Novel Immersive Media Captioning Techniques and Related Evaluation System

Dylan Bruss
dbruss@clemson.edu
Clemson University
Clemson, South Carolina, USA

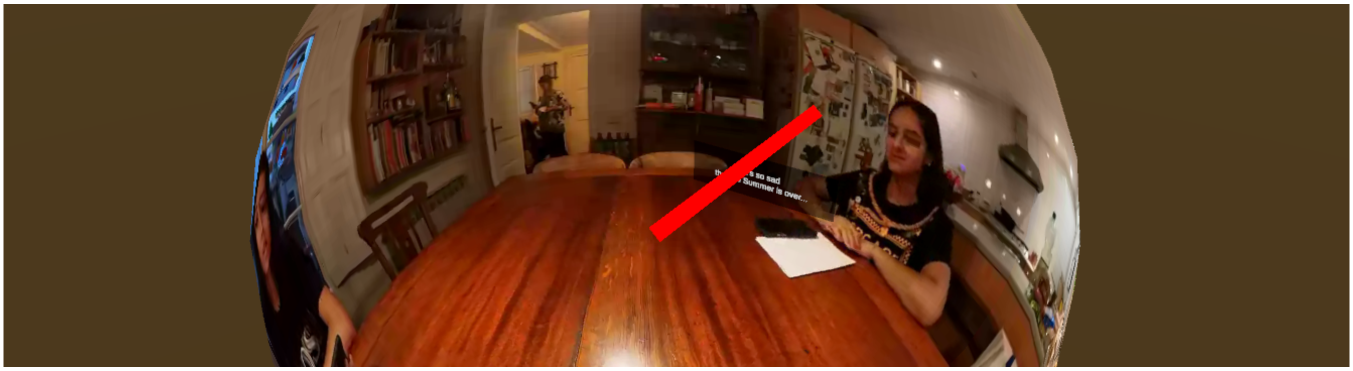


Figure 1: Third-person view of the experiment. The red line represents the user's gaze vector in 3D space.

ABSTRACT

This project presents an implementation of several novel techniques for captioning immersive media in a virtual reality environment, as well as several techniques for evaluating both caption invasiveness and reading comprehension using eye-tracking technology.

KEYWORDS

gaze detection, eye tracking, accessibility, virtual reality, subtitles

ACM Reference Format:
Dylan Bruss. 2023. Novel Immersive Media Captioning Techniques and Related Evaluation System. In *Proceedings of (Eye Tracking II).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

While commonly accepted standards for closed captioning have been available for years, the advent of virtual reality and new immersive storytelling techniques presents unique challenges for subtitling content not present in traditional mediums. While several techniques have been created and evaluated for this purpose [1, 3–5], the advent of a new medium provides an opportunity to evaluate new and different techniques to improve readability, comprehension, and ease of use. This paper implements several such techniques, as well as standardized metrics to evaluate them using eye-tracking technology. Such evaluation methods are meant to be used in addition to pre- and post-experience questionnaires to help evaluate both the utility and comfort of a particular method.

Eye Tracking II, Spring 2023, Clemson, SC
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$0.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 CHALLENGES

Immersive storytelling mediums present unique challenges for captioning that do not exist in traditional media. This section outlines some of the major challenges and constraints in the design of an immersive captioning system.

2.1 Freedom of view

In an immersive environment, users can view anywhere within a 360° sphere. This means that, unlike in traditional mediums, the user might not always be looking in the direction in which action occurs. Previously, Virtual Reality applications have used techniques like spatial audio to direct a user's vision, using the difference in attenuation between a user's ears to portray the location that a sound is coming from. This will not work for deaf users, however, who would be the primary target for accessibility features such as subtitles. As a result, subtitles must always appear within a user's view or have a mechanism for alerting the user to its location.

2.2 Caption Placement

In traditional media, captions are usually placed toward the bottom of the screen. However, as immersive environments do not have the limitation of a physical frame for content, a new placement technique must be used. While captions could be attached to the user's physical view, the transition to immersive media provides a good opportunity to explore other methods that may make the content more understandable and usable.

2.3 Motion Sickness

Motion sickness is a particular issue with many virtual reality systems. Motion sickness occurs when a user either appears to be moving in the virtual world without experiencing it in real life or, conversely, when the user moves without the movement being properly reflected in the virtual environment. Many factors play into motion sickness, but the major challenge presented for this project is the unavailability of any captioning technique that involves extreme movement (e.g., panning the entire environment so that the user is looking towards the action at all times).

3 RELATED WORK

Several other works have investigated the area of VR subtitling. Some examples will be discussed in this section.

In their paper, Brown et al. describe four methods for subtitle behavior in an immersive 3D environment in a manner meant to imitate traditional media to an extent. These methods are referred to as "120-degree", "Lag-Follow," "Static-Follow," and "Appear." "120 degree" behavior means that subtitles are displayed in triplicate towards the bottom of the environment, each at 120° angles from each other. "Lag-Follow" behavior causes the subtitles to slide toward the user's view after a certain delay. "Static-Follow" behavior means that the subtitle is directly attached to the user's view and directly reflects user movement at all times. Finally, the "Appear" behavior causes subtitles to stay in place for the duration of any given line but snap to the user's view as soon as a new line is ready to be viewed. This paper does not evaluate these methods, merely presenting them as implemented systems [1].

Orero et al. describe several methods for alleviating different issues with subtitling in an immersive context, as well as conceptually evaluating these techniques. They introduce several solutions for the issue of directing the user to the speaker in a given situation, including showing an arrow near the subtitle, in the center of the user's view, and showing a "radar" view to show the user where to look. They also discuss the techniques mentioned in Brown et al. and various combinations therein [4].

Hughes et al. describe additional, more disruptive methods for subtitle placement, as well as directing the user to the current AOI through their subtitling system called "ImAc." Subtitles can either be attached to the user's view or to the virtual world near the speaker. Users can then be directed to the speaker with a secondary subtitle with an arrow toward the speaker and the main subtitle. They also explore the arrow and radar methods mentioned previously. They also mention a mode called "auto-positioning," in which the user's field of view is automatically redirected towards the area of interest [3].

Finally, Rothe et al. attempt to evaluate some of the proposed methods through a user study. In addition, they present heatmaps from head-tracking data (notably not eye-tracking data). They note that users like subtitles attached to the headset due to its familiarity, freedom, and ease-of-use, but find it difficult to assign the text to the speaker. Conversely, dynamic subtitles, placed near each speaker, are preferred due to the ease of assigning the speaker, the fact that speakers and subtitles can be seen simultaneously, and the more natural feeling consumption. However, they dislike being forced to look at the speaker and also have a hard time figuring out where the next subtitle will appear.

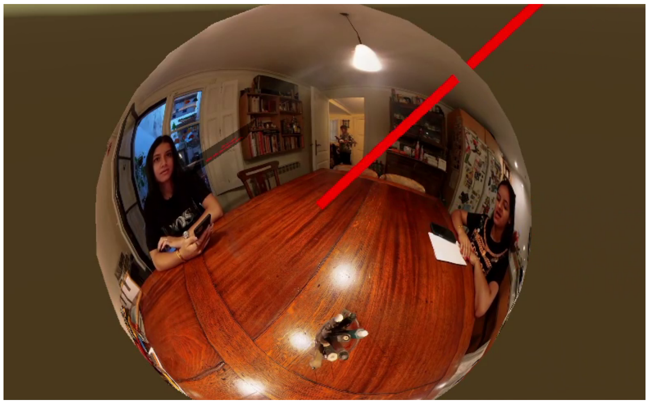


Figure 2: Videos are decompressed to a texture and applied to a sphere with inverted normals and backface culling. This setup allows for better third-person viewing for researchers outside the head-mounted display with a minimal performance impact. The red line is only rendered in third-person views and represents the user's gaze vector.

4 RESULTS

All tools were developed using the Vive Pro Eye virtual reality headset with an integrated 120Hz Tobii eye tracker. The Unity Engine was used as a development platform, and interaction with the headset was done using the OpenXR SDK.

Video is played back as a texture of an inverted sphere (See Figure 2) to allow for easier visualization for users outside of the headset. Subtitles are parsed from text files in the SRT subtitling format, with limited extensions for positioning and speaker identification.

Listed below in sections 4.1 and 4.2 are techniques for solving challenges with subtitle placement and speaker identification. Section 4.3 describes implemented evaluation techniques involving eye tracking.

4.1 Implementation of Existing Techniques

Several techniques used in previous papers have been implemented in this project to allow for evaluation with eye-tracking technology. For subtitle placement techniques in this and the following section, they can be toggled between by the researcher through the Unity Editor interface.

4.1.1 120-Degree subtitle behavior. This method, described by Brown et al. [1], is implemented by changing the location of a single subtitle depending on HMD rotation. First, the HMD rotation is flattened to the XY plane and is converted to an angle value in radians. This angle is then rounded to the closest 120° interval and applied to a user-defined offset vector to get the position of the subtitle. The rotation of the subtitle is defined as being perpendicular to the resulting vector, given an up direction closest to the global +Z axis.

4.1.2 Static-Follow. This technique, described by Brown et al. [1], is implemented by placing the subtitle at a user-defined offset vector from the headset position, rotated by the headset's 3D rotation. The rotation is defined similarly to the 120° method.

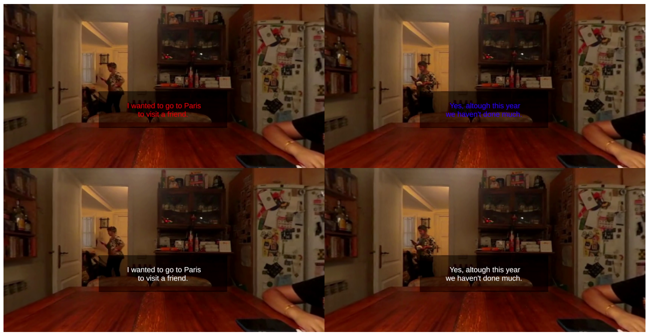


Figure 3: Color can be applied to determine which character is speaking (Top). This makes conversations easier to follow than with un-highlighted text (Bottom).

4.1.3 Lag-Follow. This technique, described by Brown et al. [1], is implemented similarly to the static follow technique. However, spherical interpolation ensures the subtitle location "lags behind" the position described in the static-follow technique. In each frame, the subtitle moves a fraction of the shortest arc distance to the vector given by the static-follow algorithm on a sphere. Spherical interpolation is used instead of linear interpolation to maintain the subtitle's distance from the user and prevent it from moving closer to the user as the user turns their head.

4.1.4 Appear. This technique is also described by Brown et al. [1]. It is implemented similarly to Static-Follow, except that the position is only set during the first frame that a subtitle is shown. This results in the subtitle "sticking" in the environment, allowing users to look away from it if desired and providing less distraction.

4.1.5 Dynamic Subtitles. This technique, described by Rothe et al. [5], is implemented using an extension of the SRT subtitling format. For each subtitle, an X and Y position is defined in the subtitle file. These are applied to the X and Y components of an Euler rotation, which is applied to a user-defined vector along the +Y axis to calculate the final location of the subtitle.

The result is subtitles that are placed close to the character that is currently speaking in a scene, helping users identify who is currently speaking and understand a scene better overall. However, this method can confuse users, as it does not direct them to the next speaker in the conversation.

4.1.6 Speaker-Based Text Highlighting. This technique is implemented using speaker annotations in the SRT subtitling format. Each speaker is given a zero-indexed id. The user can then define colors for each speaker. These colors will be applied to the subtitle text as that character speaks. This helps non-colorblind users more easily determine which user is speaking (see Figure 3). If no speaker identification is given in the SRT subtitles, a default color (typically white) is used. This technique can be applied to any subtitle placement method above or below.

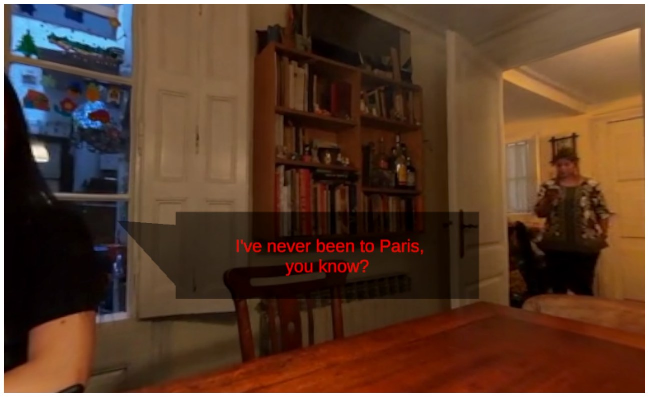


Figure 4: The Speech-Bubble method uses triangular "tails" to point a user towards the speaker.

4.2 Implementation of Novel Techniques

The following new techniques for virtual reality subtitling have been implemented specifically for this project in addition to the existing techniques above.

4.2.1 Moving-Bubble. This method is an improvement to the dynamic subtitles method above. Similar to that method, the moving-bubble approach aims to indicate the current speaker by placing the subtitle near that character. Unlike dynamic subtitles, however, subtitles are smoothly animated between positions using spherical interpolation. The animation should help direct users naturally to the next speaker without breaking immersion, mitigating the most egregious shortfall of the dynamic subtitles method.

4.2.2 Speech-Bubbles. This technique adds a triangular "tail" to the subtitle box that points to the speaker (see Figure 4). Placement of the box is done via the "Lag-Follow" method above. The tip of the tail is placed at a position $\frac{1}{10}$ of the distance to the position of the speaker in the scene (as defined by the Dynamic Subtitles method above). If this distance is less than two times the width of the subtitle, the distance is set to be two times the width of the subtitle. The base of the tail is placed at the center of the edge of the subtitle box closest to the speaker.

This allows the user to always know which character is talking in a scene without searching the scene for a subtitle. In addition, as opposed to previous methods, this uses a familiar paradigm to most users, as it is used in many mediums, from graphic novels to popular comic strips.

4.2.3 Bionic Reading. This technique, inspired by Fuhrmann and Casutt [2], renders the first half of each word in a body of text using a bold typeface. Rendering text in this manner should allow for easier and more effective reading by guiding the user's eyes from one word to the next (see Figure 5). For this implementation, words with an odd number of letters will round down the number of letters bolded. This can be applied alongside any of the subtitle placement methods above.

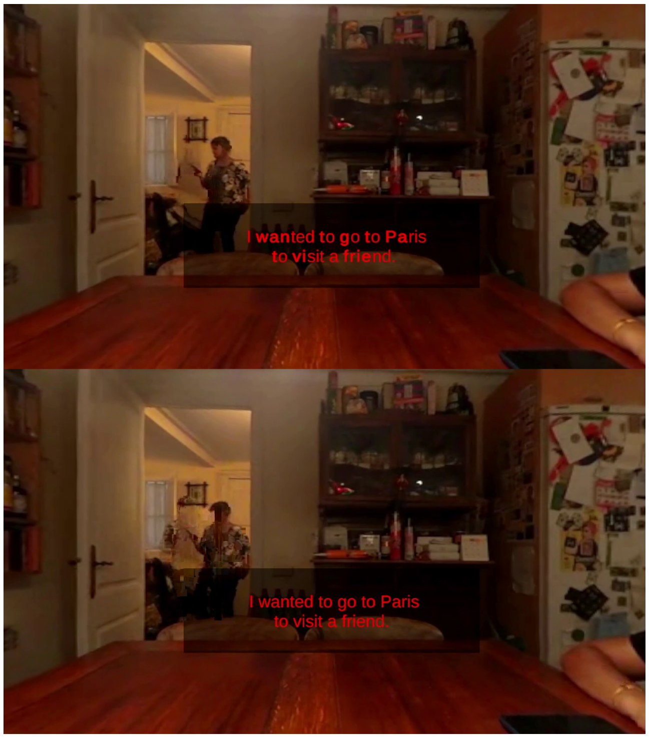
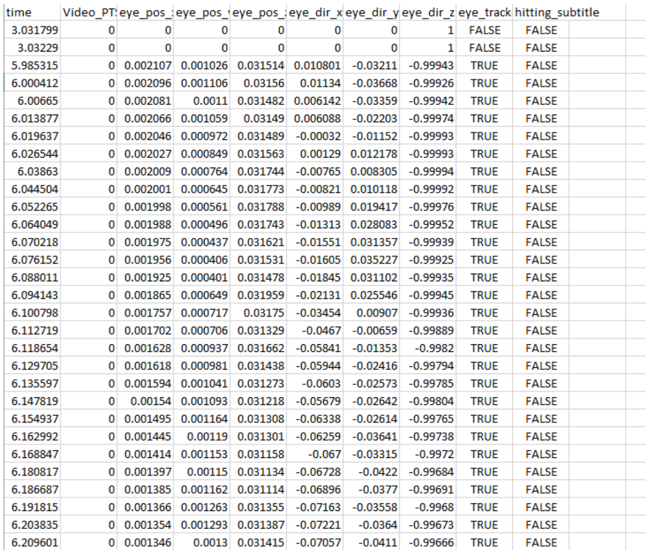


Figure 5: Bionic Reading methodologies (top) should help guide the user's gaze for easier reading than typical text (bottom). Notice how the first few letters of each word in the top subtitle are displayed using a bold typeface.



time	Video_PT	eye_pos_	eye_pos_	eye_pos_	eye_dir_x	eye_dir_y	eye_dir_z	eye_track	hitting_subtitle
3.031799	0	0	0	0	0	0	1	FALSE	FALSE
3.03229	0	0	0	0	0	0	1	FALSE	FALSE
5.985315	0	0.002107	0.001026	0.031514	0.010801	-0.03211	-0.99943	TRUE	FALSE
6.000412	0	0.002096	0.001106	0.03156	0.01134	-0.03668	-0.99926	TRUE	FALSE
6.00665	0	0.002081	0.0011	0.031482	0.006142	-0.03359	-0.99942	TRUE	FALSE
6.013877	0	0.002066	0.001059	0.03149	0.006088	-0.02203	-0.99974	TRUE	FALSE
6.019637	0	0.002046	0.000972	0.031489	-0.00032	-0.01152	-0.99993	TRUE	FALSE
6.026544	0	0.002027	0.000849	0.031563	0.00129	0.012178	-0.99993	TRUE	FALSE
6.03863	0	0.002009	0.000764	0.031744	-0.00765	0.008305	-0.99994	TRUE	FALSE
6.044504	0	0.002001	0.000645	0.031773	-0.00821	0.010118	-0.99992	TRUE	FALSE
6.052265	0	0.001998	0.000561	0.031788	-0.00989	0.019417	-0.99976	TRUE	FALSE
6.064049	0	0.001988	0.000496	0.031743	-0.01313	0.028083	-0.99952	TRUE	FALSE
6.070218	0	0.001975	0.000437	0.031621	-0.01551	0.031357	-0.99939	TRUE	FALSE
6.076152	0	0.001956	0.000406	0.031531	-0.01605	0.035227	-0.99925	TRUE	FALSE
6.088011	0	0.001925	0.000401	0.031478	-0.01845	0.031102	-0.99935	TRUE	FALSE
6.094143	0	0.001865	0.000649	0.031959	-0.02131	0.025546	-0.99945	TRUE	FALSE
6.100798	0	0.001757	0.000717	0.03175	-0.03454	0.00907	-0.99936	TRUE	FALSE
6.112719	0	0.001702	0.000706	0.031329	-0.0467	-0.00659	-0.99889	TRUE	FALSE
6.118654	0	0.001628	0.000937	0.031662	-0.05841	-0.01353	-0.9982	TRUE	FALSE
6.129705	0	0.001618	0.000981	0.031438	-0.05944	-0.02416	-0.99794	TRUE	FALSE
6.135597	0	0.001594	0.001041	0.031273	-0.0603	-0.02573	-0.99785	TRUE	FALSE
6.147819	0	0.00154	0.001093	0.031218	-0.05679	-0.02642	-0.99804	TRUE	FALSE
6.154937	0	0.001495	0.001164	0.031308	-0.06338	-0.02614	-0.99765	TRUE	FALSE
6.162992	0	0.001445	0.00119	0.031301	-0.06259	-0.03641	-0.99738	TRUE	FALSE
6.168847	0	0.001414	0.001153	0.031158	-0.067	-0.03315	-0.9972	TRUE	FALSE
6.180817	0	0.001397	0.00115	0.031134	-0.06728	-0.0422	-0.99684	TRUE	FALSE
6.186687	0	0.001385	0.001162	0.031114	-0.06896	-0.0377	-0.99691	TRUE	FALSE
6.191815	0	0.001366	0.001263	0.031355	-0.07163	-0.03558	-0.9968	TRUE	FALSE
6.203835	0	0.001354	0.001293	0.031387	-0.07221	-0.0364	-0.99673	TRUE	FALSE
6.209601	0	0.001346	0.0013	0.031415	-0.07057	-0.0411	-0.99666	TRUE	FALSE

Figure 6: An example of user data recorded with the presented evaluation system.

4.3 Implementation of Evaluation Techniques

Eye tracking data is recorded from the integrated Tobii eye tracker at 120Hz using the OpenXR Eye Gaze Interaction API. Monocular eye tracking data is provided by the OpenXR API as a single vector, specified by a vector position and quaternion rotation in 3D space. The OpenXR specification does not support binocular eye data, so it could not be recorded. Data is recorded at the end of a user session to a comma-separated value (CSV) file (see Figure 6).

The following data is recorded as a part of the CSV file.

4.3.1 Program Timestamp. The program timestamp is a 64-bit floating point value representing the total time, in seconds, between the start of the program and the recorded sample. It can be used to reconstruct a real-time playback of the user's gaze data post-experiment

4.3.2 Eye Tracking State. This is a boolean value that is true when the eye tracking sensor is properly detecting the user's gaze. A value of false usually occurs if the user's eye is closed (e.g., the user blinks).

4.3.3 Eye Gaze Vector. The position and direction of a user's gaze are recorded using two vectors with X, Y, and Z components. The gaze direction is recorded as a unit vector. External scripts can process this to generate eye gaze heat maps, detect fixations and saccades, and render the user's eye movements to animation in 3D space.

4.3.4 Subtitle ray intersection. During a session, a ray cast is done along the user's view to determine if they are looking at the subtitle or the environment around it. The ray cast is recorded to the CSV file as a single boolean, which is true if the user looks at the subtitle and false otherwise. External scripts can use this to generate statistics to determine how frequently the user looks at the subtitle. A high frequency in this category may imply that the subtitle is distracting or hard to read. Conversely, a low frequency can imply a subtitle that is

4.3.5 Video timestamp. This is the presentation time, in seconds, for the currently displayed frame of the immersive video. It can be cross-referenced with the SRT subtitle file to determine which subtitle appeared to the user during the sample. This may differ from the program timestamp, as there is a delay between the program startup and video playback. This can also be helpful in cases where decoding or other latency issues cause delays in playback.

5 FUTURE WORK

Preliminary observations imply that the eye-tracking data is precise enough to determine which word a user is looking at in a subtitle, provided that bounding boxes for each word can be generated. This would be useful information to help determine additional metrics such as reading speed, as well as cases where the user reads a subtitle more than once due to poor comprehension.

6 CONCLUSION

This project has implemented several techniques, both novel and pre-existing, for captioning immersive content, as well as a framework for eye-tracking-based evaluation of these techniques.

ACKNOWLEDGMENTS

Dr. Duchowski for his instruction and help on this project.

REFERENCES

[1] Andy Brown, Jayson Turner, Jake Patterson, Anastasia Schmitz, Mike Armstrong, and Maxine Glancy. 2017. Subtitles in 360-Degree Video. In *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video* (Hilversum, The Netherlands) *(TVX '17 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 3–8. https://doi.org/10.1145/3084289.3089915

[2] Birgit Fuhrmann and Renato Casutt. 2022. UX Reading : digitale Nutzungsinformationen fokussierter, bewusster und nachhaltiger lesen. https://digitalcollection.zhaw.ch/handle/11475/26741 Tekom-Frühjahrstagung 2022, Potsdam, Deutschland, 6.-7. April 2022.

[3] Chris Hughes, Mario Montagud Climent, and Peter tho Pesch. 2019. Disruptive Approaches for Subtitling in Immersive Environments. In *Proceedings of the 2019 ACM International Conference on Interactive Experiences for TV and Online Video* (Salford (Manchester), United Kingdom) *(TVX '19)*. Association for Computing Machinery, New York, NY, USA, 216–229. https://doi.org/10.1145/3317697.3325123

[4] Pilar Orero, Marta Brescia-Zapata, and Chris Hughes. 2021. Evaluating Subtitle Readability in Media Immersive Environments. In *9th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion* (Online, Portugal) *(DSAI 2020)*. Association for Computing Machinery, New York, NY, USA, 51–54. https://doi.org/10.1145/3439231.3440602

[5] Sylvia Rothe, Kim Tran, and Heinrich Hußmann. 2018. Dynamic Subtitles in Cinematic Virtual Reality. In *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video* (SEOUL, Republic of Korea) *(TVX '18)*. Association for Computing Machinery, New York, NY, USA, 209–214. https://doi.org/10.1145/3210825.3213556