

# Visualizing code reviews bottlenecks, from 2D to virtual reality

David Moreno-Lumbreras<sup>1,\*</sup>, Jesus M. Gonzalez-Barahona<sup>1</sup> and Gregorio Robles<sup>1</sup>

<sup>1</sup>Universidad Rey Juan Carlos, Madrid, Spain

## Abstract

**Background/Context:** Currently, the usual interface for visualizing data is based on 2D screens. Recently, devices capable of visualizing data while immersed in Virtual Reality (VR) scenes are becoming common. However, it has not been studied in detail to which extent these devices are suitable for interacting with data visualizations in the specific case of data about software development.

**Objective/Aim:** In this study, we propose VR dashboards that can aggregate and combine information that is usually represented in different 2D dashboards. Specifically, the combination of the timing of issues and pull requests in a single VR dashboard.

**Method:** We address our objective by combining a series of pre-built dashboards in KIBANA, with the help of GRIMOIRELAB and Bitergia into a single VR dashboard. We use GRIMOIRELAB for data extraction, KIBANA as a model to build the VR dashboard, and BABIAXR to build the dashboard in VR. We rely on web technologies, WEBXR and WEBVR standards for development, generating open, universal dashboards, capable of being viewed in modern browsers.

**Results:** A complete VR dashboard that combines information on Pull Requests and Issues, specifically their closing and processing time (timing). The differentiation between Issues and Pull Requests is done using two different color ranges. This approach will give way to a programmed study of the comparison of these VR dashboards with the common ones generated by KIBANA.

**Keywords:** virtual reality, dashboards, code review, pull requests, issues, data visualization

## Keywords

virtual reality, dashboards, controlled experiment, code review, Pull Request, Issues

## 1. Introduction

Software engineers mainly interact with source code using a keyboard and a mouse, typically viewing it on 2D monitors. This way of interacting does not take advantage of the many benefits of movement and perception that humans have. This is not the case of virtual reality (VR) immersion, where the subject works in a virtual 3D environment. In the last years, affordable

---

*Belgium-Netherlands Software Evolution Workshop, 2022*

\*Corresponding author.

†These authors contributed equally.

✉ david.morenolu@urjc.es (D. Moreno-Lumbreras); jesus.gonzalez.barahona@urjc.es (J. M. Gonzalez-Barahona);

gregorio.robles@urjc.es (G. Robles)

🌐 <https://dlumbrer.github.io/> (D. Moreno-Lumbreras); <https://gsyc.urjc.es/~jgb/> (J. M. Gonzalez-Barahona);

<https://gsyc.urjc.es/~grex/> (G. Robles)

🆔 0000-0002-5454-7808 (D. Moreno-Lumbreras); 0000-0001-9682-460X (J. M. Gonzalez-Barahona);

0000-0002-1442-6761 (G. Robles)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

VR devices have emerged, and new standards such as WEBXR [1] and WebGL [2] have become available, so that VR applications can be made portable to different platforms, and easily integrable with other applications and APIs. Therefore, we are now at a point where using VR for interacting with data visualizations is feasible in many environments.

In fact, in the specific case of software engineering, some scholars argue that the use of VR allows for environments that may make practitioners face lower learning curves, be more creative and achieve higher productivity [3]. However, to our knowledge, there is little evidence that VR can provide better, or even similar results to on-screen visualizations, in the specific case of software development data.

To explore visualizing software development data in VR, Bitergia<sup>1</sup> and Universidad Rey Juan Carlos have developed BABIAXR,<sup>2</sup> a toolset for visualizing data in 3D, both on-screen and in VR devices. Bitergia is a company offering commercial services in the area of software development analytics, and leads the development of GRIMOIRELAB<sup>3</sup> [4], a toolset including modules to retrieve data from many kinds of software repositories [5], store it in a database, and then process and analyze it, producing many different metrics. It also includes visualization modules that can be used to interact with the data via traditional, on-screen, web browsers. GRIMOIRELAB is now a project under the umbrella of the Linux Foundation CHAOSS community<sup>4</sup>. Data produced by GRIMOIRELAB can be fed to BABIAXR, so that the same pipeline can be used to visualize data on 2D screens and on VR devices. An example of a user story of how Bitergia uses software development data, in this case for the Xen project, can be found in [6].

The fact that we can design scenes to visualize exactly the same data, with very similar charts, in two different environments (2D charts on traditional screens in KIBANA, and 3D charts on VR devices using BABIAXR), gives us the chance of trying fair comparisons between both environments. In particular, we can design VR scenes with charts similar to the 2D, on-screen visualizations that Bitergia uses for its customers, and then run an experiment where subjects use one of those environments so that we can compare results.

The future aim is to test if VR immersion is at least as effective and efficient as 2D in-screen, for the visualization of the same data. The field of analysis for our study is Pull Requests and Issues activity. Pull Request, as part of modern code review [7, 8], is a software development activity that has been widely researched by academia in the last years [9, 10, 11]. It is of major interest to industry and practitioners as it is very human-intensive and often the cause of bottlenecks and inefficiencies [12].

By default, there are two different panels, developed by Bitergia in KIBANA, to display the timing of Issues and Pull Requests, forcing the user to move between pages to correlate the data, this is another goal to achieve by displaying both Issues and Pull Requests timing in the same VR Dashboard.

---

<sup>1</sup><https://www.bitergia.com>

<sup>2</sup><https://babiaxr.gitlab.io/>

<sup>3</sup><https://chaoss.github.io/grimoirelab/>

<sup>4</sup><https://chaoss.community>

## 2. 2D Visualizations

With the help of GRIMOIRELAB, Bitergia gathers and enriches software development data to create a set of pre-existing dashboards. In particular, there are five panels/dashboards whose main goal is to show several aspects of the Pull Request and Issues process. We focus on the timing panels (for Issues and Pull Requests) for analyzing bottlenecks on code reviews.

The timing panels (shown in Figure 1 and Figure 2), offer information about the time to close Pull Requests. This panel focuses on bringing information to the user about the evolution of the time to close a Pull Request/issue over time. This is intended to visually find previous bottlenecks in the development process.



**Figure 1:** Pull Requests Timing dashboard

The data represented is about the CHAOSS project<sup>5</sup>, an open-source project well known by the authors and the project that GRIMOIRELAB belongs to. All the data represented in KIBANA with GRIMOIRELAB have temporal information, allowing us to see the temporal evolution of this data, and see the status of the Pull Requests and Issues at different moments in time. KIBANA allows changing the time range with a menu, shown in Figure 3.

This refers to following the evolution of the software project that is represented, seeing how the code review processes have changed over time and being able to analyze bottlenecks and other problems.

<sup>5</sup><http://chaoss.biterg.io/>



Figure 2: Issues Timing dashboard

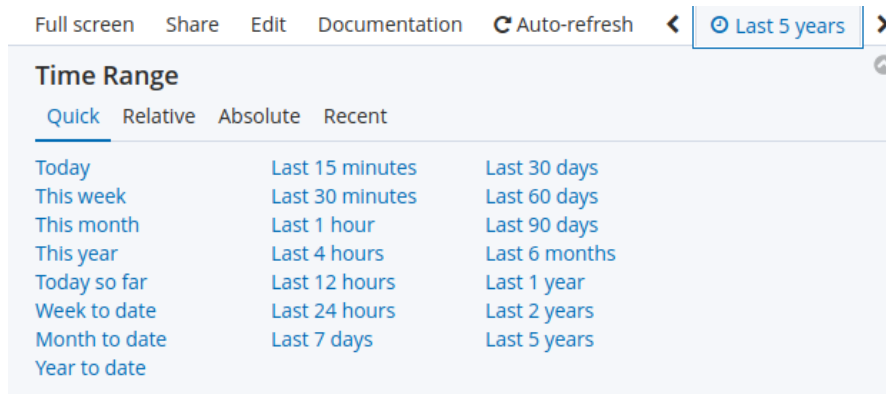


Figure 3: Kibana time range selector

### 3. VR Visualizations

BABIAXR<sup>6</sup> is a toolset for 3D data visualization in the browser. BABIAXR is based on A-FRAME,<sup>7</sup> an open web framework to build 3D, augmented reality (*i.e.*, AR), and VR experiences in the browser. A-FRAME extends HTML with new entities allowing to build 3D scenes as if they were HTML documents, using techniques common to any front-end web developer. A-FRAME is built

<sup>6</sup>BABIAXR: <https://babiaxr.gitlab.io>

<sup>7</sup>A-FRAME: <https://aframe.io>



on top of THREE.JS,<sup>8</sup> which uses the WebGL API available in all modern browsers.

BABIAXR extends A-FRAME by providing components to create visualizations, simplify data retrieval, and manage data (e.g., data filtering or mapping of fields to visualization features). Scenes built with BABIAXR can be displayed on-screen, or on VR devices, including consumer-grade headsets. Figure 4 shows a sample scene built with BABIAXR. BABIAXR is open source: Its source code is available on GITLAB<sup>9</sup> and it can be installed with NPM.<sup>10</sup>



**Figure 4:** Example of a BABIAXR Scene

### 3.1. From Source Code to a 3D Scene

BABIAXR includes components for query data from the ELASTICSEARCH database, having the same functionality as the KIBANA queries, allowing to do usual queries and aggregations to ELASTICSEARCH. Once the query is done, the data is parsed and formatted in the generic format that the BABIAXR visualizations use:

```
[
  {
    "field1": "aaa/bbb/ccc"
    "metric": x,
    "metric2": y,
    ...
  },
  ...
]
```

<sup>8</sup>THREE.JS: <https://threejs.org>

<sup>9</sup><https://gitlab.com/babiaxr/afame-babia-components>

<sup>10</sup><https://npmjs.org/package/afame-babia-components>

The scene to visualize this data is composed of a single HTML file, which uses the data retrieved by the ELASTICSEARCH queries. The HTML file imports all the dependencies (*i.e.*, A-FRAME and BABIAXR JavaScript packages) and defines the scene by including the corresponding elements and components: `babia-queries` to retrieve the ELASTICSEARCH data, `babia-treebuilder` and `babia-xxxx` which are the actual components to generate the visualizations. Each component has its own configuration, detailed in the documentation.<sup>11</sup> The listing below shows a sample scene, including some configuration parameters:

```
<a-scene id="scene">
  <a-entity id="rawdata"
    babia-queries="url: data.json">
  </a-entity>
  <a-entity id="barsviz"
    babia-bars="height: metric;
    x_axis: field; from: rawdata">
  </a-entity>
  <a-entity id="3Dbarsviz"
    babia-barsmap="from: rawdata;
    x_axis: field; z_axis: metric2; height: metric3">
  </a-entity>
  ...
</a-scene>
```

Figure 5 summarizes the complete workflow to produce a scene with BABIAXR starting from a GIT repository.

As we have already said, thanks to modern web technology, these scenes are fully usable in VR mode in any modern browser, including those that include VR goggles, Figure 6 shows a user visualizing data in a VR web scene created with BABIAXR.

### 3.2. VR Dashboard

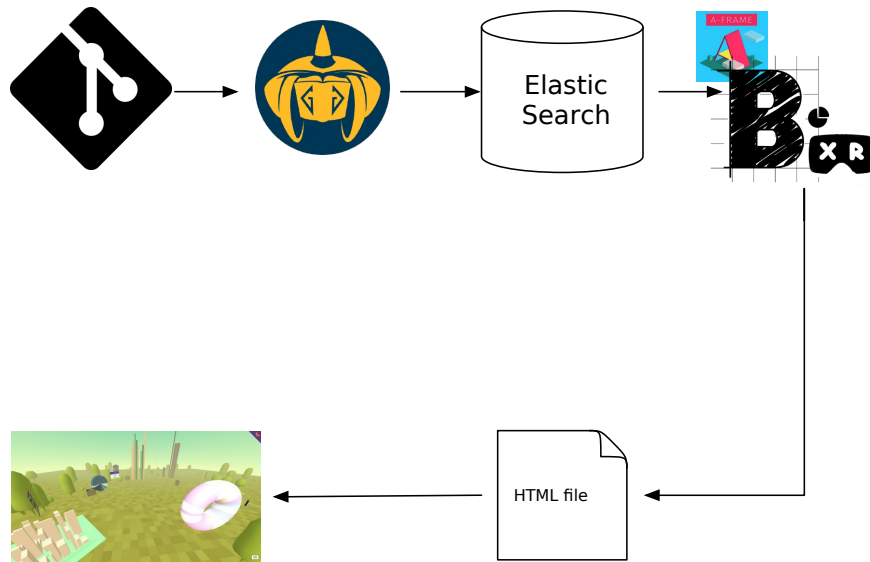
One of the biggest benefits of 3D and VR environments is that space is not as limited as in a 2D environment, specifically, in 3D and VR environments there is 360-degree space in all axes around a point, for example, the camera or the user entry point to the scene.

This feature allows various elements to be placed around a point and with a simple camera turn, which is transferred to a VR user's head turn, more information can be observed and correlated. So with BABIAXR, we have tried to bring together both KIBANA panels, both the Pull Request Timing panel and the Issues Timing panel, into a single VR scene, displaying both Issues and Pull Requests information in that scene. An example of this scene is shown in Figure 7.

We have used the museum metaphor (represent elements like in a museum) for the representation of this VR dashboard. In addition, we use two different color ranges (blue and red) to be able to distinguish between issue visualizations and Pull Request visualizations.

---

<sup>11</sup><https://gitlab.com/babiaxr/aframe-babia-components/-/tree/master/docs/APIs>



**Figure 5:** BABIAXR Workflow: From Source Code to a Scene

In addition, the feature to be able to change in different time ranges that include KIBANA, shown in Figure 3, has been completely developed for BABIAXR, allowing to change time ranges quickly. This is done with a KIBANA-like menu found on top of a VR headset controller, as shown in Figure 8

#### 4. Conclusions and Future Plan

One of the results of this study is to prepare a dashboard that combines both Issues and Pull Requests timing in a VR dashboard. This objective has been fulfilled and gives way to future comparative analysis of both environments. In the study derived from the registered report [13], it is expected to be able to use the dashboard shown in this study. We define the comparison as fairly as possible since every visualization in KIBANA has its visualization in the VR dashboard.

Given the little evidence about the convenience of using VR for visualizing and interacting with data, and in particular the relative lack of evidence comparing on-screen with VR immersion in software engineering, we consider that an experiment is important for the field. As VR devices become increasingly commonplace, as we can observe now in academia and several branches



**Figure 6:** Example of a BABIAXR Scene in VR with Oculus Quest 2 glasses

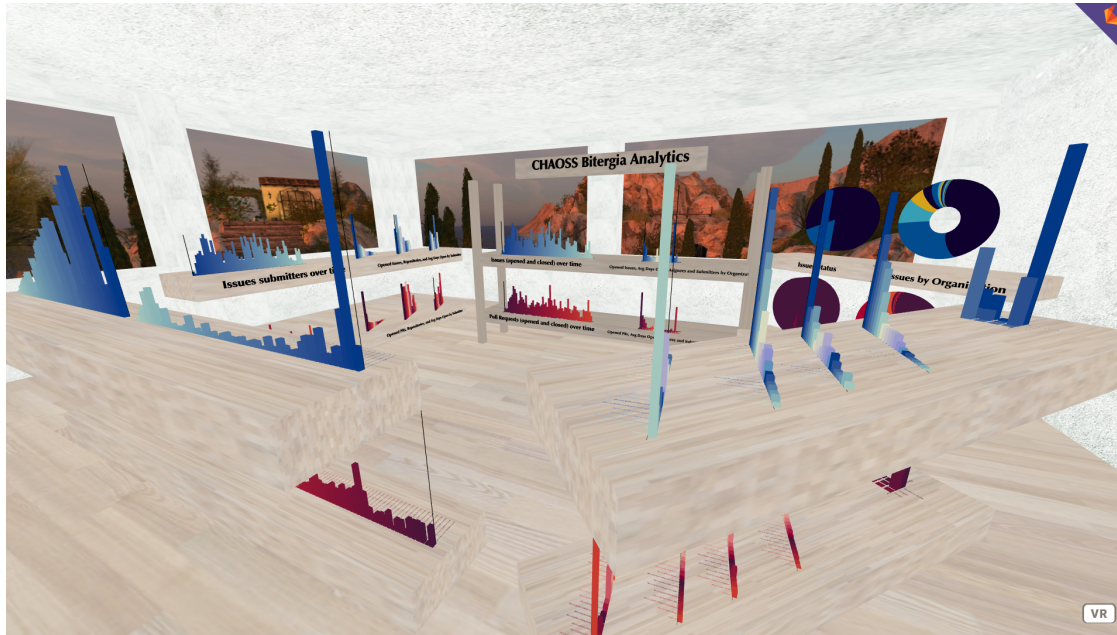
of science (medical, industrial engineering, etc.), inclusion in software engineering could be a real option, and it will be important to know as much as possible about how they impact practitioners, in comparison with traditional environments.

The analysis should be done with qualitative/quantitative studies, based on tasks and (possibly semi-structured) interviews with a sample of the people who will go through the experiment, to learn about the details of their experience. For avoiding as much as possible bias due to the different training with on-screen environments or VR environments, all subjects will go through a short training procedure, and we will mitigate other threats that can appear in the experiment.

As another future work, the development of BABIAXR will follow. A very strong point that it needs is the improvement of the interaction with the visualizations, KIBANA allows you to add filters as you click on different elements of the visualizations, that BABIAXR does not allow it at the moment, so it is the next step for have a VR dashboard in functionality equal to that of KIBANA. And in general, an improvement of the experience, interfaces, and usability after the feedback that we expect from the participants of the experiment and future ones.

## 5. Related work

Virtual reality has been shown to facilitate discovery in domains in which space plays an important role. For example in the field of brain tumors [14], perception of shapes and forms [15], paleontology [16], caves [17], and magnetic resonance imaging [18]. Data visualization in



**Figure 7:** Pull Requests and Issues timing in one VR dashboard

virtual reality allows the use of multidimensionality for abstract analysis, and even more so for large data sets.

Regarding the virtual reality data visualizations, Donalek *et al.* [19] presented immersive and collaborative data visualization using VR platforms, a big start with Unity. Before that, Bayyari *et al.* [20] presented that situational awareness in the visualization of data benefits from immersive, virtual reality display technology because such displays appear to support a better understanding of the visual information. More recently, Millais *et al.* [21] presented a comparison between 2D and VR visualizations, suggesting that users feel more satisfied and successful when using VR data exploration tools, thus demonstrating the potential of VR as an engaging medium for visual data analytics. Navigation in VR is another field of research, Drogemuller *et al.* [22] evaluated three-dimensional VR navigation technique for data visualizations and test their effectiveness with a large graph visualization.

There are other fields of study for data visualization, and Augmented Reality (AR) is one of them. Olshannikova *et al.* [23] presented an overview of the research issues and achievements in the field of Big Data visualization, and Natephra *et al.* [24] explored data visualization using AR, but using IoT sensors as data entypoint.

## References

- [1] B. Jones, M. Goregaokar, WebXR device API, W3C Working Draft, 2020. URL: <https://www.w3.org/TR/webxr/>.





**Figure 8:** BabiaXR time range selector

- [2] D. Jackson, J. Gilbert, WebGL 2.0 specification, Khronos Group Specification, 2020. URL: <https://www.khronos.org/registry/webgl/specs/latest/2.0/>.
- [3] A. Elliott, B. Peiris, C. Parnin, Virtual reality in software engineering: Affordances, applications, and challenges, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 2, IEEE, 2015, pp. 547–550.
- [4] S. Dueñas, V. Cosentino, J. M. Gonzalez-Barahona, A. del Castillo San Felix, D. Izquierdo-Cortazar, L. Cañas-Díaz, A. Pérez García-Plaza, Grimoirelab: A toolset for software development analytics, PeerJ Computer Science (2021). Accepted, publication pending.
- [5] S. Dueñas, V. Cosentino, G. Robles, J. M. Gonzalez-Barahona, Perceval: Software project data at your will, in: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, 2018, pp. 1–4.
- [6] D. Izquierdo, J. M. Gonzalez-Barahona, L. Kurth, G. Robles, Software development analytics for xen: Why and how, IEEE Software 36 (2018) 28–32.
- [7] A. Bacchelli, C. Bird, Expectations, outcomes, and challenges of modern code review, in: 2013 35th ICSE, IEEE, 2013, pp. 712–721.
- [8] P. Thongtanunam, S. McIntosh, A. E. Hassan, H. Iida, Review participation in modern code review, Empirical Software Engineering 22 (2017) 768–817.
- [9] O. Kononenko, T. Rose, O. Baysal, M. Godfrey, D. Theisen, B. De Water, Studying pull request merges: a case study of shopify’s active merchant, in: Proceedings of the 40th ICSE SEIP, 2018, pp. 124–133.
- [10] C. Maddila, C. Bansal, N. Nagappan, Predicting pull request completion time: a case study on large scale cloud services, in: Proceedings of the 2019 27th ESEC/FSE, 2019, pp. 874–882.
- [11] Y. Yu, H. Wang, V. Filkov, P. Devanbu, B. Vasilescu, Wait for it: Determinants of pull request evaluation latency on github, in: 2015 IEEE/ACM 12th working conference on

- mining software repositories, IEEE, 2015, pp. 367–371.
- [12] C. Sadowski, E. Söderberg, L. Church, M. Sipko, A. Bacchelli, Modern code review: a case study at Google, in: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice, 2018, pp. 181–190.
  - [13] D. Moreno-Lumbreras, G. Robles, D. Izquierdo-Cortazar, J. M. González-Barahona, To VR or not to VR: is virtual reality suitable to understand software development metrics?, CoRR abs/2109.13768 (2021). URL: <https://arxiv.org/abs/2109.13768>. arXiv:2109.13768.
  - [14] S. Zhang, C. Demiralp, D. Keefe, M. DaSilva, D. Laidlaw, B. Greenberg, P. Basser, C. Pierpaoli, E. Chiocca, T. Deisboeck, An immersive virtual environment for dt-mri volume visualization applications: a case study, in: Proceedings Visualization, 2001. VIS '01., 2001, pp. 437–584. doi:10.1109/VISUAL.2001.964545.
  - [15] C. Demiralp, C. Jackson, D. Karelitz, S. Zhang, D. Laidlaw, Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison, IEEE transactions on visualization and computer graphics 12 (2006) 323–30. doi:10.1109/TVCG.2006.42.
  - [16] B. Laha, D. Bowman, J. Socha, Effects of VR system fidelity on analyzing isosurface visualization of volume datasets, IEEE transactions on visualization and computer graphics 20 (2014) 513–22. doi:10.1109/TVCG.2014.20.
  - [17] E. D. Ragan, R. Kopper, P. Schuchardt, D. A. Bowman, Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task, IEEE Transactions on Visualization and Computer Graphics 19 (2013) 886–896. doi:10.1109/TVCG.2012.163.
  - [18] J. J. Chen, H. Cai, A. P. Auchus, D. H. Laidlaw, Effects of stereo and screen size on the legibility of three-dimensional streamtube visualization, IEEE Transactions on Visualization and Computer Graphics 18 (2012) 2130–2139.
  - [19] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, S. Davidoff, J. S. Norris, G. Longo, Immersive and collaborative data visualization using virtual reality platforms, in: 2014 IEEE International Conference on Big Data (Big Data), 2014, pp. 609–614. doi:10.1109/BigData.2014.7004282.
  - [20] A. Bayyari, M. E. Tudoreanu, The impact of immersive virtual reality displays on the understanding of data visualization, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 368–371. URL: <https://doi.org/10.1145/1180495.1180570>. doi:10.1145/1180495.1180570.
  - [21] P. Millais, S. L. Jones, R. Kelly, Exploring data in virtual reality: Comparisons with 2d data visualizations, in: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 1–6. URL: <https://doi.org/10.1145/3170427.3188537>. doi:10.1145/3170427.3188537.
  - [22] A. Drogemuller, A. Cunningham, J. Walsh, M. Cordeil, W. Ross, B. Thomas, Evaluating navigation techniques for 3d graph visualizations in virtual reality, in: 2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA), 2018, pp. 1–10. doi:10.1109/BDVA.2018.8533895.
  - [23] E. Olshannikova, A. Ometov, Y. Koucheryavy, T. Olsson, Visualizing big data with augmented and virtual reality: challenges and research agenda, Journal of Big Data 2 (2015). doi:10.1186/s40537-015-0031-2.

- [24] W. Natephra, A. Motamedi, Live data visualization of iot sensors using augmented reality (ar) and bim, in: 36th International Symposium on Automation and Robotics in Construction (ISARC 2019), 2019.