) is not sufficiently significant, and thus, we cannot reject H_0^1 for tasks T4-T6.

> Developers who visualize software cities for comprehension displayed on a *standard computer screen* achieve the highest *correctness* to *find outliers.*

Recollection. We ask participants to draw their recollection of the visualization of the Azureus city using an application in a tablet. We analyze the drawings based on the data-ink ratio, which is a measure of the ink used in a visualization [32]. Participants freely draw their recollection without our intervention or any special requirement. We conjecture that the more ink they use, the more details of the visualization they recollect. To measure the ink, for each drawing we extract a summary of the statistics of the used colors.[9] In particular, we collect the percentage of non-white pixels. We compared the results to a baseline of the color statistics extracted from a picture of the city visualization of Azureus displayed on the computer screen. The results are presented in Table VI.

B. User Experience

Difficulty. The variation of the perceived difficulty to *find outliers* (T1-T3) among the two media is less than the variation of the perceived difficulty within each medium. Thus, we cannot reject H_0^3 for tasks T1-T3. However, the

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

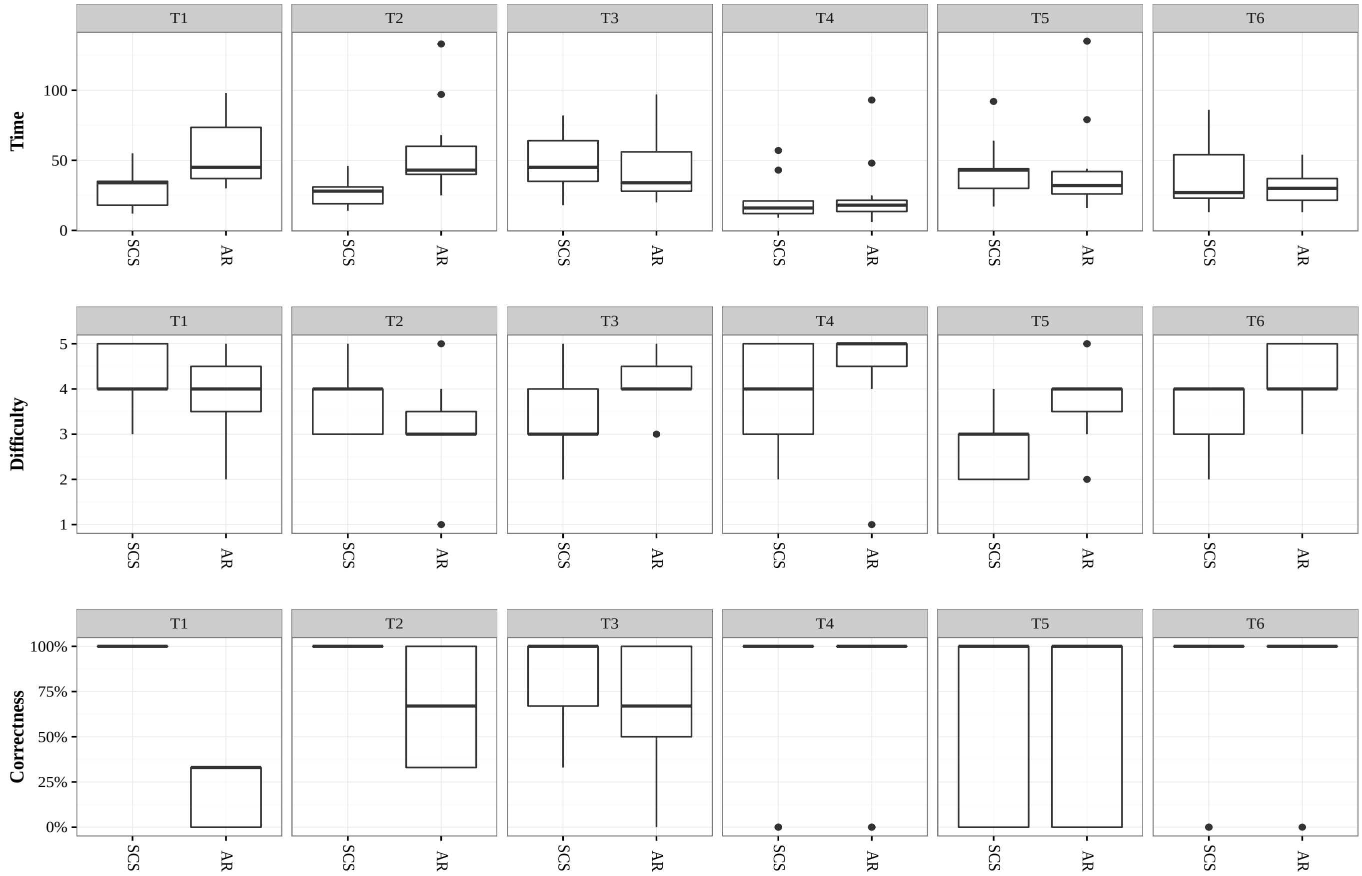


Figure 4: On the top, *time* (in seconds) that participants required to complete the tasks. In the middle, the *difficulty* perceived by participants when completing the tasks. At the bottom, the percentage of *correctness* achieved by participants.

variation of the perceived difficulty to *identify patterns* (T4-T6) is significant, and thus, we can reject H_0^3 for tasks T4-T6.

> Developers who visualize software cities for comprehension displayed in *immersive augmented reality* perceive the least *difficulty* to *find outliers.*

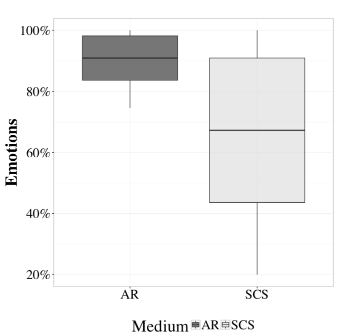


Figure 5: Emotion scores experienced by participants.

Emotions. We applied an emotion score metric introduced in a previous study [13]. The emotion score metric is the sum of the top ten emotions (ranked by intensity) weighted by their type (positive or negative). Figure 5 shows boxplots of the scores that obtained participants who used visualizations in each medium. We observe a trend, even though we did not find significant differences using the Kruskal-Wallis non-parametric test (χ^2 = 2.9896, *p*-value = 0.0838). Figure 6 shows the emotions experienced by the participants of the experiment. They are grouped by the medium used to display the visualizations. We excluded from the chart emotions that were reported only once by just one group. Positive emotions are on the left, and negative ones are on the right (a dashed vertical line separates them). Interestingly, several positive emotions (and a few negative) are common amongst participants. Participants positively felt *"curious"* (×8), *"interested"* (×5), *"playful"* (×5), *"inspired"* (×4), *"absorbed"* (×4), *"calm"* (×4), *"challenged"* (×3), *"comfortable"* (×3), *"content"* (×3), *"delighted"* (×3), *"dynamic"* (×3), *"excited"* (×3), *"fascinated"* (×3), *"pleased"* (×3), *"satisfied"* (×3), *"free"* (×2), and *"understanding"* (×2). Participants negatively felt *"uncertain"* (×4), *"hesitant"* (×2), and *"unsure"* (×2).

VI. Discussion

We revisit our research questions presented in Section I. Firstly, we discuss the effectiveness of the visualization techniques in terms of user performance, and user experience. Then, we discuss how displaying visualization techniques in immersive augmented reality helps to minimize issues

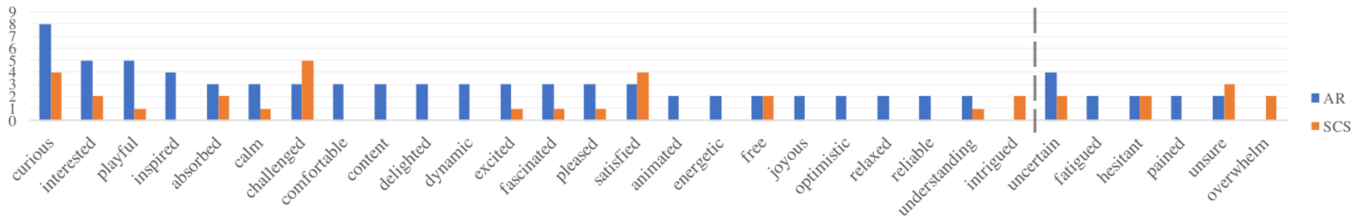


Figure 6: A histogram of frequent emotions experienced by participants who visualized 3D software cities displayed on a standard computer screen and in immersive augmented reality.

of 3D visualizations. Finally, we discuss the threats to the validity of our investigation.

RQ.1) *How can immersive augmented reality help to minimize usability issues of 3D software visualizations?*

The analysis of the data collected from the user study (Section IV) suggests that displaying the space-time cube visualization in immersive augmented reality facilitates navigation and alleviates occlusion, while selection and text readability remain open issues.

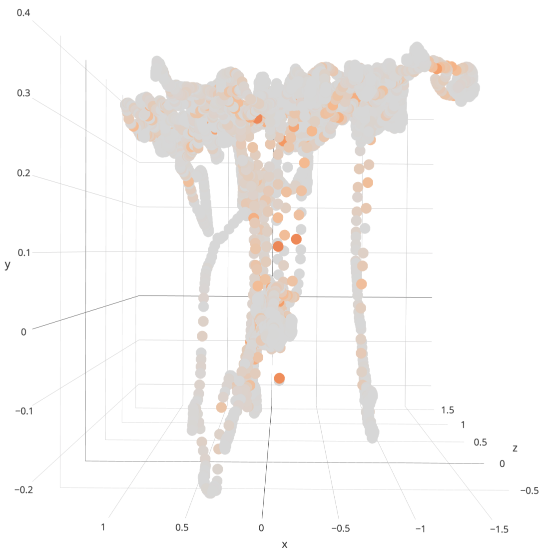
Navigation *(RQ.1.1)*. The navigation of 3D visualizations displayed in immersive augmented reality poses fewer difficulties to users than the navigation using a standard computer screen. Users can navigate immersive augmented reality as they do in real reality. They walked across the room to get closer to particular elements, and crouched to obtain a different view of elements that were closer to the ground. Figure 7a shows a 3D scatterplot of the position from the sensor in the headset used by a participant who visualized the Azureus city. Positions are colored to encode the speed of navigation. Possibly because in immersive augmented reality participants were able to see the surroundings while using the visualization (which provided anchored points that they could relate to the location of elements in the visualization) they did not reported feeling disoriented. Participants were especially creative to find solutions beyond the features included in the visualization to complete the tasks. For instance, since the 3D space-time cube visualization uses a large space, participants had to move across the room frequently. While most participants walked across the room, some opted for navigating the visualization using a wheeled chair (shown in Figure 7b). In our experience, it is critical to scale 3D visualizations in immersive augmented reality to human size to ease navigation. Figure 7c shows the navigation of a participant who used the space-time cube visualization in immersive augmented reality. In general, the space-time cube visualization involves more intensive navigation than the city technique.

Selection *(RQ.1.2)*. Participants selected buildings to read the names of the classes using a pointer centered in their field of view. Once selected, a label on top of the associated building was displayed. Participants often had trouble to stabilize the pointer on small buildings for enough time to read the labels. Participants mentioned that a feature such as a ruler could be included in the visualization to compare the sizes of buildings. Some resourceful participants used their hands and pieces of paper to measure the sides of buildings. The difficulties that users experience to select elements in 3D visualizations impact their effectiveness, and better selection mechanism can improve user performance and experience. We believe that the main asset of the immersive augmented reality medium is that it enables users to select elements in the visualization through a natural approach (*e.g.*, head movement, hand gestures, voice recognition), lowering the barrier to learn the selection mechanism, and consequently, facilitating the adoption of 3D visualizations.

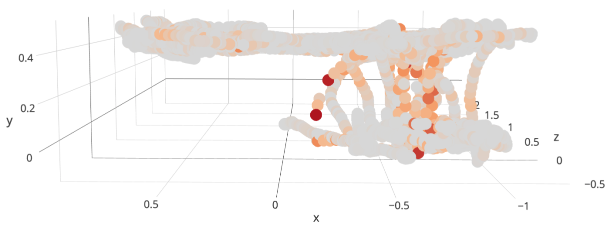
Occlusion *(RQ.1.3)*. 3D visualizations displayed on a standard computer screen suffer from occlusion. In space-cube visualizations, cubes in the same row or column tend to overlap. In city visualizations, short buildings are often hidden behind tall skyscrapers. 3D visualizations usually deal with occlusion by including additional cues. For instance, CuboidMatrix [22] uses transparency and orthogonal axes to enable the detection of walls, pillars and beams of aligned cubes. CodeCity [20] includes features to highlight, rotate, pan, and zoom. We observe that occlusion is closely related to navigation since occluded elements can be revealed by changing the position of the viewer. However, navigation of 3D visualizations displayed on the standard computer screen is difficult, and thus, dealing with occlusion is hard. In contrast, since navigation of 3D visualizations in immersive augmented reality poses fewer difficulties, occlusion is not as relevant. To minimize the occlusion in 3D city visualizations, we scaled the cities to fit the size of an office desk. Thus, short buildings that are usually hidden are revealed when the user moves. Since the the HoloLens device includes a small viewport[10], participants stood from a distance of the visualization to obtain an overview of the whole city. We also found that participants who failed to complete some tasks were misled by large and bright buildings that prevented them from seeing thin and black facade buildings, in particular when they had to find candidates for long facade classes.

Readability *(RQ.1.4)*. Text reading in 3D visualizations is still an issue. Almost all participants commented that the labels

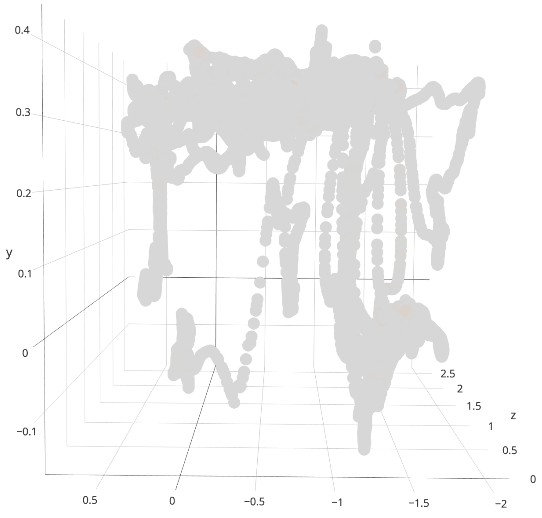
[10]A framed area on a display screen for viewing information



(a) Navigation of 3D city visualizations: Standing.



(b) Navigation of 3D city visualizations: Sitting.



(c) Navigation of 3D space-time cube visualizations.

Figure 7: Navigation of 3D visualizations displayed in immersive augmented reality. Nodes encode the location of participants, while the color encodes the speed of their movement.

with the name of the classes were hard to read. However, we noticed that various aspects affect text readability, not only the fact of displaying text in 3D. For instance, a high contrast between the color of the text and its canvas can ease reading. Also, we noticed that text that is not fixed in the visualization (*e.g.*, anchored instead to the body of users) can be the source of difficulties. Frequently, participants found it difficult to stabilize the labels to read the names of classes, since labels are visible only when the pointer is on the elements.

RQ.2) *How effective (*i.e.*, performance and experience) are 3D software visualizations displayed in immersive augmented reality to support software comprehension tasks?*

The results of the experiment (Section III) show significant differences in user performance and user experience when using the 3D software city visualization to deal with various software comprehension tasks. We observe that 3D visualizations are effective to support developers in software comprehension tasks. Moreover, the effectiveness of 3D visualizations can be increased if they are displayed in suitable media, while minimizing usability issues.

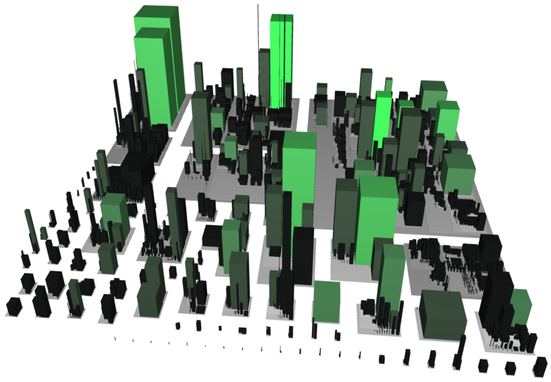
We found that 3D city visualizations displayed in immersive augmented reality promote the detection of visual patterns that developers can relate to the characteristics of software systems. For instance, when we asked participants to infer policies to name the classes in package, one participant noticed that the greenest buildings in the city are named with the *"impl"* suffix, and that the interfaces (that define the signature of the methods) of those *impl*ementation classes were usually dark and thin. Hence, to discover interfaces and their implementation, participants had to look for buildings of the same height (*i.e.*, number of methods) in which one was bright green and the other one was thin and black.

A. User Performance

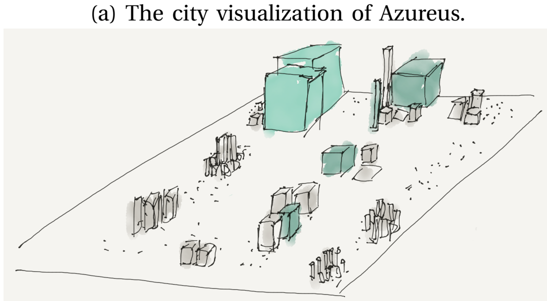
We now discuss the results of the performance of participants shown in Table V and Figure 4.

Completion Time and Correctness. Curiously, even though participants required the least time and obtained the highest correctness to find outliers when using 3D city visualizations displayed on the standard computer screen, participants perceived tasks to be the most difficult. Surely, fine-tuning the design of the immersive augmented reality deployment can improve completion time and correctness of tasks. For instance, the choice of the size of the city can impact the amount of navigation required to find elements.

Recollection. We observed that participants who used immersive augmented reality were able to recollect many details of the system. In fact, strong candidates for design problems (*e.g.*, god classes, data classes, long facades) appear frequently in the drawings. For instance, the participant who made the drawing shown in Figure 8b, remembered several distinctive patterns and outliers. Two massive green buildings in the left corner, and on the top center a candidate for god class that is surrounded by a few thin and long facade classes (Figure 8a shows the original 3D city visualization of Azureus).



(a) The city visualization of Azureus.



(b) A drawing with a participant's recollection of Azureus.

Figure 8: The recollection of a participant who visualized Azureus in immersive augmented reality.

B. User Experience

Difficulty. The differences in the perceived difficulty by participants who completed comprehension tasks that required them to identify patterns are statistically significant. Participants perceived those tasks to be the least difficult when they used the visualization displayed on the computer screen.

Emotions. We found that participants felt many positive emotions, in particular, "curious" and "challenged" can suggest that the visualization offers a good level of engagement [13]. Much less frequently, participants experienced some negative emotions. Although negative emotions were uncommon, and when felt, the emotions were reported to be fairly weak, the fact that participants felt "fatigued" and "pained", led us to think that the sessions demanded too much effort from them. We believe that this effect could be mitigated by designing shorter sessions, or by introducing breaks.

Threats to Validity

Construct validity. To mitigate the threat of formulating research questions that may not provide complete coverage of the usability issues in 3D software visualizations, we included questions that focus on the main issues reported in related work. We ensure an appropriate evaluation by expanding the design of an existing evaluation [13]. The quality of the visualizations in the two media may have effected the performance of participants, which we mitigated by transferring visualizations from the implementation displayed on the standard computer screen to immersive augmented reality by an automatic procedure. Consequently, positions, sizes, and colors of buildings were the same.

Internal validity. To mitigate bias in the selection of groups, we formed similar groups in terms of education level, gender, age and experience in software development. We also allowed participants to choose the perspective in their drawings, and add labels for clarifications, to avoid bias in the drawing skills of participants. Moreover, environmental aspects such as the room, light, and experiment length might be different. We interviewed participants in three different locations. We chose rooms with similar characteristics, and followed an identical procedure carried out by the same experimenter.

External validity. The novelty of immersive augmented reality might have affected the perception of participants. We mitigated this effect by allowing participants to familiarize themselves with the device during the training phase. To avoid a *learning effect* we opted for a between-groups design, and so each participant visualized the systems using only one of the media.

VII. Conclusion

The usability of 3D software visualizations is affected by navigation, selection, occlusion, and text readability issues. We investigated whether these issues can be minimized by displaying 3D visualizations in immersive augmented reality via a controlled experiment. In it, we compared the results of participants who visualized software systems displayed in immersive augmented reality to a baseline dataset of participants who used the configuration but displayed on the standard computer screen. We also conducted a user study, in which the same participants of the experiment used the space-time cube visualization technique displayed in immersive augmented reality to analyze the execution of programs. We found that immersive augmented reality eases navigation and reduces occlusion. Nevertheless, selection and text readability remain issues. We also found that 3D city visualizations displayed in immersive augmented reality are effective to support software comprehension tasks. Using this medium developers obtained the highest performance to find outliers, and perceived the least difficulty to identify patterns.

In the future, we plan to investigate tasks in other software concerns that could benefit from the characteristics of visualizations displayed in immersive augmented reality.

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).

References

[1] L. Merino, M. Ghafari, and O. Nierstrasz, "Towards actionable visualization for software developers," *Journal of Software: Evolution and Process*, vol. 30, no. 2, pp. e1923–n/a, 2017. [Online]. Available: http://scg.unibe.ch/archive/papers/Meri17a.pdf

[2] 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.

[3] L. Merino, "The medium of visualization for software comprehension," PhD thesis, University of Bern, Jun. 2018. [Online]. Available: http://scg.unibe.ch/archive/phd/merino-phd.pdf

[4] J. Bohnet and J. Döllner, "Visual exploration of function call graphs for feature location in complex software systems," in *Proc. of SOFTVIS*. ACM, 2006, pp. 95–104.

[5] R. M. Ripley, A. Sarma, and A. Van der Hoek, "A visualization for software project awareness and evolution," in *Proc. of VISSOFT*. IEEE, 2007, pp. 137–144.

[6] F. Steimann, T. Pawlitzki, S. Apel, and C. Kästner, "Types and modularity for implicit invocation with implicit announcement," *Transactions on Software Engineering Methodology*, vol. 20, no. 1, pp. 1:1–1:43, Jul. 2010. [Online]. Available: http://doi.acm.org/10.1145/1767751.1767752

[7] J. Waller, C. Wulf, F. Fittkau, P. Dohring, and W. Hasselbring, "Synchrovis: 3D visualization of monitoring traces in the city metaphor for analyzing concurrency," in *Proc. of VISSOFT*. IEEE, 2013, pp. 1–4.

[8] 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 *Proc. of VISSOFT*, Sep. 2014, pp. 117–126.

[9] F. Fittkau, A. Krause, and W. Hasselbring, "Hierarchical software landscape visualization for system comprehension: a controlled experiment," in *Proc. of VISSOFT*. IEEE, 2015, pp. 36–45.

[10] R. Brath, "3D InfoVis is here to stay: Deal with it," in *Proc. of VIS*. IEEE, 2014, pp. 25–31.

[11] J. Wood, S. Kirschenbauer, J. Döllner, A. Lopes, and L. Bodum, "Using 3D in visualization," in *Exploring Geovisualization*. Elsevier, 2005, pp. 293–312.

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

[13] L. Merino, J. Fuchs, M. Blumenschein, C. Anslow, M. Ghafari, O. Nierstrasz, M. Behrisch, and D. Keim, "On the impact of the medium in the effectiveness of 3D software visualization," in *Proc. of VISSOFT*. IEEE, 2017, pp. 11–21. [Online]. Available: http://scg.unibe.ch/archive/papers/Meri17b.pdf

[14] T. Schneider, Y. Tymchuk, R. Salgado, and A. Bergel, "CuboidMatrix: Exploring dynamic structural connections in software components using space-time cube," in *VISSOFT'16: Proceedings of the 4th IEEE Working Conference on Software Visualization*. IEEE, 2016, pp. 116–125. [Online]. Available: http://scg.unibe.ch/archive/papers/Schn16a.pdf

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

[16] L. Merino, M. Ghafari, C. Anslow, and O. Nierstrasz, "CityVR: Gameful software visualization," in *Proc. of ICSME*. IEEE, 2017, pp. 633–637. [Online]. Available: http://scg.unibe.ch/archive/papers/Meri17c.pdf

[17] P. Khaloo, M. Maghoumi, E. Taranta, D. Bettner, and J. Laviola, "Code Park: A new 3D code visualization tool," in *Proc. of VISSOFT*. IEEE, 2017, pp. 43–53.

[18] F. Fittkau, A. Krause, and W. Hasselbring, "Exploring software cities in virtual reality," in *Proc. of VISSOFT*. IEEE, 2015, pp. 130–134.

[19] B. Sharif, G. Jetty, J. Aponte, and E. Parra, "An empirical study assessing the effect of SeeIT 3D on comprehension," in *Proc. of VISSOFT*. IEEE, 2013, pp. 1–10.

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

[21] P. Kapec, G. Brndiarová, M. Gloger, and J. Marák, "Visual analysis of software systems in virtual and augmented reality," in *Proc. of INES*. IEEE, 2015, pp. 307–312.

[22] Y. Tymchuk, L. Merino, M. Ghafari, and O. Nierstrasz, "Walls, pillars and beams: A 3d decomposition of quality anomalies," in *Proc. of VISSOFT*. IEEE, 2016, pp. 126–135. [Online]. Available: http://scg.unibe.ch/archive/papers/Tymc16a.pdf

[23] R. Souza, B. Silva, T. Mendes, and M. Mendonça, "SkyscrapAR: An augmented reality visualization for software evolution," in *Proc. of WBVS*, 2012.

[24] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[25] L. Merino, M. Ghafari, C. Anslow, and O. Nierstrasz, "A systematic literature review of software visualization evaluation," *Journal of Systems and Software*, vol. 144, pp. 165–180, Oct. 2018. [Online]. Available: http://scg.unibe.ch/archive/papers/Meri18a.pdf

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

[27] A. Elliott, B. Peiris, and C. Parnin, "Virtual reality in software engineering: Affordances, applications, and challenges," in *Proc. of ICSE*. IEEE Press, 2015, pp. 547–550.

[28] A. Bergel, F. B. nados, R. Robbes, and D. Röthlisberger, "Spy: A flexible code profiling framework," *Journal of Computer Languages, Systems and Structures*, vol. 38, no. 1, Dec. 2011. [Online]. Available: http://bergel.eu/download/papers/Berg10f-Spy.pdf

[29] A. Bergel, *Agile Visualization*. LULU Press, 2016. [Online]. Available: http://AgileVisualization.com

[30] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, p. 591, 1965. [Online]. Available: +http://dx.doi.org/10.1093/biomet/52.3-4.591

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