Deliverable

Project Acronym:	**IMAC**
Grant Agreement number:	761974
Project Title:	*Immersive Accessibility*

D4.1-Subtitle Production Tools

Revision: 2.0

Authors: Peter tho Pesch (IRT)

Delivery date: M25 (02-10-19)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 761974

Dissemination Level		
P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

Abstract:

This document provides an overview of developed tools for subtitle production for immersive content. It accompanies the software products and describe their main features and their availability.

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	19-07-2018	Peter tho Pesch	IRT	Initial draft
0.2	01-08-2018	Kimiasadat Mirehbar	ANGLA	Anglatecnic input
0.3	01-08-2018	Peter tho Pesch	IRT	Merges IRT and Anglatecnic input
0.4	02-08-2018	Peter tho Pesch	IRT	Formatting of Anglatecnics input. Comments added, minor text changes.
0.5	08.08.2018	Peter tho Pesch	IRT	Finalized IRT input
0.6	08.08.2018	Chris Hughes	USAL	USAL input
0.9	08.08.2018	Peter tho Pesch	IRT	Minor updates, formatting and fixes Version finalized for review
0.91	17.08.2018	Enric Torres	ANGLA	Anglatècnic review
1.0	05.09.18	Peter tho Pesch	IRT	Final version
1.9	30.09.19	Peter tho Pesch	IRT	Version for internal review
1.95	30.09.19	Kimiasadat Mirehbar	ANGLA	Review comments
2.0	02.10.10	Peter tho Pesch	IRT	Final version

Disclaimer

Statement of originality:

EXECUTIVE SUMMARY

This document describes the subtitles production tools developed in T4.1 and the accompanying work. The software tools that have been developed, are the actual deliverables – this document describes their concepts and functionalities.

In T4.1, the project partners explored how professional users can create and edit subtitle files for 360° videos. Additionally, T4.1 contributed to the design and concept for presentation manners of subtitles in 360° videos. Besides implementation of editor software, prototypes have been realized to run internal tests before the selected subtitle features have been put through end user tests.

The requirements for the work done in T4.1 derived mainly from the user requirements that have been discovered in T2.3 – for both, end users and professional editors. But in addition, researches from other parties have been studied. Their results were introduced into some of the subtitle presentation concepts that have been explored.

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Enric	Torres Feixas	ANGLA	enric@anglatecnic.com
Kimiasadat	Mirehbar	ANGLA	kimia@anglatecnic.com
Àlex	Soler de Ferrater	ANGLA	alex@anglatecnic.com
Peter	tho Pesch	IRT	thopesch@irt.de
Chris	Hughes	USAL	c.j.hughes@salford.ac.uk

CONTENTS

TABLES OF FIGURES AND TABLES

List of figures

List of tables

LIST OF ACRONYMS

Acronym	Description
ACM	Accessibility Content Manager
FoV	Field of View
GUI	Graphical User Interface
HMD	Head Mounted Display
HUR	Home User Requirements
IMSC	Internet Media Subtitles and Captions
PUR	Professional User Requirements
ST	Subtitle
TC	Timecode
TTML	Timed Text Markup Language
VR	Virtual Reality
VRIF	Virtual Reality Industry Forum
W3C	World Wide Web Consortium

1. INTRODUCTION

1.1. Purpose of this document

This document provides an overview of developed tools for subtitle production for immersive content. It accompanies the software products and describe their main features and their availability.

1.2. Scope of this document

This document aims to be a guide through the progress made in development of the subtitle production tools of the ImAc platform. Whereas the (software) tools are the actual deliverables, this document describes their concepts and functionalities and puts them in context of the overall ImAc audio workflow.

1.3. Status of this document

This document has been developed as an iterative document reflecting the status of the subtitle production tools. A first version of this document has been released after the first development phase in month 12. This is the second and final version of this deliverable, addressing the improvements on the subtitle production tools that have been made in year two.

1.4. Relation with other ImAc activities

Figure 1 shows the relation between Deliverable 4.1 and other ImAc activities:

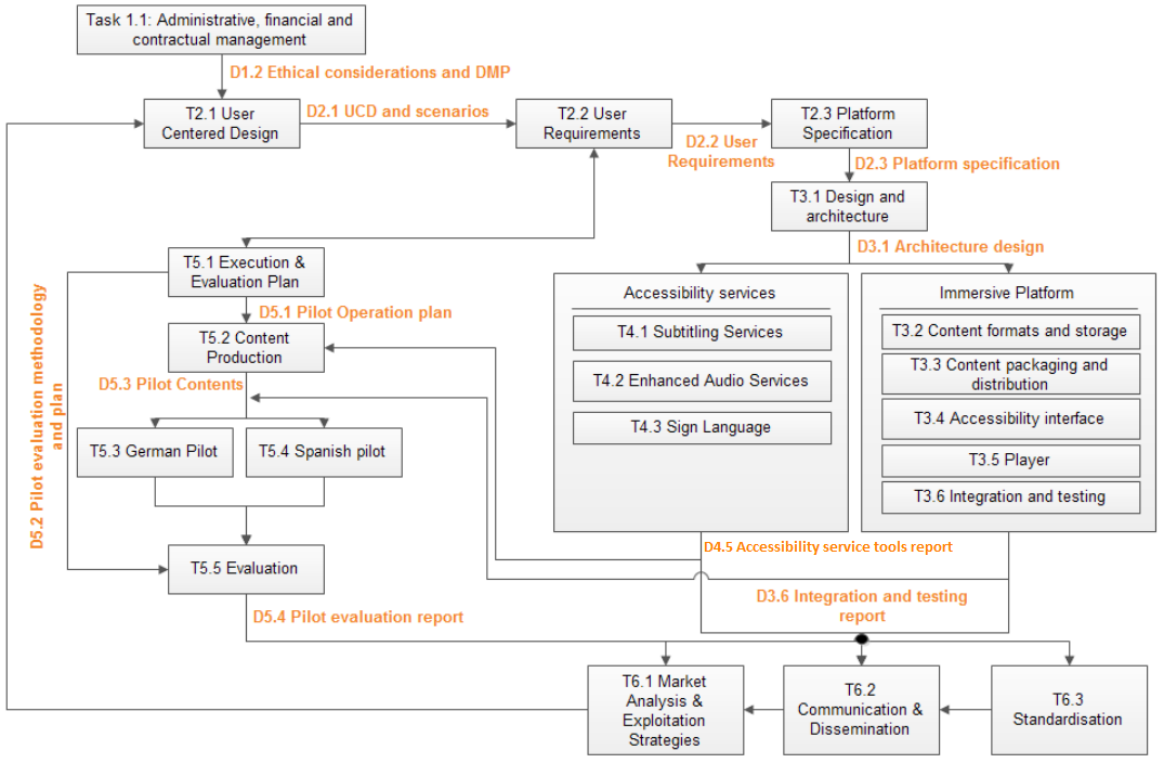


Figure 1 - Dependencies between tasks and deliverables in ImAc

Related activities include especially:

- D4.5 [1]: In deliverable 4.5 the status of all production tools is documented. Additionally, it contains a description of future work in ImAc.
- Integration in Workflow: Subtitle production tools are part of the overall ImAc workflow. The subtitle editor (refer to section 2) is directly integrated in the Accessibility Content Manager (ACM) such that subtitle files that were created are directly attached to the content.
- Related standardisation activities: Part of the work in ImAc is to describe a subtitle format for immersive media, based on the IMSC format. The work from the IMSC_VR authoring tool (section 4) is the leading activity that feeds into an immersive subtitle format. A detailed report on accessibility formats, including subtitle formats was published in D4.4 [2].

2. SUBTITLE WORKFLOW

This section provides an overview of the integration of the subtitle production workflow within the ImAc platform. Different views on the technical components in ImAc, including subtitle production tools, can also be found in ImAc Deliverable D3.1 [3].

Subtitle production is managed via the Accessibility Content Manager (ACM) which provides access to the low-res video and links the produced subtitle files to the asset (i.e. the program) the subtitle was created for. The Web ST Editor provides all functionalities to produce and author a subtitle file. Additionally, subtitle files can be imported and authored in the editor. Via the import functionality, files can be re-used that were created with third party editors and do not include additional metadata used for the 360° representation. In that scenario, the Web ST Editor is used to author only 360° relevant metadata.

The implementations on responsive subtitles and additional presentation modes do not result in a module that is used as part of the ImAc platform directly. Instead, the results were integrated in the ImAc player.

3. IMMERSIVE SUBTITLE EDITOR

3.1. Introduction

In order to put into reality immersive accessibility, access services editing tools are significant. The Web ST Editor is designed for this purpose in the case of subtitling access service.

This section describes the concepts and structural position of the Web ST Editor and the iterative path of this tool along ImAc project. The software is not publicly available at the moment and the code is also confidential.

The editor allows users to edit/produce subtitles for 360° media content. It supports editing typical subtitle attributes (e.g. subtitle timing and styling). Additionally, the user is able to add positional information in 360° scene for every subtitle which makes it appropriate for immersive content. This information is required by the ImAc player (described in Deliverable D3.5 [4]) in order to provide icons (tests were conducted with arrows and a compass), that indicate a speaker's position in the scene. This section describes the main concepts of the editor and also compares the tool in the first and second iteration of the project. In the same matter, at the end of this document the reader may find: The installation guideline (Annex I) as well as a brief user manual (Annex II).

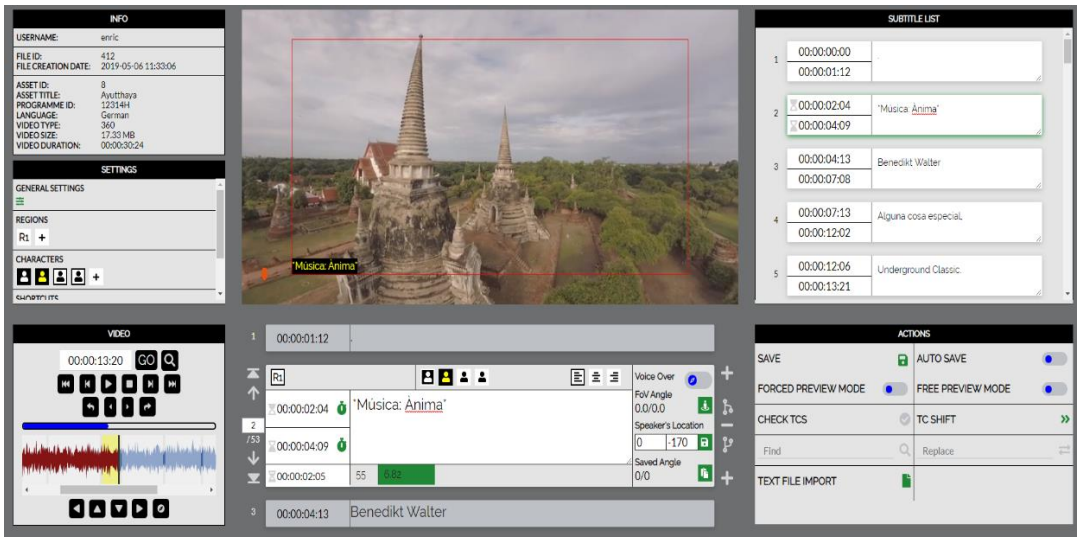


Figure 2 - User interface of the Web ST editor

3.2. Structure and Concept

This section describes the tool's main concepts and its position in structural ImAc workflow.

This editor delivers XML-based subtitle files with classic subtitle attributes alongside the information about the subtitle position in spherical space. Below an example of the delivered xml file is displayed with a number of common attributes:

Styling attributes:

```
<tt:styling>
    <tt:style xml:id="C1" tts:color="#FFFFFF" tts:fontSize="100%"/>
    <tt:style xml:id="C2" tts:color="#FFFF00" tts:fontSize="100%"/>
    <tt:style xml:id="C3" tts:color="#00FFFF" tts:fontSize="100%"/>
    <tt:style xml:id="C4" tts:color="#00FF00" tts:fontSize="100%"/>
    <tt:style xml:id="C5" tts:color="#FF00FF" tts:fontSize="100%"/>
    <tt:style xml:id="C6" tts:color="#0000FF" tts:fontSize="100%"/>
    <tt:style xml:id="C7" tts:color="#FF0000" tts:fontSize="100%"/>
    <tt:style xml:id="C8" tts:color="#000000" tts:fontSize="100%"/>
    <tt:style xml:id="A1" tts:textAlign="left"/>
    <tt:style xml:id="A2" tts:textAlign="center"/>
    <tt:style xml:id="A3" tts:textAlign="right"/>
    <tt:style xml:id="G1"/>
</tt:styling>
```

Layout attributes:

```
<tt:layout>
    <tt:region xml:id="R0" tts:origin="2.50% 54%" tts:extent="95% 40.00%" tts:
    <tt:region xml:id="R1" tts:origin="2.50% 6%" tts:extent="95% 40.00%" tts:d
    <tt:region xml:id="R2" tts:origin="2.50% 6%" tts:extent="40.00% 40.00%" tt
    <tt:region xml:id="R3" tts:origin="2.50% 54%" tts:extent="40.00% 40.00%" t
    <tt:region xml:id="R4" tts:origin="58% 6%" tts:extent="40.00% 40.00%" tts:
    <tt:region xml:id="R5" tts:origin="58% 54%" tts:extent="40.00% 40.00%" tts
</tt:layout>
<imac:coordinateSystem>OMAF</imac:coordinateSystem>
```

Body attributes:

```
<tt:body>
    <tt:div style="G1">
        <tt:p xml:id="p1" region="R0" style="A2" begin="00:00:00.000" end="00:00:00.000"
        imac:equirectangularLongitude="0.0"
        imac:equirectangularLatitude="0">
            <tt:span style=""/>
        </tt:p>
    </tt:div>
</tt:body>
```

The XML attributes visible in the codes above shed a light on the variables defined by the editor. These are required by the ImAc player.

In addition to the ST file delivered by the editor, it is also essential to recognize the editor position in the access service production ImAc workflow. For recognizing Web ST Editor's position in the workflow, knowledge of ACM (Accessibility Content Manager) is essential. The functionality of the ACM, including a user guide can be found in Deliverable D3.2 [3]. The access to this editor is granted via ACM as shown in below chart, after the assignation of a subtitling task to a subtitler via ACM, the professional user is able to see the task in Editor interface and get access to the editor directly:

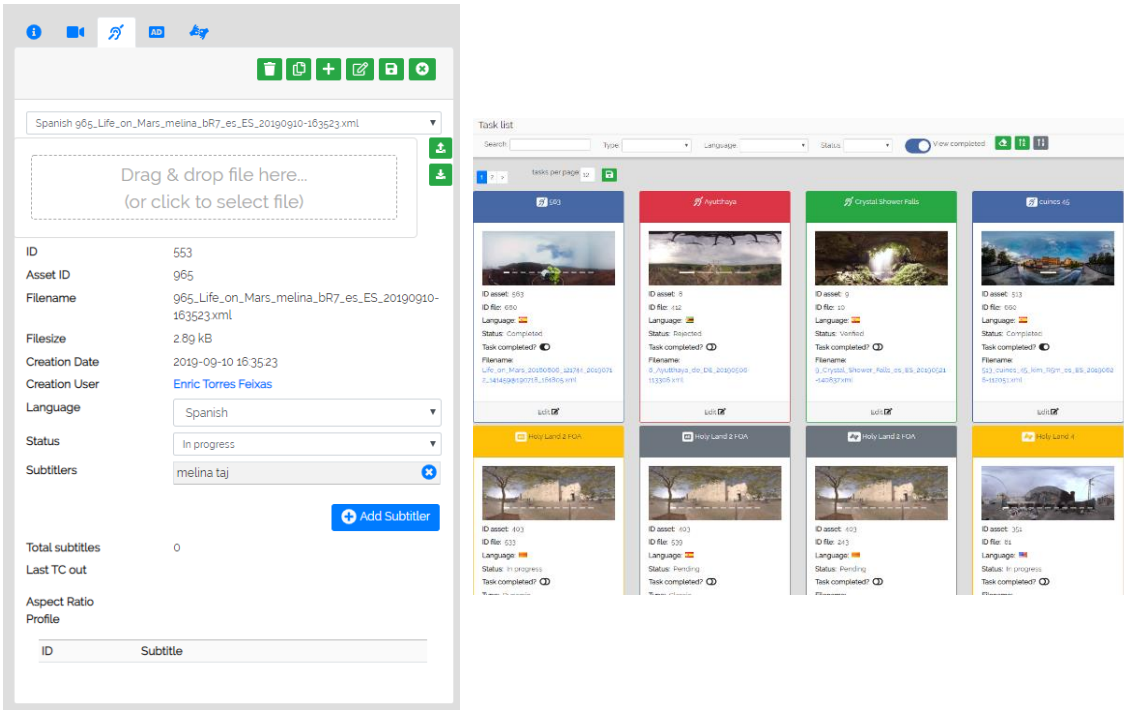


Figure 3 - User interface of the ACM, showing the integration of the Web ST editor in the management tool.

The main characteristics of the Web ST editor are as follows:

- Supporting 360º videos
- Cloud-based
- Based on working with LQ videos transferred from ACM
- Easy to work time-cueing
- Sound-wave of the video and graphical demonstration of subtitles in time
- Editing and verification preview modes
- Personalized user experience due to extremely user-based SETTINGS

3.3. Improvements of second iteration

Table 1, demonstrates the iterative path of the editor. In the second iteration, professional users pilot test result, consortium considerations and EC reviewers have been taken into account in order to transform the editor into a more feasible tool. The table does not reflect full features of the editor but merely highlights the changes made into the tool in the second iteration of the project.

First iteration vs. second iteration		
	1st iteration	2nd iteration
User friendly environment	No	Yes
Prototype vs. exploitation	Prototype	Exploitation
Video sound-wave in time	No	Yes
Shortcut keyboard buttons	No	Yes
Time cue syntax check	No	Yes
User personalised settings	No	Yes
Copy and paste angles	No	Yes
Join and split subtitle text	No	Yes
Character style customization	No	Yes
Subtitle region customization	No	Yes
Graphical subtitle time-codes demonstration	No	Yes
Text file import from PC	No	Yes
Subtitle angle index in video preview	No	Yes

Table 1 - Web ST Editor iterative comparison

The following two screenshots show the enhancements of the user interface within the second project year.

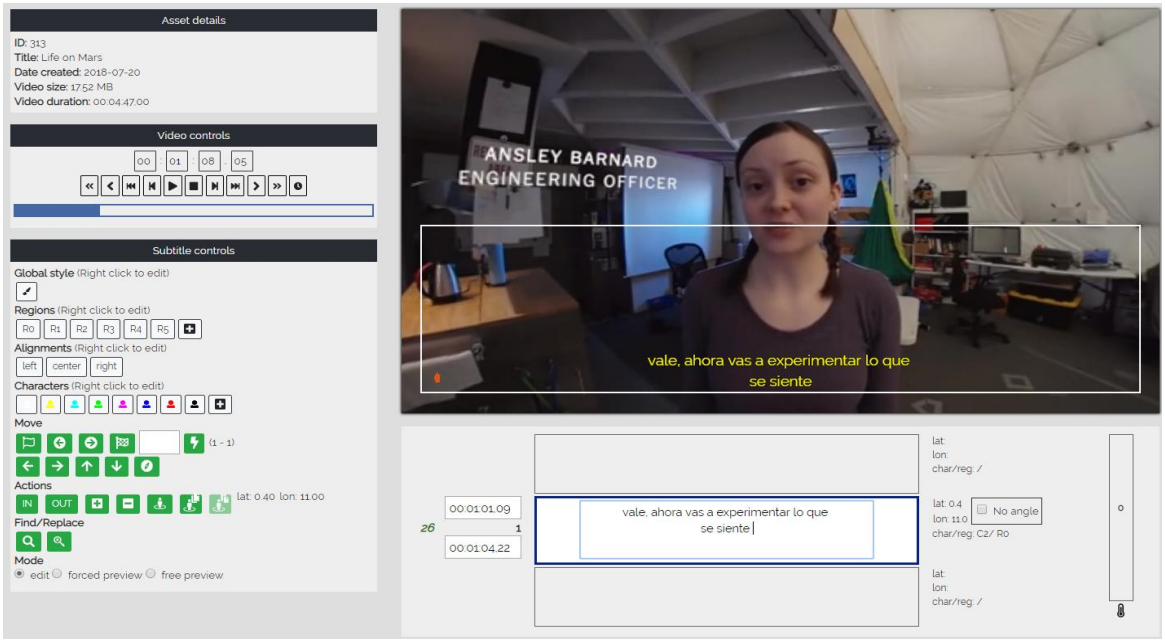


Figure 4 - User interface of Web ST Editor after the first project year

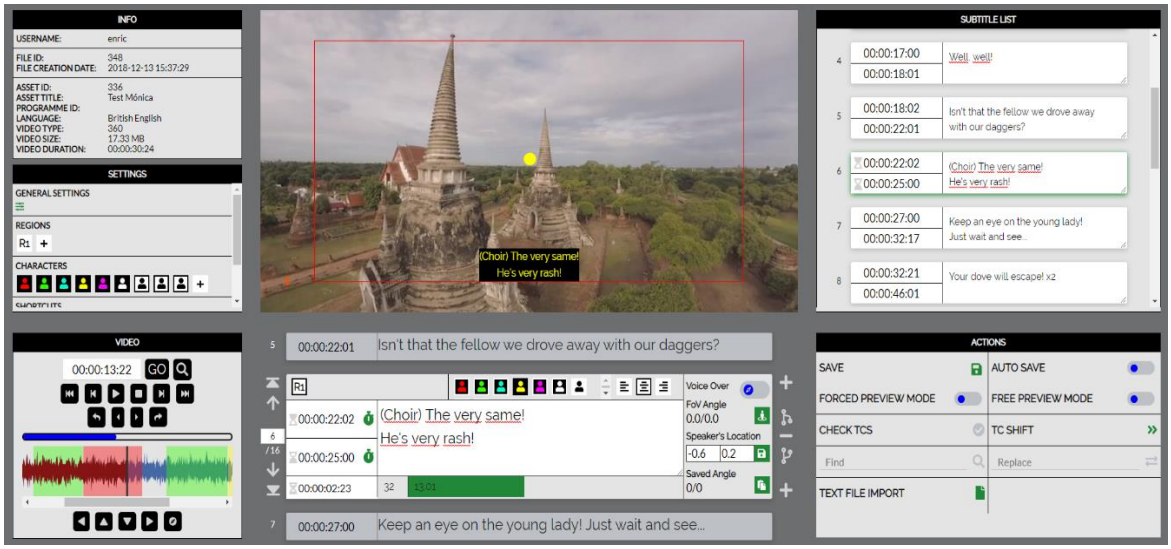


Figure 5 - User Interface of the Web ST Editor after the second project year

4. IMSC_VR PROTOTYPES

4.1. Introduction

Under this section, the work on prototypes is described, that were implemented to test and demonstrate different presentation options for subtitle in 360° as well as to support the definition of IMSC extensions that carry data relevant for 360° scenes.

As a first part, sections 4.2 and 4.3 describe the general approaches for subtitle presentation.

A prototype was published as a web application that demonstrate specific characteristics of subtitle presentation in 360°. A functional description is provided in section 4.4. The application supports two basic presentation approaches and a range of rendering parameters (like font styling and geometrical modes) that can be manipulated by the user. Main goal of the tool is to evaluate relevant subtitle rendering parameters, both static and dynamical ones and link them to the subtitle editor GUI.

The results were fed into ImAc's subtitle editor tools as well as the ImAc player application. Additionally, the web pages were used for demo purposes in standardisation meetings, mainly of the W3C, but also of VRIF.

4.2. Exploring subtitle representation

When subtitles are provided in a 360° environment, they are typically fixed at the bottom of the screen and centered horizontally. This is fine for common 2D TV, but maybe is not the best strategy for 360° media. By making use of the additional "dimension", maybe subtitles can convey more than just textual information. Whereas audio can guide a viewer through a 360° scene – by helping track the action and keeping the orientation – for people with hearing impairments such spatial cues are not available. Can similar cues be realized with an enhanced subtitle service? And how do viewers respond to them?

Thus, the main questions to be answered are:

- What purpose do subtitles in 360° media serve?
- And how can this purpose be fulfilled best?

To address these questions, alternative options for the presentation of subtitles are developed and evaluated in T4.1.

2D and 360° subtitles from an authoring perspective

Compared to "traditional" or 2D video, the presentation style in a 360° or VR environment happens in a different space and thus induces changes on the creation of subtitles. (In the following, the terms 2D and 360° video will be used).

To show the differences between 2D and 360° we will first take a closer look at "traditional" subtitles and their authoring process. Some of the key points are:

1. Subtitles are placed on a static field of view and on a fixed video. With "fixed video", we mean that the video is shown completely as it is stored in a media file and there is no interaction with the user (or subtitle editor) that influences the way the video is shown.

2. For the authoring process, the video can be seen as a flat plane that will always be rendered/shown the same way.
3. As a result from 1) and 2), the subtitles are placed at a position that is absolute to both video and field of view. The subtitle author can see and decide on which area of the video the subtitle will be shown (e.g. top, bottom).

Note: *There are web players that offer personalization options, that allow for example to change the font size or move subtitles to a different position on the screen (e.g. "always top" instead of "always bottom"). For now, we will put these options aside.*

What is positioning used for? Usually, in 2D video subtitles are placed at the bottom of the screen. A different vertical position is used for example to avoid obstructing graphics and inserts. The horizontal position is sometimes used to support speaker identification. For example, the subtitle for a person that appears at the left side of a video may be placed left aligned (typically, however, speaker identification is usually done by choosing different font colours).

What is different in 360°? Obviously, the media presentation sphere does not correlate with the field of view. Only a part of the media is shown at any time. Typically, the viewer has the freedom to choose which part to look at. Sometimes he can zoom in or out too. When showing 360° media on a PC or tablet, the term "magic window" is often used.

Subtitles for 360° can be rendered in various ways. To outline some impacts of 360°, we will look at two possibilities, which are discussed in the following section:

1. Use the field of view as reference for subtitles (refer to 4.3.1)
2. Use the video projection sphere as reference for subtitles (refer to 4.3.2)

4.3. Approaches for subtitle rendering

In pilot phase 1, ImAc uses the approach described in section 4.3.1 "Field of view (magic window) as reference". One of the main user requirements gathered by the ImAc project was that a subtitle should always be visible.

In a pre-pilot phase in the second year of the project, user tests were conducted to explore the second approach, described in section 4.3.2 "Video sphere as reference".

Test results for ImAc user tests are described in detail in Deliverable D5.4 [5].

4.3.1. Field of view (magic window) as reference

When subtitles correlate with the field of view (or magic window), they can be handled similarly as for 2D, because the field of view has similar attributes as the 2D presentation plane does:

- The field of view can be seen as a simple 2D rendering plane.
- Positioning of subtitles can be done as for a "traditional" screen.
- The ratio font size / field of view is the same as in 2D.

The main difference to a 2D presentation is that the subtitle does not correlate with the video. That leads to various effects, e.g.:

- The font size does not change when viewer zooms in or out.
- The author cannot ensure that subtitles do not obscure important areas of the video.

Additionally, when watching the video via a PC/tablet, the subtitle will not be part of the 360° scene, and thus, will not be distorted as the video is. That may compromise the immersive experience.

Modification for this approach

One piece of information that is missing in this scenario is an indication for the speaker's (or sound source) location in the scene. First user tests in ImAc (refer to D2.2, user requirement number HUR.02.31.0) suggest, that it helps users when an arrow next to the subtitle line points towards the speaker position.

In order to render this arrow, the information of the speaker's position must be obtained. In ImAc, two attributes were added to the "p"-element, that represents a subtitle block. The location of a speaker in a 360° scene is expressed by a longitude and a latitude value. Both attributes are optional and may be missing for instance in case of an "off speaker". Figure 6 shows an example:

```
<tt:p xml:id="p18" region="R1" style="S2"
    imac:equirectangularLongitude="158"
    imac:equirectangularLatitude="9"
    begin="00:00:25.243" end="00:00:27.723">
  <tt:span style="S5">Wir versuchen das</tt:span>
  <tt:br/>
  <tt:span style="S5">in die Sendung zu bekommen.</tt:span>
</tt:p>
```

Figure 6 - Subtitle block from an IMSC subtitle file, including spatial information of the speaker in the 360° scene in form of longitude and latitude.

This information is added to the existing styling attributes, that define the subtitles position and appearance within the rendering area:

```
<tt:styling>
    <tt:style xml:id="S5" tts:color="#00ffff" tts:backgroundColor="#000000c2"
        tts:fontSize="100%"/>
    <tt:style xml:id="S2" tts:textAlign="center"/>
</tt:styling>

<tt:layout>
    <tt:region xml:id="R1" tts:origin="10% 10%" tts:extent="80% 80%"
        tts:displayAlign="after" tts:writingMode="lrtb"
        tts:showBackground="always" tts:overflow="visible"/>
</tt:layout>
```

Figure 7 - style and position parameters in an IMSC subtitle file.

The Deliverable D4.4 [2] summarizes the file extensions that were finally used in the ImAc platform.

4.3.2. Video sphere as reference:

In this case, we assume that the surface where subtitles are rendered on, is equal to the video projection sphere (regarding dimension and geometry). So basically, the subtitle is part of the scene and reacts to user interaction in the same way, as the video does. For instance:

- A subtitle may move out of view when the viewer turns his head.
- A subtitle always covers a specific area of the video. It can be positioned such that it doesn't obscure important areas of the video.
- Subtitles zoom when the video is zoomed.
- Subtitles are affected by lens distortion the same way that the video is, when played on a PC or Tablet.

Modifications of this approach

This approach might provide the more immersive experience but has some drawbacks that need to be investigated. In the following some possible modifications of the basic approach are discussed.

Project subtitles on a plane instead of a sphere: First implementations suggest that rendering subtitles onto a plane instead of a sphere are more convenient to read. The plane with the size of the subtitle can be added to the scene. Position and orientation can be chosen such that the subtitle appears (approximately) at the position where it would be when it is rendered onto a sphere. That would lead to slight aberrations between subtitle and video. In the following picture the difference can be seen:

Figure 8 - Two different rendering options for subtitles. Left: subtitle is rendered onto a sphere. Right: Subtitle is rendered onto a plane.

Another variation was developed where an additional subtitle is rendered into the field of view whenever the speaker (and "main" subtitle) is out of view. This second "assistance" subtitle is rendered with a smaller font size and thus less comfortable to read. It is considered a compromise so that the viewer may look around during a conversation for a short time without missing information. An arrow next to this assistance subtitle points the viewer towards the speaker.

A sample can be seen in Figure 9. The green subtitle belongs to a man who is partly visible at the right edge of the viewport. He is talking to a woman to his left (in the scene she would be further to the right). The subtitle that belongs to her is not visible and an assistance subtitle including arrow is drawn to the screen.

Figure 9 - Sample of assistance subtitle (yellow), when subtitles are fixed to speaker and speaker is out of view

After internal discussions, the project partners have decided to test subtitles fixed to the scene (or the video) where an arrow (i.e. without the subtitle's text) is shown to point towards the speaker when he or she is outside the current viewport.

4.4. IMSC_VR prototype feature description

The following section describes the features that were implemented in the renderer tool and parameters that can be modified. During the project, the tool was updated several times and tailored to different topics being investigated in the project. Not all functionalities were kept within the websites that are still available. Two versions which can be found here:

http://subtitling.irt.de/vr_edit/subtitle_demo/
http://subtitling.irt.de/vr_edit/foxfinder_sample/

The following figures exemplarily show three versions of different test setups. The single parameters are explained in sections 4.4.1 - 4.4.3.

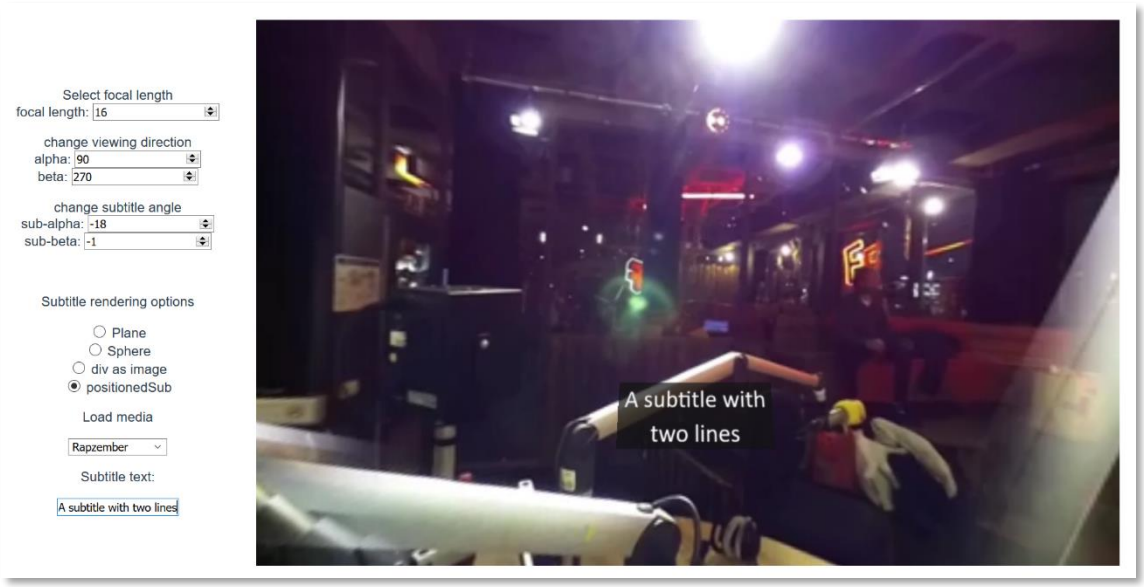


Figure 10 - Early version. Main feature to test here was the option to dynamically move subtitles left or right depending on the speakers location, while the subtitle always stays in the FoV.

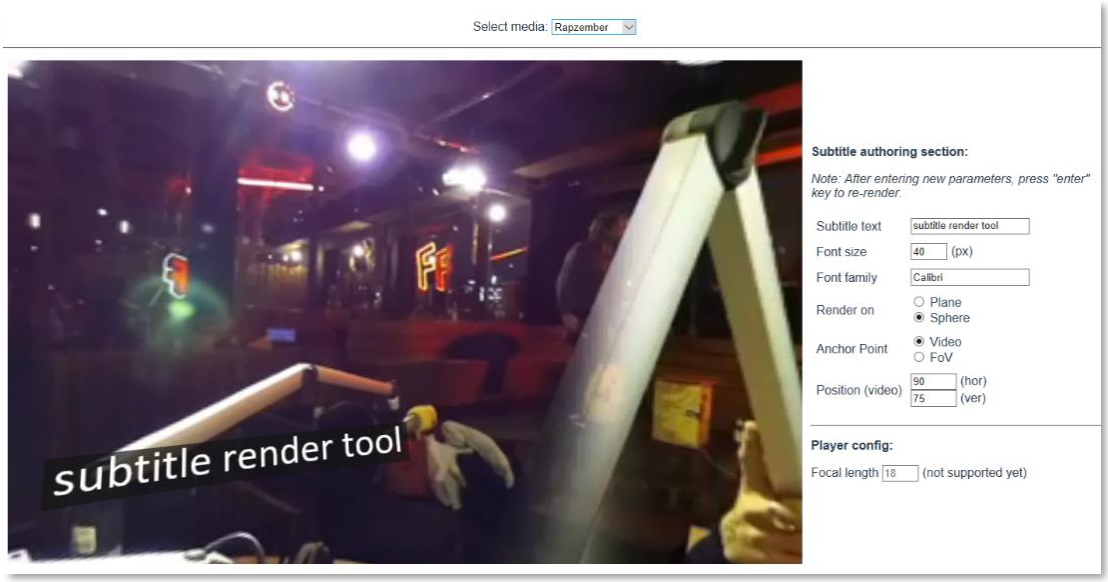


Figure 11 - Interface that allows to compare subtitle appearance (font, font size, plane or sphere rendering plane).

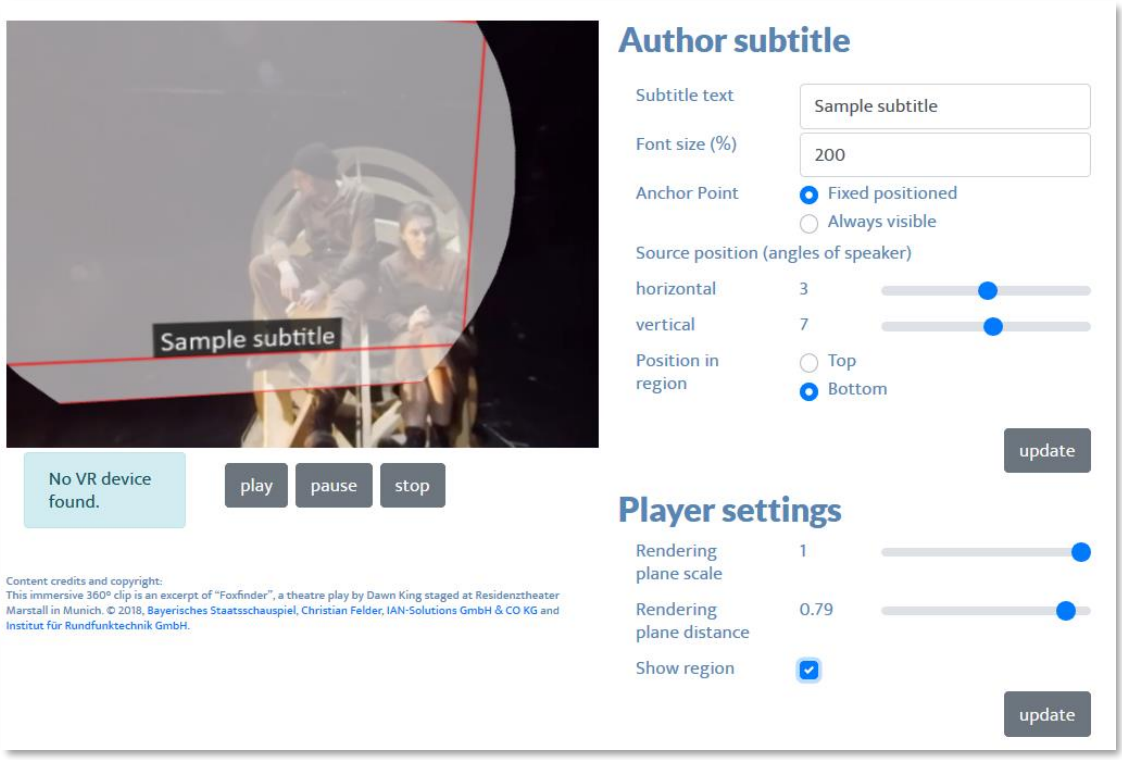


Figure 12 - Interface of the version used to demonstrate correlation between distance and size of the rendering area. The "show region" option allows to check the potition and size of the subtitle rendering plane as well as the defined IMSC region.

The following sections describe all parameters that were explored throughout the project. The features were discussed with project partners and external stakeholders. Selected features have been tested with users. A summary on user tests and their results can be found in Deliverable D5.4 [5]

4.4.1. Presentation strategies:

A few presentation strategies for subtitles have been implemented for testing. Figure 13 shows an early version of the subtitle renderer which was able to render subtitles that moved left and right depending on the speaker's location in the scene.

This setting is about testing different strategies for rendering which includes the subtitle position and how it responds to head movements. Currently two modes are supported: rendering onto the FoV (as described in 3.3.2) and rendering onto the video (as described in 3.3.1).

Researches presented by Sylvia Rothe at TVX 2018 [6] showed, that it is more comfortable for viewers when subtitles are fixed to the video. That conflicts with the user requirement that subtitles should always be visible (i.e. be shown within the field of view). Neither of the two strategies was preferred over the other, by the testing group. The implementation allows to compare the two basic strategies.

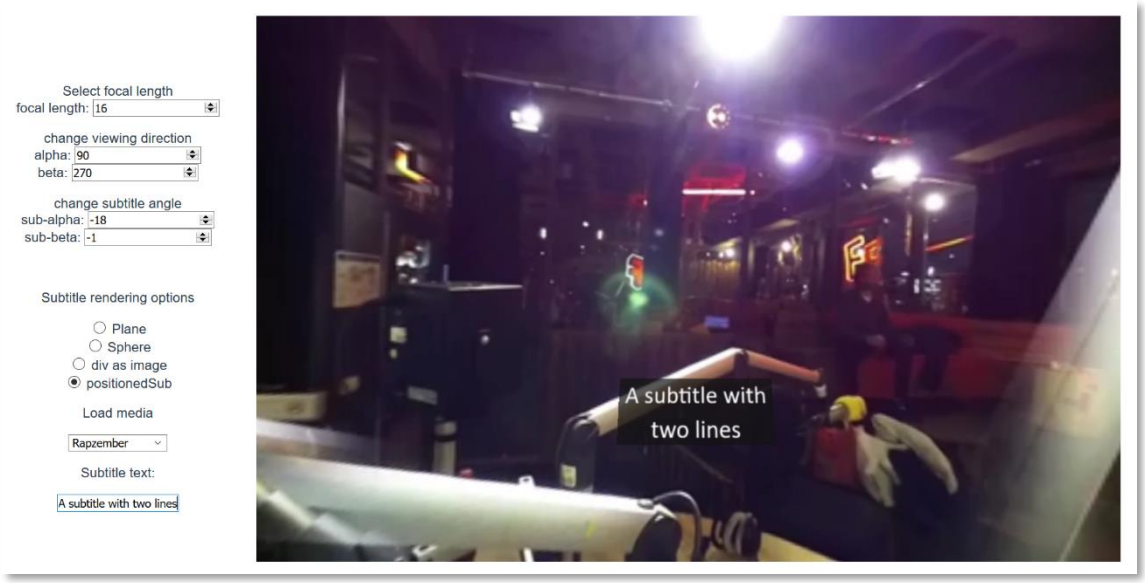


Figure 13 - Early version of subtitle lab tool for testing different presentation strategies, including one that moved subtitles left or right, depending on the speaker's location.

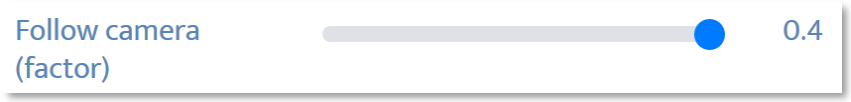
Additionally, the feature described in section 4.3.2 was implemented. In this approach, the subtitle is rendered fixed to the video, but an additional subtitle is rendered into the field of view whenever the speaker (and "main" subtitle) is out of view. This second "assistance" subtitle is rendered with a smaller font size and thus less comfortable to read. It is considered a compromise so that the viewer may look around during a conversation for a short time without missing information. A sample is shown in Figure 9, a demo can be seen at:

http://subtitling.irt.de/vr_edit/foxfinder_sample/.

The same demo page shows another feature that dynamically adapts the position of a subtitle when fixed to the video. An example is shown below (Figure 14) where the subtitle is placed on the speaker (first image), but follows the head movement up to a certain amount, when the user turns away (second and third image). That way, the subtitle is visible for a larger range of viewing directions.

Figure 14 - Three snapshots of a video showing subtitles that are placed on a speaker but follow the viewing direction of the user up to a certain amount.

The amount that the subtitle “follows” the viewers viewing direction can be set with a parameter (here called “follow cam”):



4.4.2. Positioning

The implementation includes the possibility to set the position of the subtitle in the scene. That means, the chosen position is mapped to the center of the rendering plane of the subtitle. In the rendering plane, the subtitle is rendered according to the styling and layout attributes of the IMSC file. When the subtitle is rendered fixed to the viewport, the subtitle is rendered only according to the IMSC file and the rendering plane spans over the entire viewport.

In an early implementation, the position of the subtitle could be set manually when the subtitle is rendered fixed to the viewport, but that was changed, when more IMSC attributes were supported by the prototype and the position could be set via styling and layout attributes.

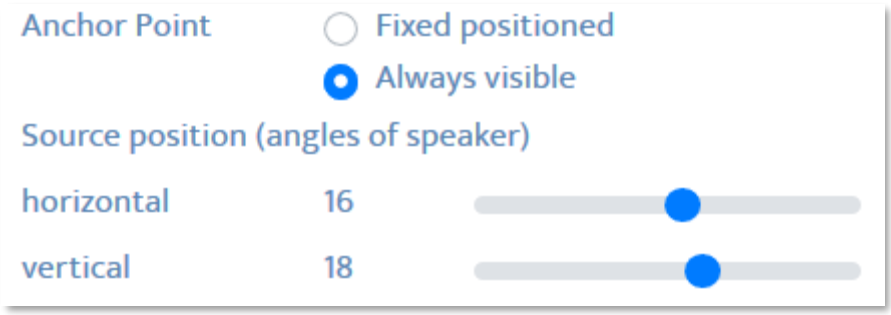


Figure 15 - Authoring area for subtitle positioning

Especially when wearing VR glasses, the exact position of the subtitle rendering plane cannot be seen, simply by watching a subtitle in the scene. To provide a better overview, a feature was implemented to show the rendering plane as well as the region that is defined in IMSC.

The following picture shows an example. On the left side, the rendering plane is scaled down. Note: In this implementation, a scale value of “1” results in a rendering plane that will fill the entire viewport. That means, changing the distance parameter will also change the size of the rendering plane. On the right side, the size of the rendering plane is set to “1”, and it can be seen, that the plane intersects with the sphere where the video is projected on.

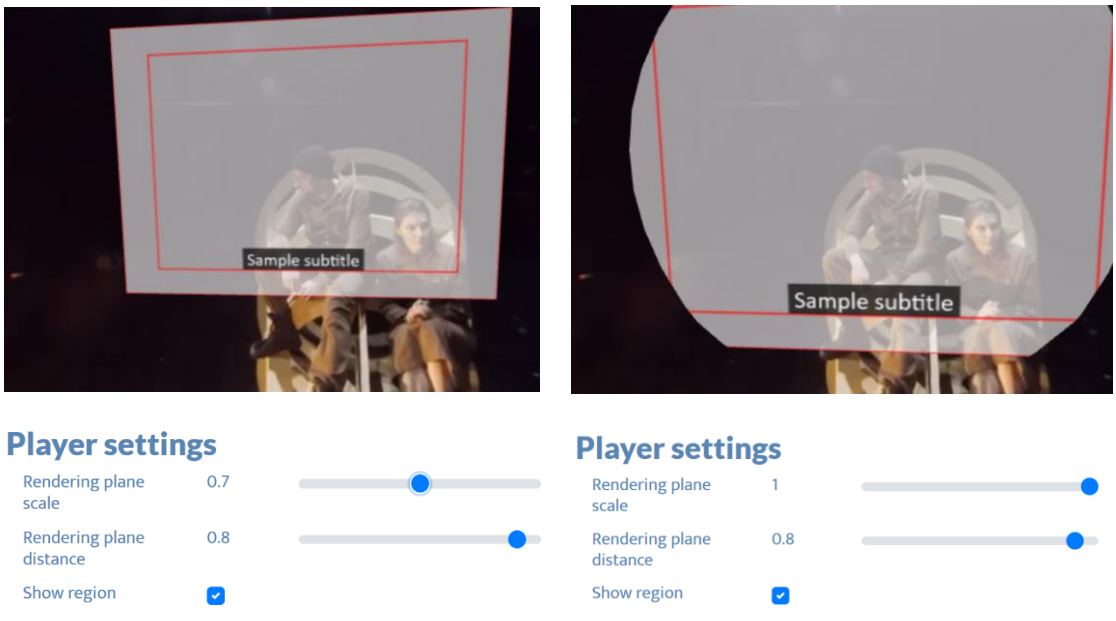


Figure 16 - Sample showing the rendering plane as well as the IMSC region of a subtitle. Both size and distance can be modified in this demo.

4.4.3. Subtitle appearance - Font family, font size, rendering mesh

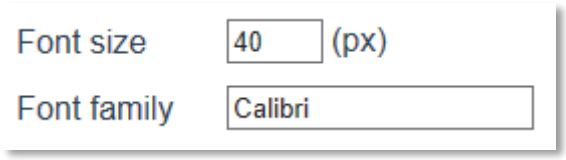


Figure 17 - Authoring section for font properties.

The font family can help to support legibility of the subtitles. That is especially true for low resolution screens, where e.g. fonts with serifs can be harder to read. In order to evaluate how users respond to different fonts on different devices, the subtitle renderer allows to set the font family. The font rendering is handled by the browser, i.e. all fonts the local browser supports can be entered.

A possible outcome could be to dynamically choose the font family in respect to the rendering device. For HMDs it is not recommended to use fonts with serifs.

Similar to "font family", the font size is important for legibility of subtitles. Additionally, this is a parameter that is used for personalization. It is a basic setting to test how different font sizes will look like on different devices. The font size was provided as pixel first (see Figure 17), but this was changed and the metric percentage was used later, since it relates to the display size.

Another aspect of subtitle appearance was the geometric figure, the subtitle is projected on. Here, the plane was compared to a sphere as described in section 4.3.2.

For the ImAc platform, the project partners have decided to use a plane as rendering object. The decision was taken based on the preferences of the partners.

5. EXPERIMENTAL RESPONSIVE SUBTITLE

In the world of web design, the challenge of providing web content to many different types of device has been met by an approach called Responsive Web Design also sometimes called Mobile First. Responsive web sites respond to the device size and capability using a fluid approach to text layout where space is limited, whilst going to fixed column widths where there is plenty of space on the screen. This approach overcomes the problem of needing different versions of a web site for each new device, and it is the approach we have adopted to provide a more appropriate way of displaying subtitles, across a range of display devices.

We have followed practices used in responsive web design to provide a framework for responsive subtitles. This adopts the principles of text flow and line length informed by semantic markup along with style sheets to control the final rendering. The subtitle file can also include rendering information to provide recommended areas of the display in which to display the subtitle so that the subtitle does not occlude faces or graphics. The key difference from the traditional approach is that the words are formed into blocks in the client and so the number of words in the subtitle blocks will vary depending on the space available in the display.

Once the decision has been taken to render the subtitle layout in the client this opens the possibility of taking into account the user preferences at the point where the subtitle blocks are rendered. Although personalization of video captions if not a new concept (YouTube and Netflix for example, allow a choice of font, size and colour) our mechanism allows the subtitles to be restructured on demand, creating new subtitle blocks that best fit the display based on the user preferences. In this way the user could increase the size of the font to make the text more readable and the client would then be able to form subtitle blocks with fewer words in order to prevent the subtitles from obscuring too much of the video.

User options can thus be accommodated for a choice of font to meet cognitive issues such as dyslexia and font size to cope with varying visual. Colour is used for speaker identification for TV subtitles and the choice of the four colours could be re-mapped to accommodate the user's colour vision. The text background could also be varied according to the user preferences for an opaque or semi-transparent background, or even no background with outlined text.

5.1. Method

In previous work [7] the advantages of adopting a subtitling approach that adopts techniques used in responsive web site design was demonstrated. The traditional approach to formatting subtitles uses fixed size blocks of text (generally 2 lines, ~30 characters) which although is sufficient for a standard television set up, is far from ideal for immersive displays.

Our responsive approach enables the subtitles to be formatted in the device to fit the display capabilities. It also provides the flexibility to respond to both changes made by the user to personalize the subtitles to fit their own requirements and other graphics within the display. This is achieved by ‘atomizing’ the subtitles in the player, and interpolating timings and formatting for each word from the original subtitle as shown in Figure 18. The ‘atomized’ words can then be reconstructed into subtitles of any length, as required during playback as shown in Figure 19.

Figure 18 - Each subtitle is atomized into words, by interpolating the timing and style from the original caption.

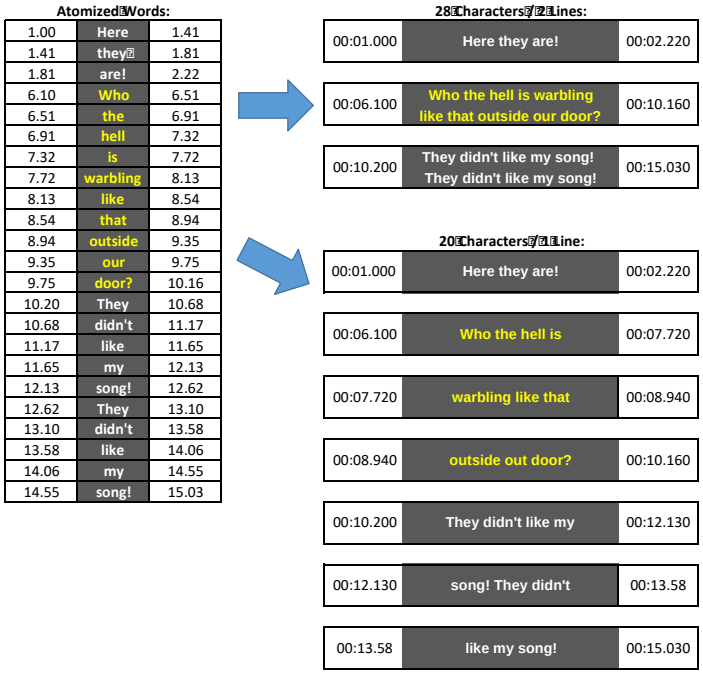


Figure 19 - The atomized words can be reconstructed into phrases of any length.

This means that the block formatting can take account of the device characteristics and user preferences to provide a tailored and completely custom user experience. This approach is particularly effective when adapting content from traditional television displays into an immersive environment, such as rendering the subtitle as speech bubbles attached to a character, or for instance if you wish to reduce the width of the subtitles in order to make room for graphics as demonstrated in Figure 20.

Figure 20 - Responsive subtitles allow the rendering area to be dynamically changed to make space for graphics, or other accessibility services such as a signer.

5.2. Implementation

As part of the ImAc project we have developed a prototype a JavaScript library for generating responsive subtitles. This adopts the principles of text flow and line length informed by semantic mark-up along with styles to control the final rendering. The library provides an extension to IMSC.js (a JavaScript library for rendering IMSC1 Text and Image Profile documents to HTML5) where IMSC documents can be loaded into a TT-object.

Our responsive subtitle library restructures a TT-object based on line character width and line count as shown in Figure 21. (a) IMSC.js converts the TTML document into a TT Object. (b) The responsive library atomizes the TT object into words, preserving an interpolated time and style for each word. (c) The words are reconstructed into phrases split by a pause in the dialogue or a change of speaker. (d) The phrases are subdivided using a best-fit algorithm to meet the line length requirements. (e) a new TT object is generated with IMSC.js.

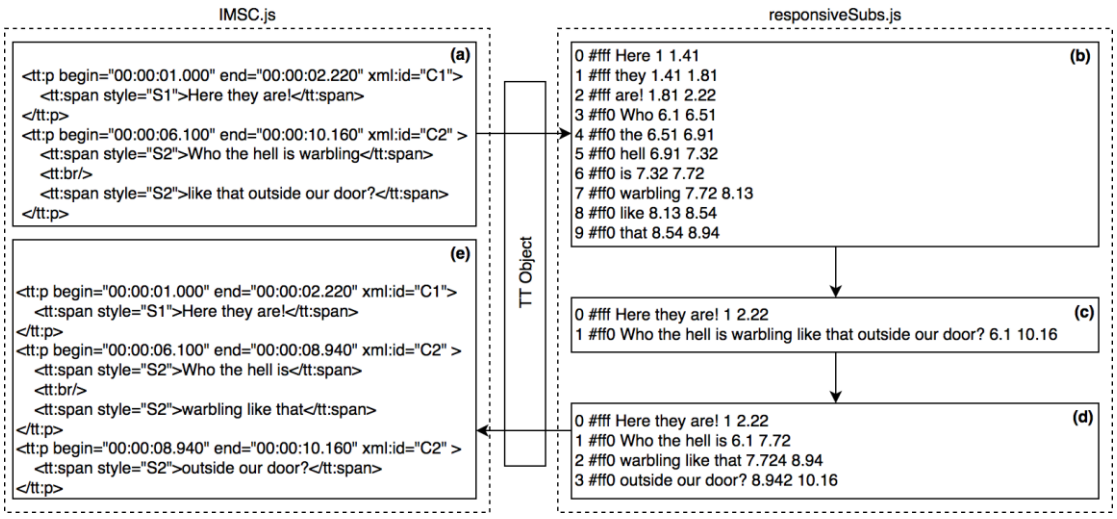


Figure 21 - Restructuring a subtitle file with responsiveSubs.js

By working directly with TT-objects allows this library to be simply connected to any application which already uses the standard IMSC.js implementation allowing customization controls to be retrofitted, such as font size as shown in Figure 22.

Subtitles are re-blocked by adhering to the number of characters that can fit into the display container at the chosen font size. Firstly, ‘paragraphs’ are constructed recombined, based on identifying a distinct segment of the text. The end of each segment is identified, either by a change of speaker or when there is a gap of more than two seconds between subtitles.

A best-fit algorithm then breaks each paragraph up to individual captions in order to fit the container. Due to the nature of the changing font size this may provide more or less captions than the original subtitles, however as the number of words is remains the same the reading speed never changes. As words are evenly distributed it also avoids leaving orphaned words.

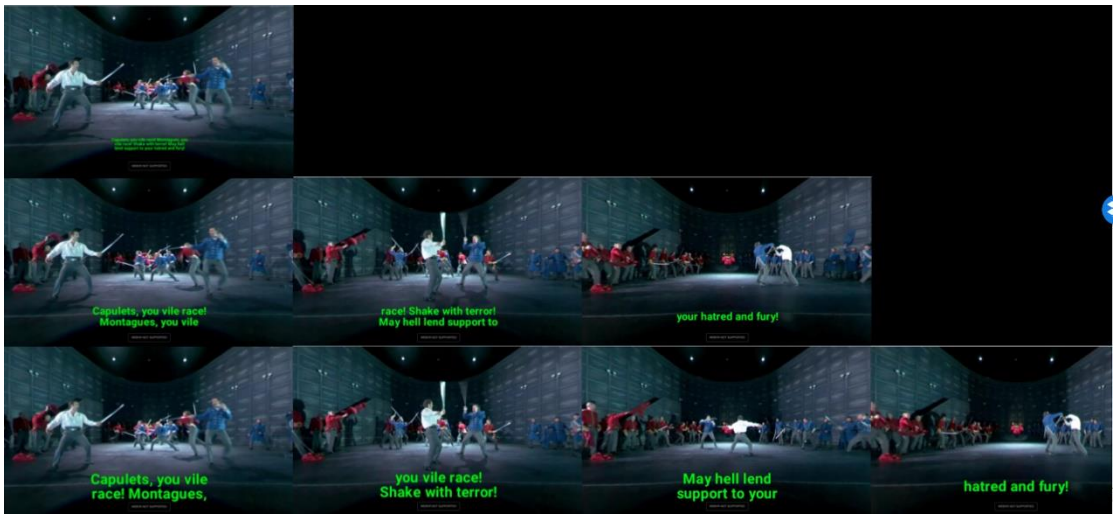


Figure 22 - Responsive subtitles re-blocked based on consumers font size requirements. The subtitles are presented differently when the user chooses a font size of 50%, 100%, 125% and 150%.

The library can also be used in non-linear VR applications, such as a high contrast reproduction of a 360° video. The responsive subtitle library has automatically identified each unique character in the scene and represented each character as a clear cylinder in the VR scene, as shown in Figure 23.

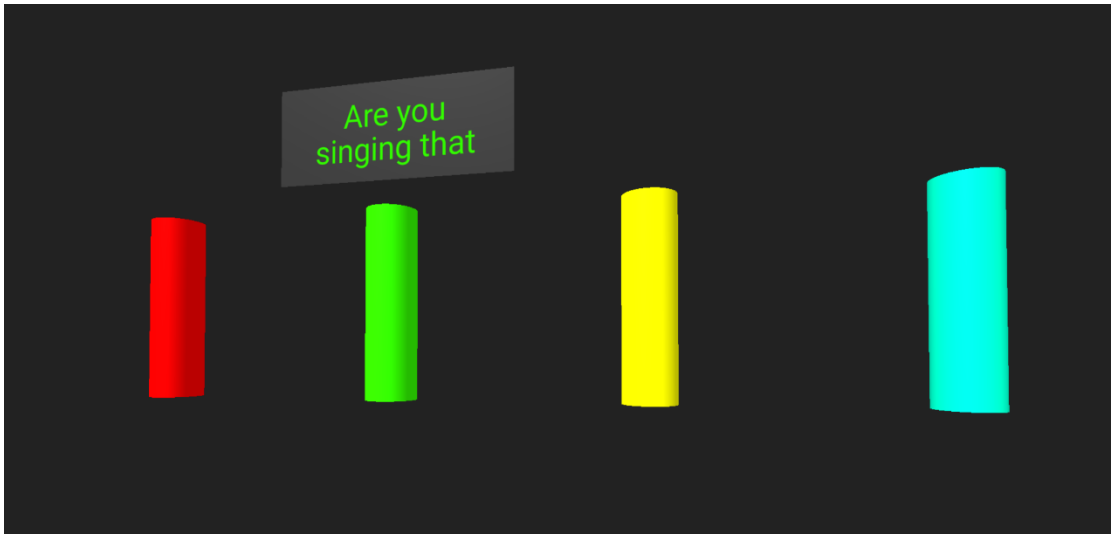


Figure 23 - Responsive subtitles can also be used in non-linear VR applications.

6. REFERENCES

[1] ImAc deliverable D4.5 – Accessibility Service Tools, Revision 2.0 September 2019, http://www.imac-project.eu/documentation/deliverables/

[2] ImAc deliverable D4.4 – Report on new accessibility formats, Revision 1.0, August 2019, http://www.imac-project.eu/documentation/deliverables/

[3] ImAc deliverable D3.1 – Architecture Design, Revision 2.2, July 2019 http://www.imac-project.eu/documentation/deliverables/

[4] ImAc Deliverable D3.5 – Player, Revision 2.1, July 2019, http://www.imac-project.eu/documentation/deliverables/

[5] ImAc Deliverable D5.4 – Pilot evaluation report, Revision 0.4, November 2018 http://www.imac-project.eu/documentation/deliverables/

[6] Sylvia Rothe, Kim Tran, Heinrich Hußmann, Dynamic Subtitles in Cinematic Virtual Reality, in International Conference in Interactive Experiences for Television and Online Video (TVX) 2018, Hanyang University, Seoul, Korea, June 2018

[7] Chris J Hughes, Mike Armstrong, Rhianne Jones and Michael Crabb, Responsive Design for Personalised Subtitles, In 12th Web for All Conference, ACM, Florence, Italy, May 2015

ANNEX I INSTALLATION OF THE WEB ST EDITOR

A How to install

Installation

A Linux operating system is installed, Debian 9 distribution. The 'root' user is used to install and set everything.

Packages

It is required to install a number of packages into the server in order to have all the tools and programs to run it:

```
apt-get update

apt-get install vim screen rsync ntp less man net-tools apache2 php php7.0-mysql php7.0-curl php7.0-gettext php7.0-mbstring php7.0-xml openssl mysql-server

php7.0-odbc curl apt-transport-https php7.0-sybase freetds-common libsybdb5 exim4
```

This will install the main tools for the system:

- Apache Web Server
- MySQL Database (MariaDB)
- PHP 7.0 engine
- System Tools

each package has to be configured.

Note: “Vim” editor is utilised to edit each file. In order to save and exit: “ESC → :wq”

Apache

The user needs to configure the project, its locations and parameters:

```
vim /etc/apache2/sites-available/acm_deliverable.conf
```

```
Alias /acm_deliverable/ /var/www/content_manager_deliverable/html/

<Location /acm_deliverable/>
  order deny,allow
  deny from all
  allow from 172.19.192.0/255.255.255.0
  allow from 127.0.0.1/255.255.255.0
  allow from all
  Options Indexes FollowSymLinks MultiViews
  php_flag magic_quotes_gpc Off
```

```
  php_flag short_open_tag Off
  php_flag register_globals Off
  php_value upload_max_filesize 5G
  php_value post_max_size 5G
  php_value memory_limit 2G
  php_value max_execution_time 600
  php_value max_input_time 60
  php_value max_file_uploads 50
  AddDefaultCharset UTF-8
  php_value session.gc_maxlifetime 14400
  ErrorDocument 404 /acm_deliverable/404_not_found.php
</Location>
```

```
vim /etc/apache2/sites-available/editor_deliverable.conf
```

```
Alias /editor_deliverable/ /var/www/content_manager_deliverable/html/ed/

<Location /editor_deliverable/>
  order deny,allow
  deny from all
  allow from 172.19.192.0/255.255.255.0
  allow from 127.0.0.1/255.255.255.0
  allow from all
  Options Indexes FollowSymLinks MultiViews
  php_flag magic_quotes_gpc Off
  php_flag short_open_tag Off
  php_flag register_globals Off
  php_value upload_max_filesize 5G
  php_value post_max_size 5G
  php_value memory_limit 2G
  php_value max_execution_time 600
  php_value max_input_time 60
  php_value max_file_uploads 50
  AddDefaultCharset UTF-8
  php_value session.gc_maxlifetime 14400
  ErrorDocument 404 /editor_deliverable/404_not_found.php
</Location>
```

As the next step, it is required to upload the Apache Web Server in order to apply this configuration.

```
a2ensite acm
a2ensite editor
service apache2 reload
```

PHP

Logs from the Apache server log trace should be disabled

```
vim /etc/php/7.0/apache2/php.ini
```

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT & ~E_NOTICE
```

CODE

A zipped file will be provided for the installation:

```
content_manager_r20.tgz
```

The user navigates to the web server code folder:

```
cd /var/www/
sudo mkdir content_manager_deliverable
sudo chown www-data:www-data content_manager_deliverable
sudo chmod 775 content_manager_deliverable
cd content_manager_deliverable
```

And unzips the code file:

```
tar xvfz content_manager_rXX.tgz
```

The path to the following file should be configured:

```
vim html/includes.inc.php
```

```
<?php
require_once("/var/www/content_manager_deliverable/includes/connection.inc.ph
p");?>
```

Configured paths and database parameters in the following file:

```
vim includes/config-local.inc.php
```

```
//PATHS
define("PATH_ROOT","/var/www/content_manager_deliverable");
//ROOT
define("ROOT_PAGES","/acm_deliverable");
define("ROOT_PAGES_ED","/editor_deliverable");
```

```
//BBDD
$bbdd_usuari='imac';
$bbdd_pwd='****';
$bbdd_servidor='localhost';
$bbdd_bbdd='content_manager_deliverable';
$bbdd_driver='mysqli';
```

MYSQL

Database is a MariaDB 10.1.26. The user "imac" is added and the database is imported:

```
mysql -u root -p
```

```
GRANT ALL PRIVILEGES ON *.* To 'imac'@'%' IDENTIFIED BY 'aYooph8ietoo';
FLUSH PRIVILEGES;
exit;
```

```
cd /var/www/content_manager/bbdd

mysql -u imac -p content_manager < content_manager_r20.sql
```

In order to import, typing the password is mandatory.

Crontab

In this file, the Linux Task Manager is configured to execute periodically important scripts (generate and clean transcoding):

```
vim /etc/crontab
```

```
#CONTENT_MANAGER
* *  * * *   root /var/www/content_manager/scripts/generate_transcoding.php
```

```
0  0    * * *   root
   /var/www/content_manager/scripts/clean_transcodings.php
```

B How to access the installed interface

Once we have everything installed and set, we should be able to access the Editing Interface normally. To access the current installed version for this delivery:

1. Open your preferred browser (Chrome and Firefox work better with the interface).
2. Browse to the path where the editor has been installed, e.g.: http://yourServerAddress/editor/
3. The main ImAc ACM login web page appears

ANNEX II WEB ST EDITOR USER MANUAL

A What is new

In general, the most brand-new feature of this editor is the possibility to work in 360º environment and the angles. The important matter is why do we work with **angles**?

Since we are subtitling a spherical video, the speakers are not always positioned in a static Field of View (FoV), because they can move in the 360º space. So when we are subtitling, sometimes the speakers will be in our FoV, but if all of a sudden they move, we need to move around to look for the speaker in his or her new position.

In order to tell the subtitling system where the speaker is, we need to look for the speaker in the video moving around with the mouse or the shortcuts. Imagine that a viewer is watching the 360º content at home. OK, the viewer is navigating through the video, BUT gets lost and misses the speaker. Maybe this viewer is deaf, so she lacks the auditory input.

The solution will be that an arrow or compass will appear to indicate where the speaker is. If we, as subtitlers, do not provide this information (or metadata) when generating the subtitles, then it would be impossible for the system to provide this information to the audience when they are playing the video at home. So, this is why the "Set angle" option to position the speaker is important. In next sectionss we see how it is possible to work with angles in the Web ST Editor.

In the case in which you have worked with ImAc Web ST Editor before you will experience new features such as the possibility to **customise shortcuts** and an **improved front-end** for users convenience alongside a **sound wave of the video in time.**

B Before starting

Before starting it is important to be sure that the requirements are met:

- Hardware: PC with at least i5 processor, 8 GB RAM. Screen resolution should be at least 1920×1080 pixels (a good graphics card is recommended).
- Browser: Last version of Chrome or Firefox (at least Chrome version 74 or Firefox version 65).
- Good internet connection as the editors are online and videos are used during the subtitle production.
- Although the video is provided to the subtitle producers it is important to notice that the video provided must be HTML5 compatible Low Quality video to assure that the 360º web player runs smoothly.

Also it is important to be aware of the following:

- The web editors are online tools, so after executing them some features may take some time before they are available such as the waveform and some data in the info box.

C How to start

C.1 Login

User accesses the Editor Interface of ACM via the web browser (illustration i) and enters username and password previously provided by administrator.

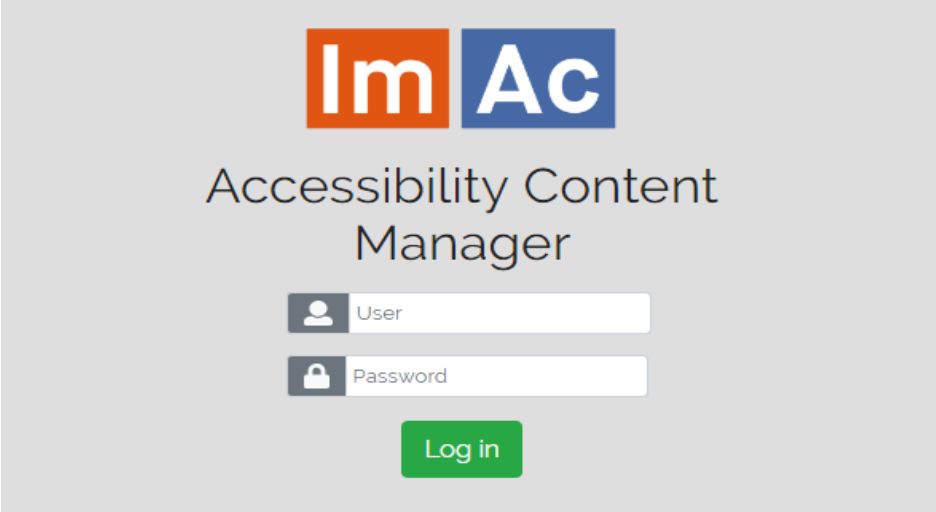


Illustration i: ACM login page

C.2 Navigation on main page

When entered, a window with the list of assigned production tasks (subtitling tasks for the purpose of this document) to the user with their corresponding videos appears (Table i). Then the user can make use of the following tools:

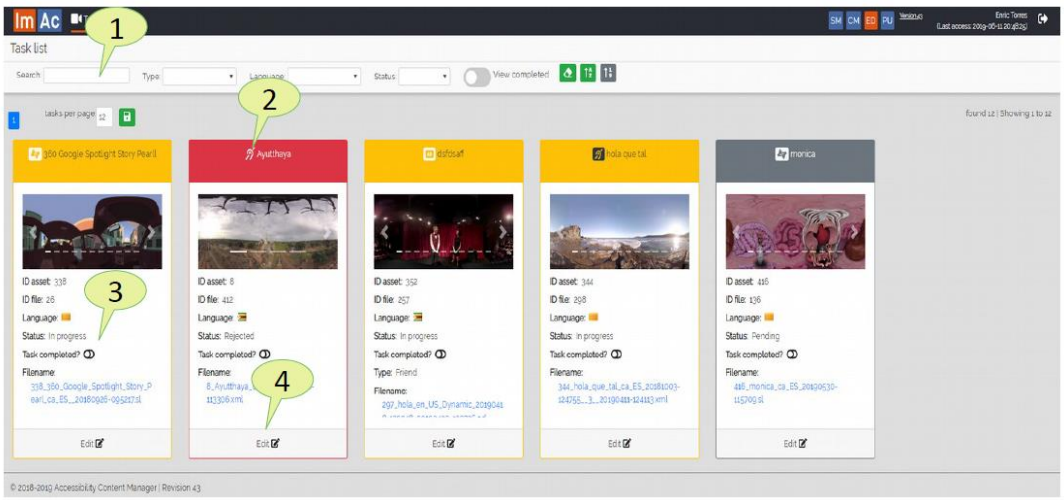
Navigation elements on the Editing interface	
	
Search bar (1)	With this tool, the user can search through their assigned production tasks and view them alphabetically, including completed tasks or not, etc.
Task icon (2)	These icons show the type of production task we are dealing with: it can be either a subtitle, a sign language or an audio description task. For the purpose of this document we will only focus on the subtitling tasks. The colour of the row shows the status of the task (“Pending”: in grey, “In progress”: in yellow, “Completed”: in blue, “Rejected”: in red, and “verified”: in green).
Task status (3)	The user can view the current status of the work and only can changed it to “Completed” when the task is done.
Edit icon (4)	By clicking on this button on a subtitle task (for the purpose of this document) the Web ST Editor will be executed with the video file and the subtitle file for the subtitle production. The same applies with the other access service tasks but with the corresponding editor.

Table i: Navigation elements on the Editing interface

C.3 Web ST Editor

Illustration ii, displays the editor window. This window is divided into an upper and down side. The upper side is only designed for viewing, setting and verification purposes. The down side is purely for editing.

The editor is responsive, so you may wish to set the browser zoom adequately (Ctrl+mouse wheel up or Ctrl+mouse wheel down) to fit all the boxes adequately in the screen.

Illustration ii: Web ST Editor window

D How to produce subtitles

It is assumed that now you are inside the Web ST Editor. Let's take a look at how we use the editor to create a subtitle from scratch.

Your first tools are video controls. Table ii demonstrates all the buttons with their functionality. With these buttons you navigate through the video, move the video FoV and jump to the video frame you wish.

Video controls	
	
The sound wave displays the video sound, the black line shows the current moment (00:04 here). The see-through colours symbolise the characters of speakers in time.	
Frame backward	This button makes the video go backwards frame by frame (Alt+shift+left).
Fast backward	This button makes the video go backwards with a fast speed (F5).
Slow backward	This button makes the video go backwards with a slow speed (F6).

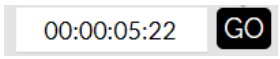
Toggle play/pause 	This button plays and pauses the video (F2). Note: The F2 shortcut is used to Play and to Pause the video alternatively, nonetheless the F3 shortcut has also been added to Pause the video only.
Stop 	This button makes the video stop and go to the beginning (F9).
Slow forward 	This button makes the video go forward with a slow speed (F7).
Fast forward 	This button makes the video go forward with a fast speed (F8).
Frame forward 	This button makes the video go forward frame by frame (Alt+shift+right).
Find subtitle by TC 	With this button, you can find the subtitle that contains the TC (Ctrl+Shift+F).
Jump backward 	This button helps the user to jump some frames backward. The number of the frames to be jumped is configurable in General Settings (see table 5) (F1).
Jump forward 	This button helps the user to jump some frames forward. The number of the frames to be jumped is configurablein General Settings (see table 5) (F4).
Move FoV left 	With this button you move the Field of View (FoV) to the left in the spherical video (Alt+left). You can also use the mouse and left button over the video and move to the left to do the same.
Move FoV up 	With this button you move the FoV up in the spherical video (Alt+up). You can also use the mouse and left button over the video and move up to do the same.
Move FoV down 	With this button you move the FoV to the down in the spherical video (Alt+down). You can also use the mouse and left button over the video and move to the bottom to do the same.
Move FoV right 	With this button you move the FoV to the right in the spherical video (Alt+right). You can also use the mouse and left button over the video and move to the right to do the same.
	Enter a specific time of the video, press GO and you are taken to that video frame.
Move FoV to "Speaker's location" 	By pressing this button the FoV moves to the angle where the speaker of current subtitle is set (Alt+F).

Table ii: Video controls

For each subtitle you have to enter the subtitle text in the text field and after finding the appropriate video frames the time codes (TC) must be entered: TCin by clicking on the TCin clock icon (Shift+Page up) and TCout by clicking on the TCout clock icon (Shift+Page down). Illustration ii displays all the information you need in this sense.

The below reading speed thermometer reflects the reading difficulty of the subtitle for the given duration. At first it is green, then if we are excessing the ideal reading difficulty it turns into red. The number (63 here) shows the remaining number of allowed characters for the subtitle.

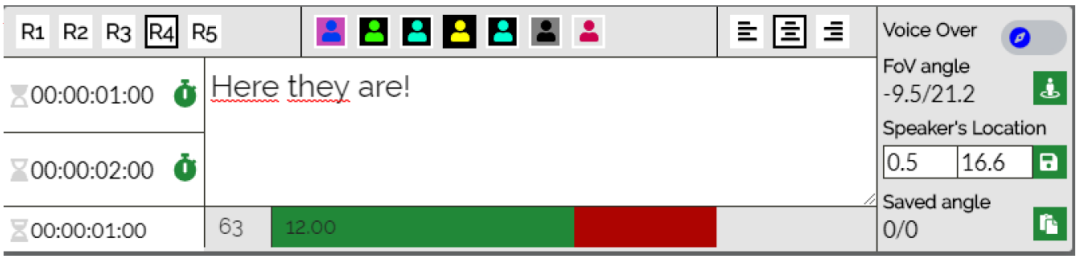


Illustration ii: Subtitle editing area

The next step, when you are sure about the text itself and its time codes, you may set region, character of the speaker and alignment for this subtitle (illustration iii). For the characters the users may use the shortcuts from Shift + F1 for Character C1 until Shift + F8 for Character C8.

Illustration iii: Subtitle editing area 2

As a result, a subtitle text and its styling are created successfully. However, this editor works in 360º media and matter of angles is important. So next step is setting proper angles for the subtitle which is done in the same area demonstrated in illustration ii.

Table iii explains the procedure of angle setting for a subtitle.

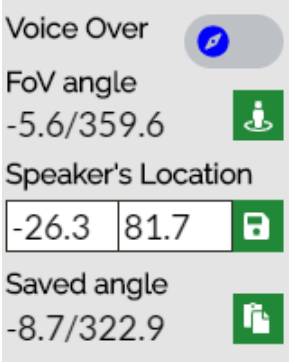
Subtitle editing – Angles	
By default, at first the video has the current angle as longitude: 0.00° and latitude: 0.00° Also, the **voice over** option can be marked when there is no speaker in the 360º scene.	
	**Voice over:** Only when there is only voice and no speaker in the 360º video frame.
	FoV angle: This is the current field of view (camera) angle and it changes while navigating through video ourselves and it corresponds to the video direction that we see (you can change the FoV angle using the navigation buttons in the video control area or moving the mouse with left button over the video). The green button next to it (Alt+Enter) sets the FoV angle to the "Speaker's location" of the subtitle (see next row).
	Speaker's location: This is set by the subtitler. It corresponds to the angle in the video where the speaker of this subtitle is located which is important in immersive media. It is important to know how to bind an angle to the subtitle. This is done solely by finding the desired angle by navigating and setting it to the subtitle. By pressing the green save button next to it, the "Speaker's location" value is copied to the "Saved angle" (Alt+C) for it to be used in other subtitles (see next row). The aim of this button is to copy this angle for later pasting it to other subtitles.
	Saved angle: This angle is kept in this register (see previous row) so it can be used in other subtitles. The button next to it pastes the "Saved angle" to the "Speaker's location" of the subtitle (Alt+V). The aim of this button is to use the angle from another subtitle that was copied previously.

Table iii: Subtitle editing – angle setting

You may want to repeat the procedure for the remaining subtitles and finish them all (text, styling, angles).

After finishing you may need some buttons in order to organise/edit/improve the sequence of the subtitles. Table iv shows you the buttons (and their shortcuts) you have available for this purpose.

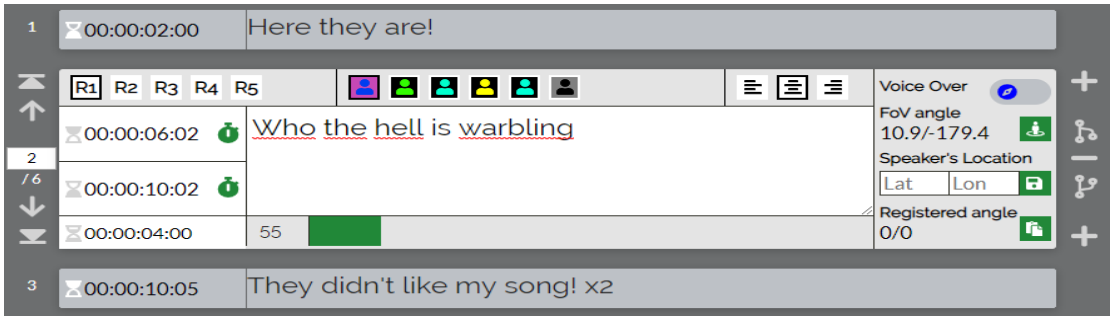
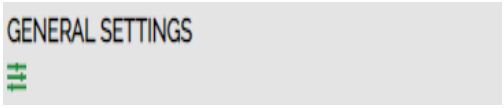
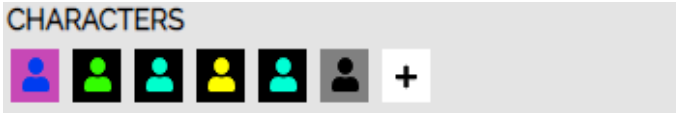
Subtitle editing – buttons	
	
First subtitle	This button takes you to the first subtitle (Alt + Page up).
Previous subtitle	This button takes you to the previous subtitle in relation to your current position (Page up).
Next subtitle	This button takes you to the next subtitle in relation to your current position (Page down).
Last subtitle	This button takes you to the last subtitle (Alt + Page down).
Jump to subtitle	This part takes you to a specific subtitle. But you need to enter the number of the subtitle. It also informs you of the number of subtitles.
Insert subtitle before	This button is useful for the insertion of a new subtitle before the existing subtitle (Ctrl+U).
Split subtitle on cursor position	This button splits the current subtitle text into two subtitles and gives us two separate subtitles (Ctrl+Insert).
Delete subtitle	This button deletes the selected subtitle (Ctrl+D).
Join subtitle with next	This button joins the current subtitle with the next one and creates a single subtitle containing both texts of the previous ones (Ctrl+Delete). The reading speed thermometer is modified accordingly but timings remain the same as the first previous subtitle.
Insert subtitle after	This button is useful for the insertion of a new subtitle after the existing subtitle (Ctrl+Shift+U).

Table iv: Subtitle editing and navigation buttons

E More options

The procedure in which we produce a subtitle is over, but we still have more options to work with.

Remember the regions, characters and buttons we introduced earlier. All these are customisable using “SETTINGS”. Table v shows the corresponding information.

Subtitle settings	
General settings	 This is related to the general setting. When clicked on the green icon the general settings dialogue appears. If any of the minimum numbers in the settings are not met, when trying to preview the final result (or check TCs) the user will get a warning or an error and needs to fix it before continuing.
Regions	 Regions correspond to the area in which the subtitle is going to appear on the FoV (Field of View) while being played. By clicking on the plus sign you create a new region and by clicking on any of the existing ones the region setting dialogue appears so you can edit the style of the region.
Characters	 Characters correspond to different speakers of the video. They are distinguished by their colour. By clicking on a character button the user is able to edit it with below settings available: • Font family, size, characteristics, etc.

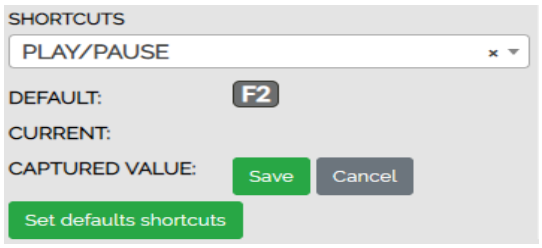
Shortcuts	 During the production of a subtitling, it is convenient to work only with keyboard instead of changing between keyboard and mouse buttons constantly. To allow this, most of the editor's functions have a shortcut key assigned by default. These default shortcuts are configurable and can be changed from here by the user. By pressing the "Set default shortcuts" the user can recover the default shortcuts. A list of the default shortcut buttons for the editor is presented as an annex at the end of this document. The shortcuts are also shown when hovering over the buttons in the Web ST Editor **WARNING: Some shortcuts are not advisable such as the ones that are used for editing the subtitle text and the ones that are used by the browser. There is a blacklist of key combinations that cannot be used as shortcuts, however the user must be aware when customizing specific shortcuts (for instance not to use the same key for two different shortcuts).**

Table v: Subtitle settings

Illustration iv, shows four different videos with different regions and styling. The dot corresponds to the current speaker's location. As you can see, it is the same colour as the character's colour.

Illustration iv: Video viewing at the centre

The last step after finishing editing is verification. For this, we have created preview modes as below.

Table vi gives the information you need in this sense in addition to general actions.

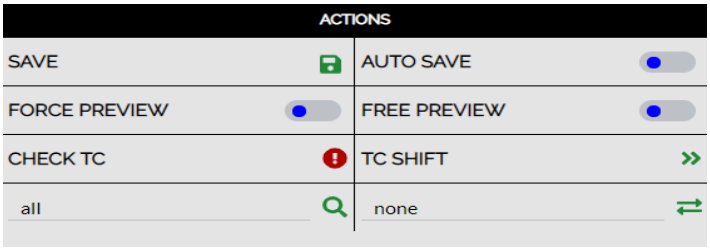
Subtitle actions	
	
Save	This button saves the subtitle work
Auto save	Saves the work automatically
Check TCs	This button checks if all of the timings and things related to them are correct. As mentioned in table ii, if the criteria set in the settings are not met, user will receive an error or warning accordingly (Ctrl+Q). Clicking on the exclamation mark, user can see the error/warning. If everything is ok, it turns green.
Forced preview	This mode is used for verification. This verification mode makes it easier for the subtitler as the video will change angle when needed during the playback of the video. Subtitles and angle are bound with the video. You cannot freely change angle at your wish, as the video itself takes you to the speaker's angle (the “Speaker’s location” of the subtitle) each time a subtitle is presented (F11). **Note**: when clicking this button, Check TCs is executed first and if there is any error the preview is not executed until the errors are fixed.
Free preview	This mode is used for verification. This verification mode is more real for the subtitler as if playing back the video with subtitles using HMD. Subtitles are bound to the video time code, but angle is not. It means that you can move angle (it is not fixed to the speaker) during the playback of the video. Any of these modes being on, the user cannot do editing anymore and needs to turn them off when editing is required (F12). **Note**: when clicking this button, Check TCs is executed first and if there is any error the preview is not executed until the errors are fixed.
TC shift	This shifts the time codes of a group of subtitles.
Find/Replace	This helps the user find specific words and replace them if needed.

Table vi: Actions

Also, at the upper side you have informative sections (Illustration v):

- On your left, general information of the production task.
- On your right, subtitle texts with their numbers and timecodes.

Illustration v: Informative sections

ANNEX III – DEFAULT SHORTCUTS

Functionality	Shortcut button
Toggle play/pause	F2
Pause	F3
Jump backward	F1
Jump forward	F4
Fast backward	F5
Slow backward	F6
Slow forward	F7
Fast forward	F8
Stop (Jump video to first frame)	F9
Frame backward	Alt + Shift + Left
Frame forward	Alt + Shift + Right
Move FoV left	Alt + Left
Move FoV right	Alt + Right
Move FoV up	Alt + Up
Move FoV down	Alt + Down
Move FoV to “Speaker’s location” of the subtitle	Alt + F
Previous subtitle	Page up
Next subtitle	Page down
First subtitle	Alt + Page up
Last subtitle	Alt + Page down
Find subtitle with video TC	Ctrl + Shift + F
Set TCin	Shift + Page up
Set TCout	Shift + Page down
Jump video to TCin frame	Ctrl + Alt + Page up
Jump video to TCout frame	Ctrl + Alt + Page down
Set character C1	Shift + F1
Set character C2	Shift + F2
Set character C3	Shift + F3
Set character C4	Shift + F4
Set character C5	Shift + F5
Set character C6	Shift + F6
Set character C7	Shift + F7
Set character C8	Shift + F8
Set FoV angle to “Speaker’s location” of the subtitle	Alt + Enter
Copy “Speaker’s location” of the subtitle to “Saved angle”	Alt + C
Paste “Saved angle” to “Speaker’s location” of the subtitle	Alt + V

Functionality	Shortcut button
Split subtitle on cursor point	Ctrl + Insert
Join subtitle with next	Ctrl + Delete
Delete subtitle	Ctrl + D
Insert subtitle before	Ctrl + U
Insert subtitle after	Ctrl + Shift + U
Check TCs	Ctrl + Q
Forced preview	F11
Free preview	F12

<END OF DOCUMENT>