How EvoStreets Are Observed in Three-Dimensional and Virtual Reality Environments

Marcel Steinbeck
University of Bremen, Germany
marcel@informatik.uni-bremen.de

Rainer Koschke
University of Bremen, Germany
koschke@uni-bremen.de

Marc O. Rüdel
University of Bremen, Germany
mor@uni-bremen.de

***Abstract*—When analyzing software systems, a large amount of data accumulates. In order to assist developers in the preparation, evaluation, and understanding of findings, different visualization techniques have been developed. Due to recent progress in immersive virtual reality, existing visualization tools were ported to this environment. However, three-dimensional and virtual reality environments have different advantages and disadvantages, and by transferring concepts, such as layout algorithms and user interaction mechanisms, more or less one-to-one, the characteristics of these environments are neglected. In order to develop techniques adapting to the circumstance of a particular environment, more research in this field is necessary.**

In previously conducted case studies, we compared EvoStreets deployed in three different environments: 2D, 2.5D, and virtual reality. We found evidence that movement patterns—path length, average speed, and occupied volume—differ significantly between the 2.5D and virtual reality environments for some of the tasks that had to be solved by 34 participants in a controlled experiment. In this paper, we analyze the results of this experiment in more details, to study if not only movement is affected by these environments, but also the way how EvoStreets are observed. Although we could not find enough evidence that the number of viewpoints and their duration differ significantly, we found indications that in virtual reality viewpoints are located closer to the EvoStreets and that the distance between viewpoints is shorter. Based on our previous results and the findings of this paper, we present visualization and user interaction concepts specific to the kind of environment.

I. Introduction

Maintaining software systems is a challenging tasks due to their complexity and versatility. In order to assist developers in understanding large scale systems, tools for (semi-)automatic software analysis have been developed in the last decades. However, processing analysis results, in turn, became difficult because a large amount of data accumulates. Software visualization tries to close the gap between data collection and data evaluation by mapping software-related attributes to visual components—utilizing the human ability to recognize patterns in data. Based on the software-as-a-city concept, originally proposed by Wettel and Lanza [1], [2], [3], *EvoStreets* [4], [5] are well suited to visualize a software's hierarchy and evolution. Elements, such as source code files, classes, or methods, are depicted as three-dimensional blocks, composed into segments according to the hierarchy they belong to and segments are depicted as nested streets, branching at each level change in the hierarchy. Exploiting 3D space, the width, height, and length of blocks are used to express certain metrics, for instance, lines of code, change frequency, or clone rate. In addition, a color gradient applied onto the blocks may be used to depict a further metric, allowing to locate local and global hotspots. Unlike CodeCities, the layout generated by EvoStreets is more resilient to changes, which allows to add, remove, and resize elements within certain limits. Thus, modifications can be tracked more easily by the human beholder.

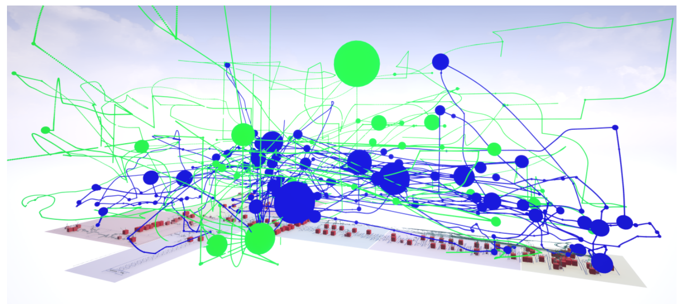


Fig. 1: Movement paths and viewpoints (depicted as spheres) of participants analyzing cloning in EvoStreets in virtual reality (blue) and 2.5D (green). A sphere's diameter shows the residence time of the corresponding viewpoint.

Several implementations of the CodeCities and EvoStreets visualization exist for two- and three-dimensional (also referred to as 2D and 2.5D) rendering on regular two-dimensional displays. Recently, in the attempt to continuously improve these techniques and in the hope that advantages observed outside of software engineering [6] also apply to software visualization, researchers begun to develop systems that make use of immersive three-dimensional virtual realty (VR). In a previously controlled experiment, we studied the performance (time required for task completion and correctness of answers) of 34 participants who were examining cloning in software using an EvoStreets visualization deployed in different environments (2D, 2.5D, and VR) [7]. We could not find enough evidence that performance is affected by any of the environments, Yet, we found in a follow up study [8] that the path length (the distance one moves within an EvoStreets), average speed (the length of a path put in relation to the time that was required to move along this path), and occupied volume (the convex hull of the movement trajectory) differ significantly between the 2.5D and VR environments for some of the tasks, indicating that movement is less extensive in

VR. That being said, more research on how human beholders interact with EvoStreets in different environments is necessary, to identify visualization and user interaction concepts that adapt to the characteristics of a particular environment. Due to the design of our experiment, that is, all environments were using the same visualization engine (*SCOOP* [9]) and applied the same visual mappings to the components of the EvoStreets, the threat that differences observed between the 2.5D and VR environments result from different EvoStreets implementations is minimized. Thus, we suspect that insights into differences specific to these environments, if there are any, can be gathered by analyzing the available data in more details.

Contributions. Based on the data recorded in our previous experiment, we extracted the viewpoints that were taken by the participants while exploring the EvoStreets in the 2.5D and VR environments. We compare these viewpoints to study if not only movement is affected by these environments, but also the way how EvoStreets are observed. Taking into account our previous results and the findings of this paper, we propose different visualization and user interaction concepts that adapt to the characteristics of 2.5D and VR environments.

Humans can move freely in EvoStreets. Their path is typically a sequence of movements from one location to another one followed by outlooks, where the users linger at the same location and turn only their field of sight (e.g., by turning their head and, thus, head-mounted display in VR). We call such locations, where users take only looks, *viewpoints*. A viewpoint can be characterized by two metrics: i) a position in 3D space and ii) a residence time, that is, the duration spent at the corresponding position. According to our previous results, movement patterns in EvoStreets differ between the 2.5D and VR environments. This observation raises the question whether viewpoints differ in these environments, too. Hence, our first research question is:

RQ1 Do the two different environments, 2.5D and VR, effect viewpoints?

In order to develop concepts that are adapting to the characteristics of the 2.5D and VR environments, comparing the metrics of viewpoints (position in 3D space and residence time) alone might not be enough. Thus, we propose to also examine how viewpoints change over time and to analyze whether there are environment-specific patterns. That being said, our second research question is:

RQ2 Are there patterns regarding the changes of viewpoints that are specific to the 2.5D and VR environments?

The remainder of this paper is organized as follows. Section II presents related research. The design of our previously conducted controlled experiment as well as our operational hypotheses are described in Section III. Results are presented and discussed in Section IV. Section V concludes.

II. Related Works

Our research presented in this paper addresses the question whether different kinds of visualization environments, namely, 2.5D desktop computers (with monitor, keyboard, and mouse) and immersive virtual reality systems (with head-mounted displays and hand-held controllers) have an effect on how humans observe and interact with EvoStreets. We believe that transferring visualization and user-interaction concepts developed for one environment one-to-one to another is suboptimal as they have been developed with different focus. Thus, we examine potential differences between these environments to find i) whether specific concepts are necessary and ii) how they should be designed to adapt to the characteristics of an environment. In this section, we will present related research of software visualization using the software-as-a-city metaphor and findings regarding differences between 2.5D and VR.

A. Visualizing Software as a City

In the attempt to assist developers in understanding complex software systems, several analysis tools have been developed. Results gathered by such tools can be modeled by different kinds of graphs, allowing to express, for instance, the hierarchical structure of a software system and dependencies between elements. These graphs are the basis for many visualizations [10]. Due to the variety of the usage scenarios and requirements, many kinds of visualization techniques exist. For a broader overview of software visualization, we refer the reader to more comprehensive surveys [11], [12], [13], [14], [10], [15], [16], [17], [18]. In the following, we will summarize techniques that come closest to EvoStreets.

The basis of EvoStreets are CodeCities, which in turn are an extension of Treemaps. The idea of treemapping is to recursively subdivide a rectangular shape into subareas according to the hierarchy of the data that is to be visualized [19], for example, source code files in directories. The size of the resultant leaf areas visually encodes a given property, for instance, lines of code, complexity, and change frequency. This allows to compare elements and identify peculiarities. By applying a color or texture, an additional property can be encoded, so that hotspots regarding this property can be determined easily. Utilizing 3D space, yet another property can be depicted by mapping its value onto the height of the leaves, yielding to three-dimensional blocks. Due to perceiving these blocks as a city, this technique is known as *CodeCity* [1], [2], [3]. Treemaps and CodeCities are designed to make optimal use of the available space. However, the generated layout is not flexible enough to visualize the evolution of a software system (elements need to be added, removed, and relocated), leading to the issue that a city's structure can change drastically from one version to another—although it must be mentioned that researchers have started to work on that particular problem recently [20]. Furthermore, the compact layout offers very little distinct patterns and, thus, hinders humans in recollecting visited places.

To overcome these shortcomings, Steinbrückner and Lewerentz [4], [5] proposed *EvoStreets*, a visualization technique in which a software's hierarchy is depicted using road junctions rather than subdivided areas. Each level of the hierarchy is mapped to a street whose width visually encodes the nesting level—the lower the level of the hierarchy is, the thinner the corresponding street gets. Similar to CodeCities, leaves are represented as three-dimensional blocks. EvoStreets, on the one hand, require more space but, one the other hand, allow parts of the city to grow and shrink without effecting the entire layout, which helps in maintaining a beholder's mental map.

Along with the internal structure of a software, relations between elements may be of interest. A common technique to visualize relations in Treemaps, CodeCities, and EvoStreets is to connect the corresponding areas (Treemaps) or blocks (CodeCity and EvoStreets) with edges. Yet, if drawn as straight lines, edges easily create visual clutter due to crossing each other. Holten proposed hierarchical edge bundling, an approach which reduces some of the clutter by drawing edges as B-Splines [21], [22]. The location of the control points of the B-Splines is based on the hierarchical structure of the visualized elements. By sharing control points among elements with similar nesting, edges with similar from and to location are bundled analogously to a cable tie.

Recently, researcher have started to visualize EvoStreets in VR [23], [9], [7] to study if this environment helps in solving developer tasks. The details of our previously conducted controlled experiment [7] (which forms the starting point of our research and from which we extracted the viewpoints for this paper) is presented in Section II-C and III.

B. Virtual Reality Versus Desktop

Visualization techniques using 2.5D and VR have been explored for quite some time. As early as 2000, Knight and Munro gave an overview of software visualization in VR [24]. Since then, 2.5D and VR environments have been used to visualize static [25], [26], [27], [28], [29], [30], [31], [32] as well as dynamic information [33], [34], [35]. There is already a great body of knowledge on 2.5D and VR visualization outside of computer science [36], [37], [38]. Studies have shown that head-mounted displays (HMDs) may have a positive effect on the orientation of human beholders in three-dimensional environments [6]. Sousa Santos et al., on the other hand, found in an experiment on navigation that, although being generally satisfied with VR, participants performed actually better in the desktop environment. Ruddle et al. studied the navigation in computer-simulated worlds using HMDs and regular desktop environments [39], [40]. In their first experiment, where participants had to navigate through several rooms within virtual buildings [39], they found that the HMD had an advantage on the speed of the participants. In their second experiment [40] Ruddle et al. looked into proprioceptive feedback and its influence on navigation within a virtual maze. According to their findings, viewing the mazes using a HMD had little effect. The conflicting results of Sousa Santos et al.'s and Ruddle et al.'s experiment is a subject of further research.

C. Comparing EvoStreets in Different Environments

Running a controlled experiment with nine participants (and a subsequent observational user study), Merino et al. [41] compared CodeCities in 2.5D and augmented reality (VR), studying the effect of these environments on navigation, selection, occlusion, and text readability. They found that AR assists in navigation and reduces occlusion, while performance in program-comprehension tasks was adequate. Yet, they identified text readability as an open issue.

This paper is based on the results of previously conducted case studies [7], [8], in which we investigated whether performance (time required for task completion and correctness of answers) and movement trajectories (path length, average speed, and occupied volume) in EvoStreets are affected by different environments—orthographic projection with keyboard and mouse (2D), 2.5D projection with keyboard and mouse (2.5D), and virtual reality with head-mounted display and hand-held controllers (VR). Analyzing clones in existing software systems, 34 participants had to solve three different tasks (derived from a typical visual-analytics context), each of them deployed in all these environments. The subject systems are listed in Table I. Figure 2 shows the corresponding EvoStreets.

TABLE I: Subject systems of our previously conduced controlled experiment. Guava was used for training.

Name	Lines of Code	Files	Clones
(Guava)	75,042	516	53
Jillion	75,520	929	66
JRuby	227,145	1360	110
Spring Boot	181,795	3042	228

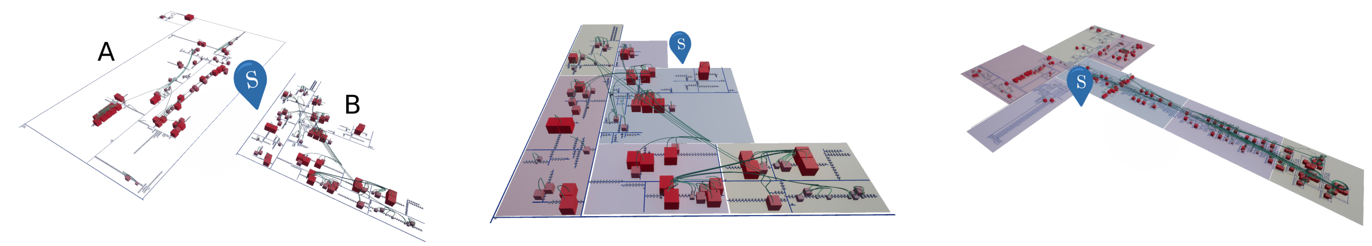
The participants were recruited through convenience sampling. About half the participants (17) were undergraduate students attending a VR project on EvoStreets. The others were (graduate and advanced undergraduate) students of a software engineering course taught by our research group (5), regular students (3), researchers (7), and professional developers (2). In order to counter learning effects, the participants were divided into six groups of equal size where each group was using a different combination of task and environment. We could not find enough evidence that any of the environments effects performance significantly. Yet, we found indications that movement in virtual reality, compared to the 2.5D environment, is less extensive. The setup of the experiment is described in more detail in the next section.

III. Experimental Setup and Hypotheses

The data we gathered to answer our research questions stated in Section I are taken from previously conducted case studies [7], [8]. The following section first describes the setup of the underlying experiment and then presents our operational hypotheses.

A. Environments

The experiment presented in [7] compared the effects of the three different environments 2D, 2.5D, and VR on visual clone analysis in EvoStreets. For this purpose, three different tasks had to be solved by the participants, each of them taking place in one of these environments. In a follow-up study [8] we examined the recorded movement trajectories of the participants regarding path length, average speed, and occupied volume in the 2.5D and VR environments. For this paper, we refined the movement data and extracted the viewpoints taken



(a) Task 1 (count connections in different systems): Decide for the two systems A and B which one contains more fragments cloned in other source files.

(b) Task 2 (count connection between subsystems): Decide for the six subsystems which pair of subsystems shares the most connections.

(c) Task 3 (count blocks in subsystems): Decide for the five subsystems which contains the most source files with clones (red colored blocks).

Fig. 2: The EvoStreets that had to be analyzed by the participants. S marks the starting points.

by the participants while exploring the EvoStreets. Following the same arguments as in [8], that is, comparing the 2D environment with the 2.5D and VR environment is difficult because movement in 2D contains only two axes, we focus on the 2.5D and VR environments only. The setup of the 2.5D and VR environments was as follows:

2.5D environment (2.5D) The three-dimensional EvoStreets model of the corresponding subject system is rendered on a regular two-dimensional display. The participants were able to move within the EvoStreets, that is, change their position in 3D space and rotate the view, by using a keyboard and mouse.

VR environment (VR) Using the same visual mappings as for the 2.5D environment, the EvoStreets are rendered on a head-mounted display (HTC Vive), presenting the scene in stereoscopic 3D. Using hand-held controllers and with full-room tracking mode enabled on an area of around 2.5 $\times$ 2.5 meters, the participants were able to navigate through the EvoStreets in 3D space.

As already stated in Section I, both environments were using the same visualization engine and applied the same visual mappings (which are described in Section III-B in more detail) to the components of the EvoStreets, minimizing the threat that differences observed between the 2.5D and VR environments are the result of different implementations.

B. *Visual Mappings*

The EvoStreets visualization engine SCOOP supports the mapping of arbitrary metrics to the size-related components width, height, and length of buildings. In addition, a color gradient may be applied to depict a forth metric. Relations between elements can be expressed by connecting buildings with hierarchically bundled edges. In our previously conducted experiment, existing Java projects were analyzed. The corresponding files were depicted as buildings. Streets, in turn, showed the hierarchy of the directory layout. The clone rate (the fraction of tokens contained in a clone put in relation to the overall number of tokens) of each file was equally mapped to the width, height, length, and color gradient components, allowing to view the same information from all angles within the EvoStreets. Cloning between files was depicted using hierarchical edge bundles.

C. *Tasks*

The participants had to solve three different tasks (in the area of clone detection) which were designed based on the three main user goals proposed by Basit et al. [42]. Each of the tasks was deployed in one out of three different environments: 2D, 2.5D, and VR—the 2.5D and VR environments were introduced in Section III-A. Accordingly, each participant used all three environments once. In the following, we give a brief description.

Task 1 (Figure 2a) In the first task, two EvoStreets (A and B) representing two different, independent software systems were placed next to each other. By counting and comparing the number of non-recursive connections (depicted by the hierarchical bundled edges), the participants had to determine which contains more fragments cloned in other source files.

Task 2 (Figure 2b) In the second task, the EvoStreets B of task 1 was subdivided into six subsystems (which were depicted by colored areas). The participants had to count and compare the connections between the subsystems and determine which pair of subsystems shares the most connections.

Task 3 (Figure 2c) Similar to task 2, a software system subdivided into five subsystems was presented. By counting and comparing the red colored blocks (source files containing clones), the participants had to determine which subsystems contains the majority of source files affected by cloning.

We note that the answers of our tasks could be computed by algorithms and presented to the participants without the necessity to look at or even interact with the EvoStreets. However, visual analytics acknowledge that specific questions are raised only through interacting with a visualization. That being said, developers of visualization tools cannot know in advance all possible questions. Furthermore, visual analytics does not claim to provide all possible answers, but to give first impressions by visualizing data in a structured way, thus, being helpful at an early stage of data analysis where specific problems and hypotheses are not known yet.

D. Data Extraction

While running the experiment, the positional data of the participants were recorded and stored in CSV files. The output files are structured as follows:

Participant	X	Y	Z	Delta (MS)
...	...	...	...	...
p05	46974.429	62996.296	35450.210	11
p05	47049.300	62990.160	35635.566	11
p05	46996.183	63176.117	35584.589	3451
p05	46939.242	63360.578	35532.324	11
p05	46878.437	63543.417	35478.738	12
...	...	...	...	...

Whenever a participant changed his/her position (X, Y, Z) using the keyboard (2.5D environment) or hand-held controller (VR environment), the time (in milliseconds) how long he/she stayed at this position was recorded. Entries with a very low time (circa 11 milliseconds) are the result of the sampling rate of the recorded movement. Changing the view through rotation does not alter the currently taken position. Thus, viewpoints are not split when a participant used the mouse (2.5D environment) or moved his/her head (2.5D environment).

In order to extract appropriate viewpoints, that is, positions where the participants stayed to observe the EvoStreets, a threshold above which an entry is considered a viewpoint must be chosen. Yet, there is no optimal value. If a too low threshold is chosen, the extracted viewpoints may contain false positives, for example, locations where a participant looked at the keyboard to find a specific key. On the other hand, if a too high threshold is chosen, relevant viewpoints may be missed. We note that our results may vary depending on the chosen threshold and, thus, decided to take three different values:

- ≥ 1000 This threshold may contain false positives because some of our participants were neither familiar with the controls of the 2.5D and VR environments nor the concepts of EvoStreets in general.
- ≥ 5000 We expect that the entries filtered with this threshold do not contain false positives but, at the same time, miss several relevant viewpoints.
- ≥ 3000 This threshold is a compromise between the first two. It may contain a few false-positives and may miss some relevant entries.

The tasks that had to be solved by the participants were derived from a typical clone visual-analytics context. In order to find an answer, blocks and edges had to be counted and compared. It was not possible to identify the relevant elements of an EvoStreets without examining it. Hence, we suspect that the smallest threshold (1000) does not filter out viewpoints that were decisive for finding an appropriate answer.

E. Hypotheses

Our first research question *RQ1* asks whether viewpoints are affected by the two different environments 2.5D and VR. In previous case studies [7], [8] we found indications that movement was less extensive in the VR environment for some of the tasks. That is, path length was significantly shorter in task 2 and occupied volume was significantly smaller in task 2 and task 3—the p-value of task 1 was 0.07, thus, relatively close to our chosen significance of 0.05. Based on these results, we suspect that in the 2.5D environment the participants chose locations with a broader overview, allowing them to observe larger parts of the EvoStreets (as opposed to the VR environment). With regard to the position of viewpoints, we formulate our first hypothesis:

H1.H In the VR environment viewpoints are located closer to the EvoStreets.

Although path length and occupied volume give no direct information about the number of viewpoints, the average speed metric (which puts the length of a path in relation to the time that was required to move along this path) allows to make further assumptions. In our previous study we found that average speed in the VR environment was significantly less in task 2 and task 3, indicating that i) more viewpoints were taken while exploring the EvoStreets and ii) the residence time at individual viewpoints was longer (movement speed was the same for both environments, thus average speeds can be compared with each other). With regard to *i*, we postulate the following hypothesis (we will address *ii* below):

H1.N In the VR environment more viewpoints were taken.

As described in Section III-A, the way how participants could move themselves in the EvoStreets was fundamentally different between both environments. In the 2.5D environment, movement was provided by means of a keyboard (translation) and mouse (rotation). In contrast, in the VR environment one could change his/her position (translation) by using a hand-held controller and adapt his/her view (rotation) through head and body rotation—in both environments movement was implemented as a fluid motion in a particular direction. This may have an effect on the distances between the chosen viewpoints. For example, due to having a background in 2.5D computer gaming, many developers already have experience with navigating through 3D space using a keyboard (we mapped translation to the WASDQE keys, which is a widespread pattern in computer gaming) and mouse. When running our experiment, we noticed that most participants were very skilled in dealing with the controls of the 2.5D environment, allowing them to target certain viewpoints more precisely than in the VR environment, though, the majority already had experience with movement in VR. In addition, we assume that viewpoints in the 2.5D environment are located farther away from the EvoStreets (hypothesis *H1.H*), making it necessary to overcome a greater distance to reach a position which provides a new visual perspective. Taking this into account, our next hypothesis is as follows:

H1.D The distance between consecutive viewpoints is shorter in the VR environment.

As already mentioned above (*H1.N*, assumption *ii*), we assume that residence time is longer in the VR environment because average speed was significantly less in task 2 and task 3. Therefore, the hypothesis is as follows:

H1.T Residence time is longer in the VR environment.

Our second research question *RQ2* asks whether the position and residence time of viewpoints change over time, and whether these changes show patterns that are specific to individual environments. The tasks that had to be solved by the participants are based on tasks which are common in the area of clone detection. To find an answer, the participants had to explore the entire EvoStreets—all three tasks could not be solved if one stayed at a single viewpoint. Accordingly, moving around to get an initial overview was inevitable. Once first findings were made, a closer look to particular elements of the EvoStreets was necessary. For example, in task 2 the participants first had to get a rough idea of how the EvoStreets is structured, which subsystems exist therein, and how these subsystems are delimited from each other. The time this process takes may vary between individual participants—some prefer to acquire as much overview as possible, others choose to step into the EvoStreets directly. Yet, gaining abstract information, for instance, the location of relevant hotspots, is essential. Once first impressions have been gathered, the number of connections between the subsystems had to be counted and compared in order to determine the pair of subsystems sharing the most clones. While gaining a first overview is best from a viewpoints located farther away, comparing specific information makes it necessary to get closer to the EvoStreets because a higher level of detail is required. Based on these assumption, we present our next hypothesis (for the 2.5D and VR environments respectively):

H2.TH As time progresses, the height of viewpoints becomes smaller.

Continuing with the previous example, some of the subsystems of the EvoStreets in task 2 could be excluded directly after gaining an initial overview because, without much effort, one could see that the number of edges leaving/entering (the edges were undirected, thus there is no difference) a subsystems was less than in other subsystems. To find final evidence regarding the pair of subsystems sharing the most connections, though, some pairs had to be analyzed more deeply because the answer to this question cannot be determined without precise examination of the traces of edges—making it necessary to move between these pairs. However, due to examining a subset of the entire EvoStreets only, movement is less extensive at this stage of analysis. Thus, our hypothesis is as follows (for the 2.5D and VR environments respectively):

H2.TD As time progresses, the distance between viewpoints becomes shorter.

Our last two hypotheses tie in with *H2.TH* and *H2.TD*, which claim that, for both environments, the height of viewpoints and the distance between consecutive viewpoints change over time. Another interesting question in this context is, whether these changes are bigger in one environment than the other. To put up a guess, we use the same arguments as already for hypothesis *H1.H*: If movement in VR is less extensive (and we found indications for that), there are reasons to assume that the change of the height of viewpoints and the change of the distance between consecutive viewpoints is bigger in the VR environment:

H2.THV The change of the height of viewpoints (*H2.TH*) is bigger in the VR environment.

H2.TDV The change of the distance between consecutive viewpoints (*H2.TDV*) is bigger in the VR environment.

The metrics adhered to the hypotheses of research question *RQ1* (height of viewpoints, number of viewpoints, distance between consecutive viewpoints, and residence time) extend our previous results and provide additional insights into whether and how human beholders in EvoStreets are affected by the 2.5D and VR environments. The hypotheses of research question *RQ2*, on the other hand, examine how viewpoints change over time, giving insights into patterns that are specific to a certain environment. Based on our findings, concepts that support the usage of a specific environment can be developed.

IV. Results

Based on the thresholds described in Section III-D, we extracted the viewpoints of all participant in the 2.5D and VR environments. We compared the resultant viewpoints between these environments with respect to the number of viewpoints, their height and residence time, and the distance between consecutive viewpoints. We also analyzed how height and distance changed over time and whether these changes are bigger in the VR environment. In the following, we first present our results (Section IV-A and IV-B) and then (Section IV-C) discuss our findings and propose visualization and user interaction concepts that adapt to the characteristics of the 2.5D and VR environments.

A. Comparing Viewpoints (H1.H, H1.N, H1.D, H1.T)

In order to validate our hypotheses *H1.H*, *H1.N*, *H1.D*, and *H1.T*, we compared the extracted viewpoints of the 2.5D and VR environments with each other using the Mann-Whitney-Wilcoxon test. Table II lists the corresponding p-values for each task and threshold. The smallest threshold (1000) shows several significant results: i) the height of viewpoints (*H1.H*) is less in the VR environment for task 1 and 3, ii) the distance between consecutive viewpoints (*H1.D*) is shorter in the VR environment for all three tasks, and iii) the residence time (*H1.T*) is longer in the VR environment for task 2. In our first case study [7], we analyzed the correctness of the answers given by the participants for each task and environment and found that, according to the rate of correctness, task 3 is by

TABLE II: P-values for hypotheses *H1.H*, *H1.N*, *H1.D*, and *H1.T* for each task and threshold.

		Threshold								
		1000			3000			5000		
Hypothesis	Description	1	2	3	1	2	3	1	2	3
H1.H	Height of viewpoints	**0.0000**	0.3665	**0.0000**	**0.0002**	**0.0420**	**0.0002**	**0.0061**	0.0639	**0.0065**
H1.N	Number of viewpoints	0.0824	0.2726	0.1700	**0.0158**	0.1038	0.1954	**0.0145**	**0.0252**	0.6767
H1.D	Distance of consecutive viewpoints	**0.0000**	**0.0000**	**0.0000**	0.0710	**0.0000**	**0.0023**	0.3070	**0.0120**	0.0879
H1.T	Residence time	0.2680	**0.0001**	0.3572	0.8797	0.2705	0.3185	0.9790	0.8471	**0.0247**

far the most difficult task whereas task 2 is clearly the most simplest one (task 1 is in between the two). With this in mind, one theory to explain result *i* is that VR leads to viewpoints with a lower height only if a tasks exceeds a certain level of difficulty. On that basis, result *iii* suggests that residence time in VR is longer only if a task falls below a certain level of difficulty. However, whether this theory is actually true, we cannot tell.

According to our results, we must reject *H1.N* for all three tasks and *H1.T* for two out of three tasks with threshold 1000 (we will discuss the thresholds 3000 and 5000 below). This is remarkable insofar as that these hypotheses were derived from previous results. That is, in our previous study we found indications that average speed was significantly less in the VR environment for task 2 and task 3. Thus, we expected that either the number of viewpoints in VR is greater or residence time in VR is longer (or a combination of both). Based on these findings we assume that our minimum threshold (1000) filters out several locations where participants did not move. Anyhow, we do not consider these locations as appropriate viewpoints because staying at a location for less than one second does not seem to be suitable for observing relevant elements in EvoStreets. Yet, this could be an indication that movement in the VR environment tends to be more clipped, that is, in VR one stays at several locations for a short time to adapt his/her navigation. This assumption is conclusive because many developers have a background in computer gaming and, thus, are experienced with navigation in 2.5D environments. Gaming in VR, on the other hand, is a relatively new concept. Although the mechanics of the VR environment could be trained in a preliminary training level (and the fact that about half of the participants already had advanced know-how with EvoStreets in VR environments), the lack of experience could lead to clipped paths. In addition, these results could be an indication that moving with hand-held controllers is unnecessarily cumbersome, regardless of prior experience.

Some of the results fall above our chosen level of significance of 0.05 with higher thresholds (3000 and 5000). For example, hypothesis *H1.T* can be accepted for task 2 with threshold 1000, but must be rejected with threshold 3000 (p-value is 0.2705). To our surprise, with threshold 5000 the p-value becomes very high (0.8471). Similarly, *H1.D* is significant for all tasks with threshold 1000, for two out of three tasks with threshold 3000, and for only one task with threshold 5000. Another interesting observation is that *H1.N* is significant for task 2 with threshold 5000 (p-value is 0.0252), although it is not significant with lower thresholds (p-value is 0.2726 with threshold 1000 and 0.1038 with threshold 3000). We examined the viewpoints of all participants and found that the number of viewpoints decreases very much with increasing threshold. For example, one of the participants had 10 viewpoints with threshold 1000, 3 viewpoints with threshold 3000, and only a single viewpoint with threshold 5000 (task 3 in the 2.5D environment). Thus, we suspect that the thresholds 3000 and 5000 are too aggressive in filtering out valid viewpoints.

B. Viewpoints Over Time (H2.TH, H2.THV, H2.TD, H2.TDV)

Due to our findings regarding the thresholds 3000 and 5000 (they seem to be too aggressive in filtering out valid viewpoints), we calculated the change of the height of viewpoints (*H2.TH*) and the change of the distance between consecutive viewpoints (*H2.TD*) using our lowest threshold (1000) only. To validate our hypotheses, we aggregated viewpoints as follows. For each participant p of environment 2.5D/VR:

1. Normalize the duration of task solving of p to the [0, 1] interval.
2. Subdivide this interval into five phases of equal size $P_1 \dots P_5$ (corresponding to the comprehension stages described below).

$3a$. Calculate the median of the height of all viewpoints of p within phase P_n for $1 \leq n \leq 5$. This yields h_{pP_n}.

$3b$. Calculate the median of the distance of all consecutive viewpoints of p within phase P_n for $1 \leq n \leq 5$. This yields d_{pP_n}.

By grouping viewpoints into different phases, we are able to calculate the changes described in *H2.TH* and *H2.TD* for each transition from one group of viewpoints to another. We decided to subdivide task solving into five phases to represent five different stages of analysis—analogously to Sillito et al. [43] who organized 44 types of questions programmers ask into four different categories based on the kind and scope of information required to answer a question at a certain stage of source code comprehension. We call these stages: *orientation*, *exploration*, *examination*, *comparison*, and *validation*. In the first phase (orientation), the participants try to locate themselves in the EvoStreets (based on their starting point, cf. Figure 2). In the second phase (exploration), the participants start to get a rough overview of the EvoStreets to identify groups of elements which might be relevant for their task. In the third phase (examination), each group of elements is analyzed with regard to the task that must be solved. In the forth phase (comparison), the results of the different groups of elements are compared with each other. In the final phase (validation) the final result is determined and checked. We note that subdividing task solving into groups of equal time ranges is a naïve approach, but, due to the lack of additional information such as audio or video files that record how participants felt at a certain time, we are unable to weight the phases. Yet, we believe that this preliminary approach is suitable to get first indications on whether and how viewpoints change over time.

Mapping a viewpoint to a single phase is not always possible. There may be situations in which a viewpoint's duration (its residence time) causes the viewpoint to be located in multiple phases—for example, if a viewpoint starts in the phase exploration and ends in the phase examination. In such cases, it is not clear to which phase a viewpoint should be mapped. Splitting viewpoints and mapping them to mutiple phases assumes that participants change phases without movement. However, we suspect that phase transition does not occur in the middle of viewpoints. To cover different aspects, we decided to examine three different mapping strategies: Mapping the start, center, and end of a viewpoint's lifespan.

Our hypothesis *H2.TH* states for both environments that, as time progresses, the height of viewpoints becomes smaller. Due to starting at the ground of an EvoStreets (cf. Figure 2), it is to be expected that viewpoints in the exploration phase are located farther away from the EvoStreets than in the orientation phase. However, in subsequent phases a downtrend is postulated. Figure 3 shows the median of the heights of the viewpoints (cf. 3*a*.) as boxplots for all phases, tasks and mappings. To our surprise, the median of the heights depicted in the boxplots do not show a clear downtrend. On the contrary. There seem to be several uptrends between phases in the 2.5D environment for task 1 and task 3. Especially in task 3 participants seem to prefer viewpoints located farther away in the examination (3), comparing (4), and validation (5) phases. For the VR environment there is no clear trend for any of the tasks and mappings—yet, some differences can be seen in task 3. To validate our findings, we ran the Mann-Whitney-Wilcoxon test. With regard to downtrends, none of the p-values is significant. With regard to uptrends, the differences between phases exploration (2) and examination (3) for task 3 in the 2.5D environment are statistically significant (p-values are as follows: *Start* 0.0204, *Center* 0.0060, and *End* 0.0058).

Figure 4 shows the median of the distances between consecutive viewpoints (cf. 3*b*.) as boxplots for all phases, tasks, and mappings. As for the height of viewpoints, a clear downtrend (as postulated by hypothesis *H2.TD*) is not present. Only for task 1 in the 2.5D environment with mapping *Start* and *End* indications for a downtrend between phases exploration (2), examination (3), and comparing (4) can be seen. Similar to Figure 3, task 3 shows several uptrends for the 2.5D environment in later phases. Considering our results regarding the change of height, this is to be expected because if participants move upwards at later stages of analysis, the distance between viewpoints with a different perspective also becomes greater. Again we ran the Mann-Whitney-Wilcoxon test to validate our results and found significant p-values. The uptrend between phases orientation (1) and exploration (2) with mapping *Start* and *End*, and between phases exploration (2) and examination (3) with mapping *Center* for task 1 in the 2.5D environment are significant (p-values are 0.0137, 0.0439, and 0.0279 respectively). With regard to downtrends, only task 1 in the VR environment between phases examination (3) and comparing (4) with mapping *Start* is significant (p-value is 0.0216). That being said, we must reject our hypotheses *H2.TH* and *H2.TD*. Likewise, we could not find statistically significant p-values for our hypotheses *H2.THV* and *H2.TDV*.

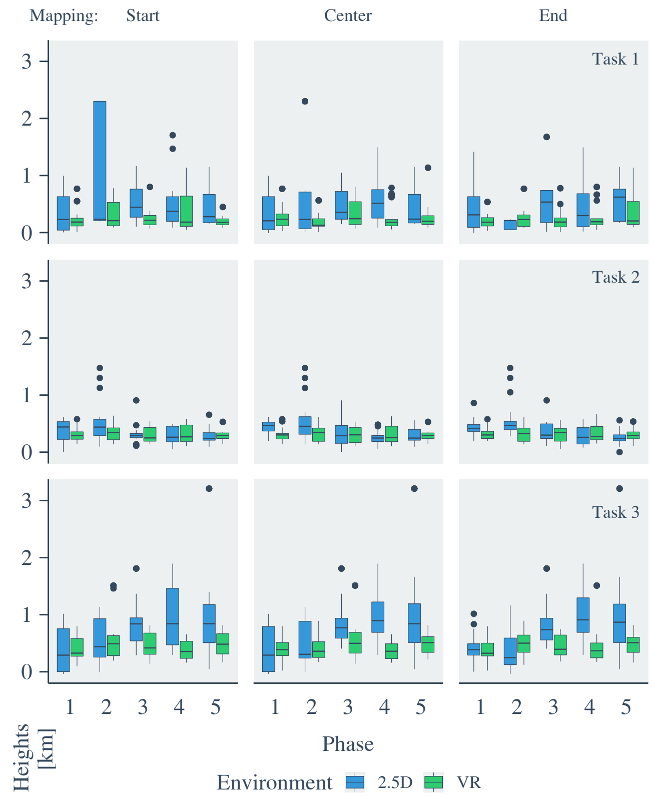


Fig. 3: Height of viewpoints in different phases.

C. Discussion

In the following, we discuss our results and propose six visualization and user interaction concepts that, according to our findings, adapt to the characteristics of the 2.5D and VR environments. Our list does not claim to be complete, but it is intended to provide a first draft for possible improvements. Furthermore, some of the concepts might be useful for both environments, yet, we suspect that they are more profitable for one environment than for the other.

Zoom function (2.5D): We found indications that in the 2.5D environment the chosen viewpoints are farther away from the EvoStreets (*H1.H*). This could be an indication that human beholders prefer to take positions with a broader overview. Furthermore, we could not find evidence that, as time progresses, the height of viewpoints in the 2.5D environment

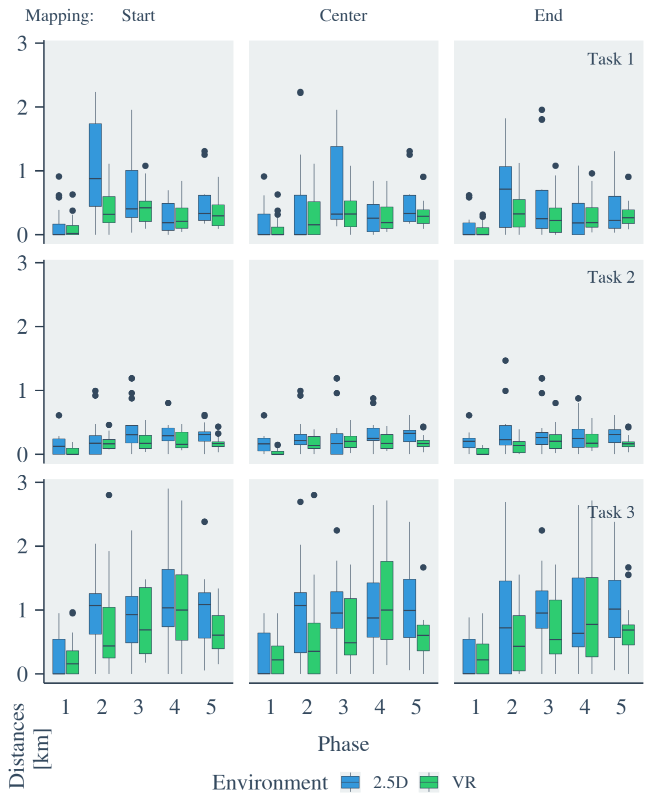


Fig. 4: Distance between consecutive viewpoints in different phases.

changes towards the EvoStreets (*H2.TH*). If a user observes an EvoStreets from farther away, and prefers to maintain a certain distance, several details, such as the exact number of ingoing and outgoing edges, can be seen less well or not at all. To compensate this, it might be beneficial to add a zooming function (similar to a binocular) to the 2.5D environment, which allows to select a portion of an EvoStreets. The selected elements can be visualized in an enlarged form using an overlay. This concept can be extended by supporting multiple overlays at once so that different parts of an EvoStreets can be compared simultaneously.

Screenshot function (VR): While in the 2.5D environment the participants seem to prefer viewpoints with a broader overview, viewpoints in the VR environment were located closer to the EvoStreets (*H1.H*). On the one hand, concrete details, such as a fine grained texture applied onto the blocks, are easier to see. On the other hand, comparing elements located at different places, for example in different subsystems, is less convenient because one has to go back and forth between these elements. By adding a screenshot function, the currently shown view can be saved for later use. In addition, multiple screenshots can be stored in a queue or stack. This allows to compare elements located at different places without the necessity to move between these places. One could argue that the zooming function proposed above for the 2.5D environment may also be useful in this case. However, due to staying closer to the EvoStreets, selecting relevant parts is more difficult because the lower a viewpoint is, the higher the probability that blocks are occluded by other blocks gets.

Teleport function (2.5D): We also found indications that the distance between consecutive viewpoints is greater in the 2.5D environment than in the VR environment (*H1.D*). That is, one has to overcome a greater distance to reach the next (or previously visited) viewpoint. To reduce the time that is necessary to travel between viewpoints, a teleport function providing two different modes can be added. The first mode allows to jump between already visited viewpoints. The second mode could try to estimate useful viewpoints, for example, viewpoints with a new perspective, that have not been visited yet. We could not find enough evidence that the distance between consecutive viewpoints changes significantly over time. Thus, this function might be useful at all stages of an analysis.

Visualize aggregated metrics (2.5D): EvoStreets map different software-related aspects (metrics) to visual components. The farther away an EvoStreets is observed the more abstract the visualized elements get. For example, the hierarchical structure of an entire software system can be best seen from a viewpoint located far away. However, the details of a certain building or the exact route of an edge cannot be perceived very well. To compensate this, we proposed a zooming function. Yet, the fact that EvoStreets are observed in a more abstract manner can be utilized by introducing a visual component which groups elements and depicts an aggregated metric. In task 2 and 3 of our experiment we used a colored plane to visually subdivide subsystems. This concept can be extended by giving the color a semantic. For instance, the average clone rate of all files of a subsystem could be mapped to the color of the corresponding plane. If EvoStreets in the 2.5D environment are observed from farther away (*H1.H*), the aggregated values can be compared across all parts of a system easily. This technique may also be useful in a VR environment, but based on our findings we believe it is less beneficial than in a 2.5D environment because one has to move upwards.

Minimap (VR): In order to examine all aspects of a software systems, an abstract as well as a detailed view on the elements of an EvoStreets is necessary. According to our results, the viewpoints taken by the participants in the VR environment are located closer to the EvoStreets (at least for two out of three tasks), indicating that a more detailed view is preferred. Yet, it could be beneficial to provide a quick overview of the entire system. Based on our findings, we propose to provide a minimap visualization. A minimap could depict an EvoStreets (or larger parts of it) in bird's eye view, similar to the 2D environment we used in our previous experiment [7]. Furthermore, a minimap may enhance orientation in VR by highlighting a user's current position. Minimaps are well known in 2.5D computer gaming and have proven their usefulness. Anyhow, we suspect that minimaps are of higher relevance when visualizing EvoStreets in a VR environment.

Marking blocks (VR): To further enhance orientation in VR, a function that allows to mark certain blocks could be implemented. Blocks with a mark can be highlighted with a beam of light pointing to the sky. There are also potential synergies with the minimap visualization described above and the teleport function proposed for the 2.5D environment. By showing marked blocks in a mimimap, the current position can be determined more easily. Being able to teleport between marked blocks could be useful to travel large distances quickly.

D. Threats to Validity

In this section, we want to point out threats that may affect validity. Several of those threats are inherited from our original case studies [7], [8], others are specific to our analysis presented in this paper.

Internal validity: To ensure that the demographic characteristics of the participants and the order of the tasks does not affect results, the participants were randomly divided into six distinct groups using a balanced latin-square order, that is, each group was using a different combination of task and environment. Most of the participants were already familiar with head-mounted displays and hand-held controllers, so that the novelty effect of the VR environment should not have much impact. Yet, some of the participants had no prior experience. For minimizing novelty effects, every participant had the chance to make himself/herself familiar with the visualization and interaction concepts in a training level for both of the 2.5D and VR environment. Furthermore, there was no time pressure, neither in the training level nor in the actual tasks. That being said, visualizing EvoStreets in VR is a relatively new concept (and VR is still closely associated with gamification), so we cannot exclude that participants spent more time in the VR environment than necessary.

While running the experiment, the positional data of the participants were automatically recorded by the experimental system. Although claimed objective, the experimenter could have influenced these data by his or her personality, different speeches or fatigue. In addition, all participants received the same tasks (only the order of the presented environments changed) and even though we told the participants to not talk with each other about the contents of the experiment (many of the participants knew each other), we cannot rule out that they still exchanged impressions.

Our sample size is relatively small and its composition is unknown, so we cannot use parametric statistical tests. Instead, we used non-parametric tests, eliminating the effects of outliers. However, this has the disadvantage that these tests break down the concrete quantifiable values to rank number and, therefore, ignore differences.

External validity: The tasks that had to be solved by the participants are based on different clone detection tasks, as it is well known that cloning is a common issue in software. The layout generated by the EvoStreets is characteristic for the analyzed Java systems and the metrics that were mapped to the visual components. Accordingly, our results may refer to only these factors.

Construct validity: In order to extract suitable viewpoints (and their corresponding residence time), we had to filter measures from the recorded movement data using different thresholds (1000, 3000, and 5000 milliseconds). Although we suspect that these thresholds are adequate, we cannot rule out that our findings may vary with different values. To analyze the change of viewpoints over time, we subdivided task solving into different phases of equal length. Due to the lack of additional information, we were unable to weigh these phases though. Similar to the thresholds we chose for viewpoint extraction, our results may vary if phases are not equally weighted or if task solving is subdivided into a different number of phases.

V. Conclusion

Based on the positional data recorded in a previous controlled experiment, we analyzed how participants observed EvoStreets in a 2.5D and virtual reality environment. To measure observation we extracted viewpoints. A viewpoint is a position in 3D space where one stays a certain amount of time without movement (we call this time *residence time*). In order to filter out locations where participants stayed without observation, but, for instance, to adapt their navigation, we had to choose a threshold (regarding residence time) above which a measure is considered a viewpoint. However, there is no optimal value. If a too low threshold is chosen, the extracted viewpoints may contain several false positives. If a too high threshold is chosen, relevant viewpoints may be missed. Thus, we decided to filter the recorded data using three different values: 1, 3, and 5 seconds. In addition, we introduced metrics to capture different aspects of observation, namely, number of viewpoints, height of viewpoints, distance between consecutive viewpoints, and residence time, and compared these metrics between the 2.5D and virtual reality environments. Furthermore, we analyzed whether and how viewpoints change over time with respect to the height of viewpoints the distance between consecutive viewpoints. We found that the height of viewpoints is significantly less in the virtual reality environment for almost all tasks and thresholds. Likewise, we found that the distance between consecutive viewpoints is significantly less in the virtual reality environment for almost all tasks with threshold 1 and 3 seconds. According to our results, however, we consider the thresholds 3 and 5 seconds as too aggressive in filtering out valid viewpoints. Thus, we analyzed the change of viewpoints with our lowest threshold of 1 second only. Using this threshold, we grouped the viewpoints of the participants into five different phases, each of them representing a different stage of the participants' task solving. Our hypotheses state that, as time progresses, the height of viewpoints become smaller and the distance between consecutive viewpoints become shorter. Although we could not find enough evidence to confirm our hypotheses, we found indications that the height of viewpoints becomes larger at later phases of an analysis in the 2.5D environment for two out of three tasks.

Finally, we proposed six different visualization and user interaction concepts that, according to our results, adapt to the characteristics of the 2.5D and virtual reality environments. Our list of concepts does not claim to be complete, but is intended to provide a first draft for possible improvements. These findings may be the starting point of future research.

References

[1] R. Wettel and M. Lanza, "Visualizing software systems as cities," in *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, June 2007, pp. 92–99.

[2] ——, "Codecity: 3d visualization of large-scale software," in *Companion of the 30th International Conference on Software Engineering*, ser. ICSE Companion '08. New York, NY, USA: ACM, 2008, pp. 921–922. [Online]. Available: http://doi.acm.org/10.1145/1370175.1370188

[3] ——, "Visual exploration of large-scale system evolution," in *2008 15th Working Conference on Reverse Engineering*, Oct 2008, pp. 219–228.

[4] F. Steinbrückner and C. Lewerentz, "Representing development history in software cities," in *ACM International Symposium on Software Visualization*. ACM, 2010, pp. 193–202.

[5] F. Steinbrückner, "Consistent software cities: supporting comprehension of evolving software systems," Ph.D. dissertation, Brandenburgische Technische Universität Cottbus, 06 2013. [Online]. Available: https://opus4.kobv.de/opus4-btu/frontdoor/index/index/docId/1681

[6] S. S. Chance, F. Gaunet, A. C. Beall, and J. M. Loomis, "Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration," *Presence: Teleoper. Virtual Environ.*, vol. 7, no. 2, pp. 168–178, 1998.

[7] M. Steinbeck, R. Koschke, and M.-O. Rüdel, "Comparing the evostreet visualization technique in two- and three-dimensional environments—a controlled experiment," in *International Conference on Program Comprehension*. IEEE Computer Society Press, 2019, pp. 231–242.

[8] S. Marcel, K. Rainer, and M. Rudel, "Movement patterns and trajectories in three-dimensional software visualization," in *2019 19th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, Sep. 2019, pp. 163–174.

[9] M. Rüdel, J. Ganser, and R. Koschke, "A controlled experiment on spatial orientation in vr-based software cities," in *Working Conference on Software Visualization*, Sep. 2018, pp. 21–31.

[10] R. Koschke, "Software visualization in software maintenance, reverse engineering, and re-engineering: a research survey," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, no. 2, pp. 87–109, 2003.

[11] S. Carpendale and Y. Ghanam, "A survey paper on software architecture visualization," University of Calgary, Technical Report 2008-906-19, 2008.

[12] A. R. Teyseyre and M. R. Campo, "An overview of 3d software visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 1, pp. 87–105, 2009.

[13] S. Bassil and R. K. Keller, "A qualitative and quantitative evaluation of software visualization tools," in *Proceedings of the Workshop on Software Visualization*. IEEE, 2001, pp. 33–37.

[14] S. Bassi and R. K. Keller, "Software visualization tools: Survey and analysis," in *International Workshop on Program Comprehension*. IEEE, 2001, pp. 7–17.

[15] P. Caserta and O. Zendra, "Visualization of the static aspects of software: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 7, pp. 913–933, 2011.

[16] S. Diehl, Ed., *Software Visualization: International Seminar Dagstuhl Castle, Germany*, ser. Lecture Notes in Computer Science. Springer, 2001.

[17] S. Diehl, *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. Springer, 2010.

[18] 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, 2018. [Online]. Available: https://doi.org/10.1016/j.jss.2018.06.027

[19] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the Conference on Visualization*. IEEE Computer Society Press, 1991, pp. 284–291. [Online]. Available: http://dl.acm.org/citation.cfm?id=949607.949654

[20] W. Scheibel, C. Weyand, and J. Döllner, "Evocells - A treemap layout algorithm for evolving tree data," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2018, pp. 273–280.

[21] D. H. R. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 741–748, Sep. 2006.

[22] ——, "Visualization of graphs and trees for software analysis," Ph.D. dissertation, Technical University of Delft, 2009.

[23] J. Vincur, P. Navrat, and I. Polasek, "Vr city: Software analysis in virtual reality environment," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, July 2017, pp. 509–516.

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

[25] N. Capece, U. Erra, S. Romano, and G. Scanniello, "Visualising a software system as a city through virtual reality," in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis, P. Bourdot, and A. Mongelli, Eds. Cham: Springer International Publishing, 2017, pp. 319–327.

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

[27] T. Panas, R. Berrigan, and J. Grundy, "A 3d metaphor for software production visualization," in *International Conference Information Visualization*. IEEE, 2003, pp. 314–319.

[28] T. Panas, T. Epperly, D. Quinlan, A. Saebjornsen, and R. Vuduc, "Communicating software architecture using a unified single-view visualization," in *Engineering Complex Computer Systems, 2007. 12th IEEE International Conference on*. IEEE, 2007, pp. 217–228.

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

[30] P. Khaloo, M. Maghoumi, E. Taranta, D. Bettner, and J. Laviola, "Code park: A new 3d code visualization tool," in *Working Conference on Software Visualization*. IEEE, 2017, pp. 43–53.

[31] L. Merino, J. Fuchs, M. Blumenschein, C. Anslow, M. Ghafari, O. Nierstrasz, M. Behrisch, and D. A. Keim, "On the impact of the medium in the effectiveness of 3d software visualizations," in *Working Conference on Software Visualization*. IEEE, 2017, pp. 11–21.

[32] A. Schreiber and M. Brüggemann, "Interactive visualization of software components with virtual reality headsets," in *Working Conference on Software Visualization*. IEEE, 2017, pp. 119–123.

[33] K. Ogami, R. G. Kula, H. Hata, T. Ishio, and K. Matsumoto, "Using high-rising cities to visualize performance in real-time," in *Working Conference on Software Visualization*. IEEE, 2017, pp. 33–42.

[34] F. Fernandes, C. S. Rodrigues, and C. Werner, "Dynamic analysis of software systems through virtual reality," in *Symposium on Virtual and Augmented Reality*, Nov. 2017, pp. 331–340, in Spanish.

[35] L. Merino, M. Hess, A. Bergel, O. Nierstrasz, and D. Weiskopf, "Perfvis: Pervasive visualization in immersive augmented reality for performance awareness," in *ACM/SPEC International Conference on Performance Engineering*. ACM Press, 2019, pp. 13–16.

[36] D. A. Bowman, E. T. Davis, L. F. Hodges, and A. N. Badre, "Maintaining spatial orientation during travel in an immersive virtual environment," *Presence: Teleoper. Virtual Environ.*, vol. 8, no. 6, pp. 618–631, 1999. [Online]. Available: http://dx.doi.org/10.1162/105474699566521

[37] B. E. Riecke, D. W. Cunningham, and H. H. Bülthoff, "Spatial updating in virtual reality: the sufficiency of visual information," *Psychological Research*, vol. 71, no. 3, pp. 298–313, May 2007. [Online]. Available: https://doi.org/10.1007/s00426-006-0085-z

[38] J. W. Regian, W. L. Shebilske, and J. M. Monk, "Virtual reality: An instructional medium for visual-spatial tasks," *Journal of Communication*, vol. 42, no. 4, pp. 136–149, 1992. [Online]. Available: http://dx.doi.org/10.1111/j.1460-2466.1992.tb00815.x

[39] R. A. Ruddle, S. J. Payne, and D. M. Jones, "Navigating large-scale virtual environments: what differences occur between helmet-mounted and desktop displays?" *Presence: Teleoperators & Virtual Environments*, vol. 8, no. 2, pp. 157–168, 1999.

[40] R. A. Ruddle and P. Péruch, "Effects of proprioceptive feedback and environmental characteristics on spatial learning in virtual environments," *International Journal of Human-Computer Studies*, vol. 60, no. 3, pp. 299 – 326, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1071581903001733

[41] L. Merino, A. Bergel, and O. Nierstrasz, "Overcoming issues of 3d software visualization through immersive augmented reality," in *Working Conference on Software Visualization*. IEEE Computer Society Press, 2018, pp. 54–64.

[42] H. A. Basit, M. Hammad, S. Jarzabek, and R. Koschke, "What do we need to know about clones? deriving information needs from user goals," in *International Workshop on Software Clones*. IEEE Computer Society Press, Mar. 2015, pp. 51–57.

[43] J. Sillito, G. C. Murphy, and K. De Volder, "Asking and answering questions during a programming change task," *IEEE Transactions on Software Engineering*, vol. 34, no. 4, pp. 434–451, July 2008.