

# Visual Analytics Tools for Polycode Stimuli Eye Gaze Tracking in Virtual Reality

Konstantin Ryabinin<sup>1,2</sup>, Konstantin Belousov<sup>1,2</sup> and Roman Chumakov<sup>2</sup>

<sup>1</sup>Saint Petersburg State University, University Embankment, 7/9, Saint Petersburg, 199034, Russia

<sup>2</sup>Perm State University, Bukireva Str., 15, Perm, 614068, Russia

## Abstract

This paper is devoted to extending the previously created unified pipeline for conducting eye-tracking-based experiments in a virtual reality environment. In the previous work, we proposed using SciVi semantic data mining platform, Unreal Engine and HTC Vive Pro Eye head-mounted display to study reading process in the immersive virtual reality. The currently proposed extension enables to handle so-called polycode stimuli: compound visual objects, which consist of individual parts carrying different semantics for the viewer. To segment polycode stimuli extracting areas of interest (areas, where the informant's eye gaze is being tracked) we adopt Creative Maps Studio vector graphics editor. To integrate Creative Maps Studio into the existing pipeline we created plugins for SciVi platform to load and handle the segmented stimuli, place them in the virtual reality scenes, collect corresponding eye gaze tracking data and perform visual analysis of the data collected. To analyze the eye gaze tracks, we utilize a circular graph that allows comprehensive visualization of hierarchical areas of interest (mapping them to color-coded graph nodes grouped into the hierarchy with a help of multilevel circular scale) and corresponding eye movements (mapped to the graph edges). We tested our pipeline on two different stimuli: the advertising poster and the painting "The Appearance of Christ Before the People" by A. Ivanov (1857).

## Keywords

Eye Gaze Tracking, Virtual Reality, Visual Analytics, Image Segmentation, Polycode Stimulus

## 1. Introduction

Precise tracking of eye movements is a mature methodology for studying the comprehension processes in different human activities, including reading, viewing of pictures, and even spatial navigation [1, 2, 3, 4]. Often, eye tracking is used in the interdisciplinary research involving specialists in physiology and different humanitarian sciences (like linguistics, sociology, marketing, urbanistics, etc.). The latest advances in computer vision and corresponding hardware drastically reduced the costs of eye tracking systems allowing them to be used by a wide range of research groups all over the world [5, 6].

Recently, several vendors introduced head-mounted displays (HMDs) with integrated infrared eye trackers, enabling to detect eye gaze within virtual reality (VR) scenes [7]. This, in turn,

---

*GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27–30, 2021, Nizhny Novgorod, Russia*

✉ kostya.ryabinin@gmail.com (K. Ryabinin); belousovki@gmail.com (K. Belousov); chumakoff.r.v@gmail.com (R. Chumakov)

🆔 0000-0002-8353-7641 (K. Ryabinin); 0000-0003-4447-1288 (K. Belousov); 0000-0002-6860-1932 (R. Chumakov)

© 2021 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)



brings eye-tracking-powered research to the new level, allowing to build and study virtual environments, which are impossible to recreate in the real world [8].

The data collected by the eye tracking device are to be analyzed using different metrics to discover the meaningful information about the informants' perception features. The most common metrics involve fixations (moments when eyes are stationary), saccades (moments when eyes rapidly move between viewing positions), and scanpaths (sequences of eye gaze fixations and interconnecting saccades in chronological order), which are detected in specific spatial zones called areas of interest (AOIs) [9, 10].

To work with the above metrics, visual analytics tools are demanded [11, 12]. But while there are quite a lot of software tools for conducting research with traditional stationary eye trackers, there is a lack of corresponding tools integrated with VR engines, which allow utilizing the eye trackers built in the HMDs [13]. In our previous work, we proposed a flexible pipeline to track and analyze the eye gaze in a VR environment, utilizing high-level semantic data mining and visual analytics platform SciVi [13]. The present work is aimed to build new stages upon this pipeline to allow handling of polycode stimuli in the eye-tracking-based research.

Polycode stimulus is a compound visual object presented to the informant during the eye-tracking-based experiment. This object consists of several subobjects representing different semiotic codes (entities of different semantics, for example, elements of different nature and/or meaning). Typical examples of polycode stimuli are modern advertisements, which contain images and short text blocks. More sophisticated cases are paintings, which typically can be split up into many areas representing different topics, which are combined by the artist into a solid work, reflecting his/her thoughts, emotions, and world outlook.

Study of the polycode stimuli comprehension is important both in fundamental and applied humanitarian sciences. Eye tracking metrics are a powerful way to obtain objective data for this study, but a problem is in stimuli segmentation to define particular AOIs. In the general case, such segmentation cannot be performed automatically due to the complexity of stimuli structures. To solve this problem, user-friendly high-level segmentation tools are needed, which allow experts to perform the segmentation manually or in an automated yet human-supervised manner.

In the present work, we propose reusing Creative Maps Studio vector editor [14] to provide Digital Humanities (DH) research experts with the ergonomic tools for segmenting multimodal stimuli into hierarchical AOIs, which can then be exported to the SciVi platform and used in the VR-based eye tracking pipeline. To test the approach proposed, we studied two use cases: the advertising poster and the painting "The Appearance of Christ Before the People" by A. Ivanov [15]. Both use cases include the segmentation phase (performed within Creative Maps Studio), the VR-based presentation phase (performed with help of Unreal Engine, with eye tracks being detected and recorded using SciVi), and the visual analytics phase (performed within SciVi).

## 2. Key Contributions

In this work, we propose an improvement for our previously created visual analytics pipeline leveraging eye tracking experiments in the VR environment [13]. The distinctive new features

proposed are the following:

1. The polycode stimuli segmentation tool based on Creative Maps Studio [14].
2. The corresponding plugins for the SciVi platform, which allow to automatically extract the hierarchical AOIs from the segmented polycode stimuli and visually analyze the corresponding scanpaths during the eye-tracking-based experiments.

The above features allow us to study the perception of polycode stimuli in VR.

### 3. Related Work

Eye tracking is a well-documented methodology to study the information perception of humans since it has been proven that eye movements reflect the comprehension process [1]. K. Rayner presented an elaborate review of the research related to the discovery of information processing features through eye gaze tracking [2], describing different information processing contexts and providing the most considerable metrics. K. Holmqvist et al. summarized the most popular and practically significant methods and measures of eye-tracking-based research in a comprehensive guide [4] that can leverage a quick start for novice scholars. One more concise yet informative review of eye tracking metrics is provided by Z. Sharafi et al. [10]. Later, nearly the same authors published a longer practical guide on eye tracking research [11]. This guide is claimed to be related to using eye tracking in software engineering, but the mentioned methods and means can be used far beyond this area. The above guides indicate scanpaths as basic structures to analyze for revealing the perception features. This is why scanpaths are the focus of our present work.

To handle the eye tracks detected, visual analytics tools are needed. T. Blascheck et al. collected the state-of-the-art visualization techniques used in eye tracking [12]. According to this review and some other reports of the same author, a handy technique to visualize scanpaths is a circular heat map [16, 17]. It is a graph-based representation, in which AOIs are drawn as nodes in a circle, and saccades are depicted as arcs between them. Since we already used a similar representation of linked data in our previous research [18], we decided to adopt this visualization concept to solve our eye tracking tasks.

The recent advances in microelectronics enabled the rapid evolution of mobile eye trackers, which can now be integrated with VR HMDs. N. Stein et al. carried out some benchmarks among the most popular VR HMDs with integrated eye trackers and stated that the characteristics of these devices are fair enough to conduct eye-tracking-based research [7]. This corresponds to the fact that in the last few years, many VR-based research projects emerged involving eye tracking. For example, D. Sonntag et al. designed an immersive traffic simulation to monitor the elderly pedestrians' spatial navigation skills [3]. A. Skulmowski et al. used VR to simulate the critical emotionally charged situation in which the informants encounter a moral choice and social judgments [8]. This kind of experiment would be nearly impossible in the real world, while it is safe and easy in a VR environment.

B. Farnsworth described the most popular software programs used by scholars for eye tracks analysis [6]. However, these programs are mostly designed for stationary eye tracking devices, and there is a lack of software that would enable seamless integration with VR scenes observed

with eye-tracker-powered HMDs. This is why the above-mentioned research projects involved ad-hoc software solutions to manage VR scenes and obtain needed eye tracking data instead of using some unified solutions. To support the spreading of eye-tracking-based experiments in VR environments, this gap has to be bridged. While there are some attempts to create a unified software for eye tracking in VR [19], many problems are not yet solved.

In our research, we use so-called polycode stimuli to conduct eye-tracking-based experiments. A polycode stimulus is a visual object that consists of separable parts having different semantics for the informant. These semantics are denoted as semiotic codes. The codes have their representation types called the modalities. For example, there are image-based and text-based codes. In our work we focus on unimodal (single-typed codes) and bimodal (two types of codes) stimuli.

As an example of unimodal stimulus, we consider a painting “The Appearance of Christ Before the People” by A. Ivanov [15]. One can extract from this painting such codes as different social roles, clothing, and gestures of depicted persons, spatial separation of persons’ groups, etc. The codes are used by the painting’s author to represent the dynamics of the painting’s plot, so the viewer (in our case, the informant) has to decode these codes to understand the story. The ability to decode the semiotic codes is tied to the cultural context. For example, one can expect completely different interpretations of the story told in “The Appearance of Christ Before the People” of informants familiar and unfamiliar with Christian culture and European painting (the comparison of the eye gaze tracks of such informants’ sample is beyond of this work’s scope but appears to be an interesting topic for the future Digital Humanities research). Polycode nature of paintings is an interesting material to study processes of perception and symbols’ decoding. Previously, “The Appearance of Christ Before the People” painting was used to study certain aspects of semiotic codes decoding [20]. Currently, VR environment and eye gaze tracking capabilities can bring this study to a qualitatively new level, covering the entire comprehension process, from perception up to interpretation.

Bimodal stimulus in our research is represented by so-called creolized text: a visual object containing both picture and text. We took an advertising poster that was previously studied in [21] to see whether the VR environment introduces new features to the perception process.

In our research, we rely on the results obtained in [22, 23, 24], but the distinctive feature of the present work is a focus on the VR. Although VR provides promising possibilities for Digital Humanities research, there are only a few works related to the comprehension processes in the VR environment [25]. The possible reason for this is the absence of high-level software tools to help the researchers to set up and conduct the related experiments. This is why we aim to create a high-level user-friendly pipeline integrating SciVi semantic data mining platform, Creative Maps Studio vector drawing editor, and Unreal Engine graphics rendering tools.

## 4. Background

The reported work is the next step in the development of the unified flexible pipeline for studying information comprehension processes in a VR environment. Previously, we proposed an extensible pipeline for composing virtual scenes, collecting the eye gaze tracks of the informants in these scenes, and analyzing the collected data through the visualization of the

scanpaths [13].

The core of the proposed pipeline is the ontology-driven client-server semantic data mining platform SciVi. The SciVi server is written in Python using Flask, the SciVi thin client is written in JavaScript using HTML5 and CSS3. This platform provides a high-level graphical user interface to declare particular data mining pipelines in a form of data flow diagrams (DFDs), which consists of nodes representing the data handling operators and links representing data flow. Each operator has its own ontological description stored in the SciVi knowledge base (using ONT format to represent ontologies and ONTOLIS editor [26] to create/modify them). This description includes the operator's typed inputs, outputs, and settings, as well as a link to the operator's implementation. Based on this description, SciVi automatically generates the graphical user interface for tuning and chaining these operators, as well as internal interoperability mechanisms to execute these operators on the available computing nodes in the network and interchange data between them. This approach ensures the easy extensibility and high configurability of the SciVi platform, allowing us to adapt SciVi for solving different data mining tasks in different application domains.

To adapt SciVi to the eye tracking tasks we created a corresponding set of operators described in [13]. These operators leverage VR scene tuning, eye gaze tracks collecting, and scanpaths visualization. To render the VR scene, we use Unreal Engine, for which we created a small WebSocket-based communication module (written in C++) to enable interoperability with SciVi at runtime. To present the VR scene to the informant and track the informant's eye gaze we use the HTC Vive Pro Eye HMD, which is one of the most popular state-of-the-art VR headsets with an integrated precise infrared eye tracker. To visually analyze the scanpaths, we adopt the SciVi::CGraph module [18] to draw a circular graph of AOIs and interconnecting saccades.

The focus of our previous work was on the study of the reading process in VR, so we considered texts as stimuli in the eye gaze tracking experiments. The present work is devoted to the generalization of the proposed approach enabling to study the comprehension of polycode stimuli.

For the polycode stimuli segmentation, we use the web-based vector graphic editor Creative Maps Studio [14]. This application was originally developed for creating and editing naive maps. Its server side is written in Python and client side is written in JavaScript. React, FastAPI and PostgreSQL are used as the main software Creative Maps Studio is based on. The tools provided by the Creative Maps Studio are quite versatile and therefore can be used, inter alia, for the segmentation of images by outlining different areas and storing their verbal descriptions.

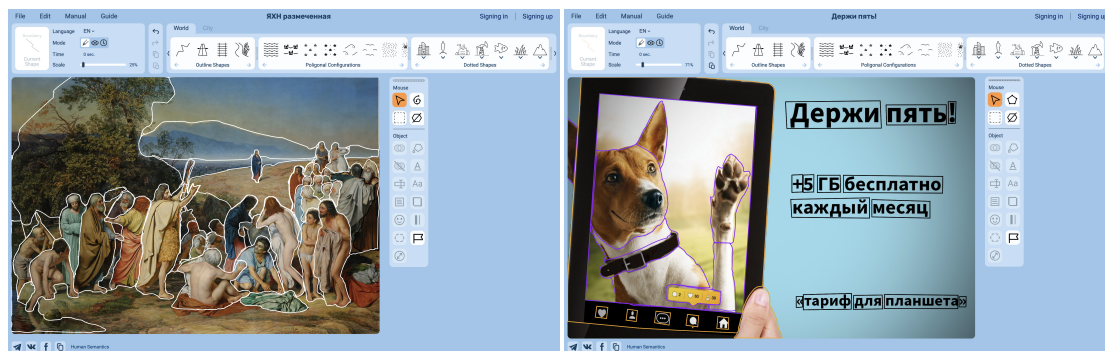
## 5. Polycode Stimuli Segmentation

As mentioned above, automatic segmentation of polycode stimuli by marking individual AOIs is nearly impossible, so the process has to be supervised by a human expert. In some cases, semi-automatic methods can be used (for example, to extract areas with text and images). But, for example, in the case of segmenting the paintings, there are almost no clues to build a universal segmentation algorithm that could perform automatic AOIs outline. So, either advanced machine learning algorithms or ergonomic manual segmentation tools are required to fulfill this task in a general case. In the present work, we start with the manual segmentation

of the polycode stimuli, yet take a very complex stimulus for testing.

Technically, the polycode stimulus is represented by a raster image. To enable its ergonomic manual segmentation, we extended the functionality of the Creative Maps Studio web application by adding the function of setting the raster background to the canvas (the area where the user draws a vector image). Having a stimulus image as a background, the expert can outline expected AOIs with the help of vector-based contours (possibly closed sequences of quadratic Bézier curves and line segments). Contour drawing is a built-in function of Creative Maps Studio. In addition to the graphical representation, the contours can also have verbal metadata: name, description, and associated emoticons. Verbal metadata helps to identify the AOIs in the further analysis of scanpaths. Some of the objects in the image can have a complex structure. An example of such an object is an image of a group of persons, where each person represents some independent semantics for the viewer but is united with the other persons in a group by some attribute. In this case, individual outlined regions can be joined into the named groups representing hierarchical AOIs.

Fig. 1 shows the segmentation of two images used as stimuli for eye-tracking-based experiments within this work. Left one represents a unimodal stimulus made of “The Appearance of Christ Before the People” painting. Right one represents a bimodal stimulus (creolized text) made of the advertising poster.



**Figure 1:** “The Appearance of Christ Before the People” painting by A. Ivanov (left) and the creolized text (right) segmented in the Creative Maps Studio.

## 6. Visual Analytics of Scanpaths

### 6.1. Design of Experiment

Once the segmentation of a stimulus is done, the actual experiment can be conducted to reveal the specifics of comprehending this stimulus by different informants. The general experiment setup is described in detail in [13].

Each VR-based experiment with eye gaze tracking involves three different software programs running on different computers interconnected via the network. The main program is the SciVi server that manages the entire experiment process. Being a Python-based software, it can run on almost any computer. In our case, it runs on the desktop computer in the laboratory. The



second one is the SciVi Web client that provides a control panel for the experiment director. It can run in any HTML5-capable browser. During the experiment, the director normally uses a notebook to be mobile in the laboratory. The third software program is a VR application based on the Unreal Engine. It runs on the desktop computer with NVidia Titan RTX GPU capable of rendering the VR scene for HTC Vive Pro Eye HMD.

First, the experiment director explains to the informant the anonymity of data being collected as well as the safety regulations related to the immersion in VR. If agrees, the informant signs the informed consent to participate in the experiment (having an option to decline and leave any time during the experiment).

The first step of the experiment is the eye tracker calibration leveraged by the standard procedure provided by the HTC Vive Pro Eye driver. After that, the experiment director sets up SciVi to show the test pattern to the informant and visually inspect the gaze detection data in order to check if calibration results are valid and if the informant feels comfortable (the SciVi setup is described in detail in Section 6.2). The test pattern consists of numbered circles of different sizes and colors the informant is asked to look at in ascending order of numbers.

Next, an actual stimulus is presented and the tracks of the informant's eye gaze are collected and stored (see Section 6.2 6.2). The rendering result of the stimulus image is presented in Fig. 2.



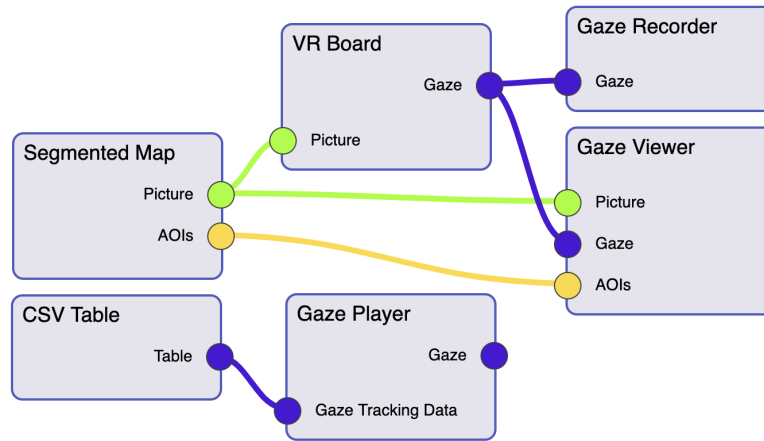
**Figure 2:** “The Appearance of Christ Before the People” painting by A. Ivanov presented in VR using the Unreal Engine and the SciVi platform.

At the moment, the last step of the experiment is the visualization and analysis of individual scanpaths (see Section 6.3). But in the future, we plan to further extend the data processing pipeline by adding the metrics considering eye gaze tracking results of multiple informants to discover common patterns and search for the comprehension features of different informants' groups.

## 6.2. Recording of Eye Gaze Tracking Data

The process of monitoring and recording of eye gaze tracking data for a particular stimulus is declared in SciVi by the DFD shown in Fig. 3.

This DFD is composed by the experiment director within the SciVi Web interface provided



**Figure 3:** SciVi DFD declaring the monitoring and recording of eye gaze tracking data.

by the SciVi thin client. The “Segmented Map” operator allows loading the stimulus image segmented in the Creative Maps Studio. This operator has two outputs: the raster picture corresponding to the stimulus and the array of AOIs encoded as named groups of named paths (arrays of 2D points). The picture (encoded using the Base64 standard) is transmitted to the “VR Board” operator that is responsible for communication with the VR application. This communication is based on the WebSocket protocol. The VR application renders the scene and shows the received image to the informant (see Fig. 2). The gaze of the informant’s eyes is detected by the built-in eye tracker of the HTC Vive Pro Eye HMD. The detection results are collected by the SRanipal SDK API that allows to get the scene object the informant is looking at, as well as 3D coordinates (in the scene space) of the gaze origin and case focal point. If the object is the plain displaying the stimulus image (see Fig. 2), the gaze focal point is converted from the scene space to the texture space of this plain in order to get the coordinates related to the stimulus image.

The gaze-related data are transmitted back to SciVi via the WebSocket using the following CSV-based encoding: timestamp, gazeU, gazeV, gazeOriginX, gazeOriginY, gazeOriginZ, gazeFocusX, gazeFocusY, gazeFocusZ. While gazeU and gazeV are used to perform a hit test with AOIs, other values are needed to detect saccades and fixations.

The “Gaze Viewer” is a visualization operator dedicated to monitoring the gaze tracks being collected. The rendering result of this operator is shown in Fig. 4. The blue circle represents the gaze focal point. The white contour indicates the AOI the informant is looking at, and the corresponding AOI name is displayed on the left (the names of groups the AOI belongs to are prefixed with the slash symbol in hierarchical order).

The “Gaze Recorder” operator allows recording gaze tracking data to a CSV file, while the “CSV Table” operator is dedicated to loading these data back to SciVi, and the “Gaze Player” operator allows playing them back. In Fig. 3 this operator is not connected with the others, but it can, for example, be linked to the “Gaze Viewer” instead of “VR Board”, so the recorded data can be viewed again. Using the described operators, the experiment director can easily set up SciVi for particular eye-tracking-based experiments. It is worth noting that AOIs can be defined



John the Baptist with a cross / cross

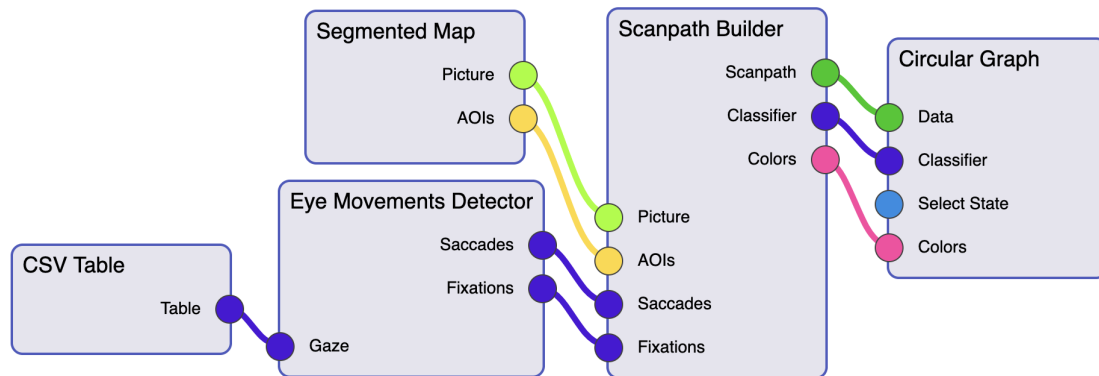


**Figure 4:** Eye tracking data monitoring in SciVi.

(or redefined) after the data are recorded because the recorded files contain just the raw eye gaze tracks over a particular stimulus.

### 6.3. Scanpath Detection and Visualization

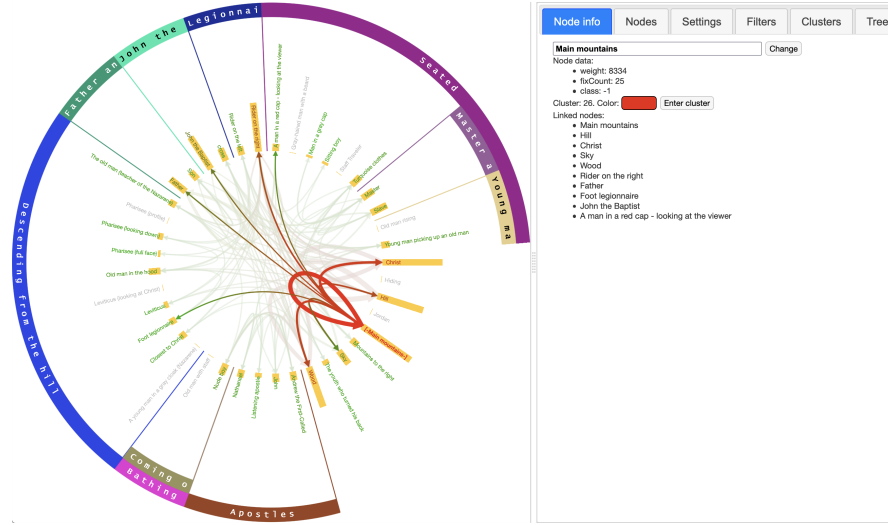
Based on the recorded eye gaze tracking data, the scanpaths are calculated and visualized using our modified version of a radial transition graph [13]. The corresponding SciVi DFD is shown in Fig. 5.



**Figure 5:** SciVi DFD declaring scanpath building and visualization.

The “Eye Movements Detector” operator determines whether the gaze tracking samples belong to saccades or to fixations. This operator is based on the algorithm proposed by J. Llanes-Jurado et al. [27] (we have ported the reference implementation of this algorithm to JavaScript to be able to run it on the SciVi client side). The “Scanpath Builder” operator assembles the scanpath by joining fixations and interchanging saccades, producing the data suitable for displaying with

SciVi::CGraph visualization module represented by the “Circular Graph” operator. The example of the rendering result is shown in Fig. 6.



**Figure 6:** SciVi circular graph representing saccades and fixations within AOIs of “The Appearance of Christ Before the People” painting.

The AOIs are displayed as graph nodes placed in a circle. The total fixation duration in each AOI is shown as histogram box behind the name of this AOI, and the number of fixations is mapped to its color (for this, the array of colors is produced by the “Scanpath Builder” operator as one of the outputs). The saccades are represented by graph edges rendered as arrows with the number of saccades mapped to the arrows’ thickness. The AOIs hierarchy is displayed with a help of the multilevel ring scale drawn around the graph nodes [18].

Compared to our previous work [13], AOIs hierarchy representation is a new distinctive feature of scanpaths visual analytics proposed. Together with the filtering capabilities of SciVi::CGraph adopted to the eye gaze tracking tasks in [13], this new feature allows efficient analysis of the scanpaths on the complex polycode stimuli.

## 7. Conclusion

In the present work, we extended our previously proposed pipeline for visual analytics of eye gaze tracks collected in a VR environment. The extensions developed allow us to handle polycode stimuli, enlarging the variety of experiments we can conduct using HTC Vive Pro Eye HMD, Unreal Engine, Creative Maps Studio vector graphics editor, and SciVi semantic data mining platform. The tools created were tested on two different polycode stimuli: unimodal one (a painting containing a lot of different persons and environmental objects) and bimodal one (an advertising poster containing picture and text). The preliminary results of eye gaze tracking analysis were obtained for pipeline testing purposes, while the experiments involving real informants will be a part of future work. After the tests were conducted, the viability of our pipeline has been confirmed, and we can conclude that this pipeline possesses great potential for Digital Humanities research.

Next, we intend to implement more complex eye gaze tracking metrics as individual SciVi platform plugins to discover the perception features of informants in a VR environment. Also, we plan to study possible ways to automate multimodal stimuli segmentation process with a help of machine learning techniques.

## Acknowledgments

This study is supported by the research grant No. ID75288744 from Saint Petersburg State University.

## References

- [1] K. Rayner, K. H. Chace, T. J. Slattery, J. Ashby, Eye Movements as Reflections of Comprehension Processes in Reading, *Scientific Studies of Reading* 10 (2006) 241–255. doi:10.1207/s1532799xssr1003\_3.
- [2] K. Rayner, Eye Movements in Reading and Information Processing: 20 Years of Research, *Psychological Bulletin* 124 (1998) 372–422. doi:10.1037/0033-2909.124.3.372.
- [3] D. Sonntag, J. Orlosky, M. Weber, Y. Gu, S. Sosnovsky, T. Toyama, E. N. Toosi, Cognitive Monitoring via Eye Tracking in Virtual Reality Pedestrian Environments, in: *Proceedings of the 4th International Symposium on Pervasive Displays*, 2015, pp. 269–270. doi:10.1145/2757710.2776816.
- [4] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, J. van de Weijer, *Eye Tracking: A Comprehensive Guide To Methods And Measures*, OUP Oxford, 2011.
- [5] Tobii, 2021. URL: <https://www.tobii.com>, last accessed 8 Jul 2021.
- [6] B. Farnsworth, 10 Free Eye Tracking Software Programs [Pros and Cons], 2021. URL: <https://imotions.com/blog/free-eye-tracking-software/>, last accessed 8 Jul 2021.
- [7] N. Stein, D. C. Niehorster, T. Watson, F. Steinicke, K. Rifai, S. Wahl, M. Lappe, A Comparison of Eye Tracking Latencies Among Several Commercial Head-Mounted Displays, *i-Perception* 12 (2021) 1–16. doi:10.1177/2041669520983338.
- [8] A. Skulmowski, A. Bunge, K. Kaspar, G. Pipa, Forced-Choice Decision-Making in Modified Trolley Dilemma Situations: a Virtual Reality and Eye Tracking Study, *Frontiers in Behavioral Neuroscience* 8 (2014). doi:10.3389/fnbeh.2014.00426.
- [9] A. Poole, L. J. Ball, Eye Tracking in HCI and Usability Research, *Encyclopedia of Human Computer Interaction* (2006) 211–219. doi:10.4018/978-1-59140-562-7.ch034.
- [10] Z. Sharafi, T. Shaffer, B. Sharif, Y.-G. Guéhéneuc, Eye-Tracking Metrics in Software Engineering, in: *2015 Asia-Pacific Software Engineering Conference (APSEC)*, 2015, pp. 96–103. doi:10.1109/APSEC.2015.53.
- [11] Z. Sharafi, B. Sharif, Y.-G. Guéhéneuc, A. Begel, R. Bednarik, M. Crosby, A Practical Guide on Conducting Eye Tracking Studies in Software Engineering, *Empirical Software Engineering* 25 (2020) 3128–3174. doi:10.1007/s10664-020-09829-4.
- [12] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, T. Ertl, State-of-the-Art of Visualization for Eye Tracking Data, in: *EuroVis - STARs*, 2014, pp. 63–82. doi:10.2312/eurovisstar.20141173.

- [13] K. Ryabinin, K. Belousov, Visual Analytics of Gaze Tracks in Virtual Reality Environment, *Scientific Visualization* 13 (2021) 50–66. doi:10.26583/sv.13.2.04.
- [14] R. Chumakov, K. Ryabinin, K. Belousov, J. Duan, Creative Map Studio: A Platform for Visual Analytics of Mental Maps, *Scientific Visualization* 13 (2021) 79–93. doi:10.26583/sv.13.2.06.
- [15] A. A. Ivanov, The Appearance of Christ Before the People, 1857. URL: [https://en.wikipedia.org/wiki/The\\_Appearance\\_of\\_Christ\\_Before\\_the\\_People#/media/File:Александр\\_Андреевич\\_Иванов\\_-\\_Явление\\_Христа\\_народу\\_\(Явление\\_Мессии\)\\_-\\_Google\\_Art\\_Project.jpg](https://en.wikipedia.org/wiki/The_Appearance_of_Christ_Before_the_People#/media/File:Александр_Андреевич_Иванов_-_Явление_Христа_народу_(Явление_Мессии)_-_Google_Art_Project.jpg).
- [16] T. Blascheck, M. Raschke, T. Ertl, Circular Heat Map Transition Diagram, in: *Proceedings of the 2013 Conference on Eye Tracking South Africa*, 2013, pp. 58–61. doi:10.1145/2509315.2509326.
- [17] T. Blascheck, B. Sharif, Visually Analyzing Eye Movements on Natural Language Texts and Source Code Snippets, in: *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*, 2019, pp. 1–9. doi:10.1145/3314111.3319917.
- [18] K. Ryabinin, K. Belousov, S. Chuprina, Novel Circular Graph Capabilities for Comprehensive Visual Analytics of Interconnected Data in Digital Humanities, *Scientific Visualization* 12 (2020) 56–70. doi:10.26583/sv.12.4.06.
- [19] J. C. E. Iacobi, Software for Analyzing User Experiences in Virtual Reality using Eye Tracking, 2018. URL: <https://kth.diva-portal.org/smash/get/diva2:1231972/FULLTEXT01.pdf>, last accessed 14 Jul 2021.
- [20] K. I. Belousov, E. A. Podtikhova, Reflections on Conceptual Composition of Text, *Voprosy Kognitivnoy Lingvistiki* 1 (2014) 62–74.
- [21] A. Konovalova, T. Petrova, Lexical Ambiguity in Slogans: Does It Make a Polycode Text Easier to Recognize?, in: *Proceedings of the 4th International Conference on Neurobiology of Speech and Language*, 2020, pp. 54–55.
- [22] T. Petrova, E. Riekhakaynen, Eye-tracking study of processing Russian polycode texts, in: *The 20th European Conference on Eye Movements, ECEM 2019*, 2019.
- [23] T. Petrova, N. Kalugina, Eye-Tracking Study of Reading Polycode Texts: Evidence From Russian, *Perception* 48 (2019).
- [24] T. E. Petrova, E. I. Riekhakaynen, Processing of Verbal and Non-verbal Patterns: An Eye-Tracking Study of Russian, in: *Third International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing*, volume 797, 2019, pp. 269–276. doi:10.1007/978-981-13-1165-9\_24.
- [25] J. Mirault, A. Guerre-Genton, S. Dufau, J. Grainger, Using Virtual Reality to Study Reading: An Eye-Tracking Investigation of Transposed-Word Effects, *Methods in Psychology* 3 (2020). doi:10.1016/j.metip.2020.100029.
- [26] S. Chuprina, O. Nasraoui, Using Ontology-based Adaptable Scientific Visualization and Cognitive Graphics Tools to Transform Traditional Information Systems into Intelligent Systems, *Scientific Visualization* 8 (2016) 23–44.
- [27] J. Llanes-Jurado, J. Marín-Morales, J. Guixeres, M. Alcañiz, Development and Calibration of an Eye-Tracking Fixation Identification Algorithm for Immersive Virtual Reality, *Sensors* 20 (2020). doi:10.3390/s20174956.