Digital Cinema Stereoscopic Subtitle Workflow

Alaric Hamacher[1]
[1](Kwangwoon University, Republic of Korea)

Abstract: Subtitles in Digital Cinema are well established for monoscopic formats. They are often used, when dubbing is not available such as film festivals and international arthouse distribution. A wide range of authoring tools is available and subtitles can easily be added in Digital Cinema Packages. This simple language version creation method facilitates international distribution of movies. Mastering of stereoscopic movies and subtitles however, requires specialized tools for visualization and authoring. It is therefore common to see stereoscopic movies with lower quality subtitles or no subtitles as all when the language versions are burned into the movie. This makes the creation of international language versions much more costly than for monoscopic movies. Although Digital Cinema standards specify monoscopic and stereoscopic subtitles, they are rarely applied because of a lack of affordable authoring tools. The purpose of this paper is to elaborate a workflow for stereoscopic subtitle creation. The proposed method is based on a workflow using Davinci Resolve and takes advantage of its scripting features.

Keywords: Digital cinema, subtitles, stereoscopic, Davinci Resolve

I. INTRODUCTION

One of the big advantages of digital film distribution is dynamic combination of multiple assets such as audio, video and subtitles. This provides an efficient and cost-effective method to distribute multiple language versions as well as additional services using one single media. It has become common practice to find movies not only with different language versions, but also different audio mixes, audio description or subtitle and closed caption tracks. Many tools are available for mastering monoscopic and stereoscopic movies, and dedicated professional mastering stations such as Clipster or Pronto have the ability to create Digital Cinema Packages (DCP)[1] including stereoscopic subtitles. They are part of modern postproduction studio facilities which can edit and repackage the DCPs for theatrical releases of Hollywood movies.

However, for shorts, independent arthouse movies and many international film festivals these tools are often not available. They fall back on smaller software solutions or request filmmakers themselves to take care of subtitling. Smaller distributors and festivals manage to create different language versions and subtitles for monoscopic movies with existing tools which most modern editing programs provide. However, most of these software tools lack the ability to manage stereoscopic material and subtitles. As a result stereoscopic DCPs are created with monoscopic subtitles, which can cause a very uncomfortable viewing experience and in some cases ruin the filmmaker's artistic intentions.

1.1 Problem of stereoscopic subtitles

Efficient and affordable stereoscopic subtitle creation is important for Digital Cinema in order to avoid the most common problems in projection and distribution: Monoscopic movies are flat and they are usually projected on a plane screen in the theater. Subtitles are overlayed on the image and appear to the viewer on the screen. Stereoscopic movies, on the other hand are not seen on the surface of a plane screen, they are perceived as a volume. Similar to holograms, the image is transplane as the canvas on which the images are projected is not visible[2]. In three dimensional projections parts of the image can be in front of the screen, and other parts behind the screen. When monoscopic subtitles are overlayed with the stereoscopic image they are perceived exactly on the plane of the cinema screen. This does not cause particular problems when the movie action appears behind the screen. However, in scenes where parts of the action appear in front of the screen, the presence of monoscopic subtitles causes visual conflicts. Fig. 1a illustrates this case: objects and subtitles overlap and the subtitle can appear to be inside the objects, instead of floating in front of them as shown in Fig. 1b. The arrangement of space changes with the action. For this reason the creation of stereoscopic subtitles requires special attention and requires additional adjustments to avoid conflicting images and assure visual comfort[3].

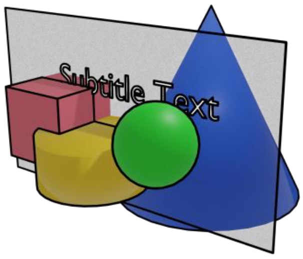


Fig 1a. monoscopic subtitle

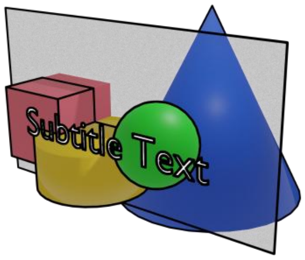


Fig 1b. stereoscopic subtitle

All efforts in mastering the movie and assuring its best possible projection to the audience can contribute significantly to the success of a movie.

1.2 Present Technical Overview

The number of tools for the creation of stereoscopic subtitles is unfortunately very limited. Many subtitle creation tools exist on a professional, semi-professional and amateur level for the creation of monoscopic subtitles. Table 1 shows examples of software tools for monoscopic subtitles:

Table 1. Video Subtitling Solutions overview

Software	**Platform**	**Stereoscopic**	**Main Purpose**
Clipster	hardware	yes	professional realtime mastering station
Aegisub	multiplatform	no	easy creation and editing of video subtitle files
Subtitle Edit	online	no	versatile multiformat subtitle editor
Youtube	online	no	simplified semi-automatic subtitle creator
Jubler	multiplatform	no	easy creation and editing of video subtitle files
Davinci Resolve	multiplatform	partially	professional postproduction and mastering

Most software solutions implement a basic video player combined with an editor input window which allows simulation of the subtitle experience. While some software programs exist online such as Subtitle Edit and Youtube, others are dedicated applications. In many cases the user interface displays audio waveforms of the sound tracks in order to help visually match display length of the subtitles with a voice track. Many tools go back to the time of DVD creation and are supported by a very active community. The software Davinci Resolve stands out of this list, because it is rather a complex editing and mastering tool which also has the feature to create subtitles. It also offers a built-in voice recognition and translation tool for automatic subtitle generation. Since pure subtitle creation tools work similar to transcript editors with timecode, their graphical performance is generally limited to text file output. As a postproduction tool, Davinci Resolve is able to perform complex image renderings and exports a variety of professional media file formats. Davinci Resolve comes in a free lite version and in a paid studio version. The studio version supports stereoscopic postproduction, however it does not have support for stereoscopic subtitles. The software runs on a variety of computer configurations which determine the performance of the video render process. To complete the overview of subtitle solutions the table includes Clipster [4], a dedicated hardware tool, with its own software for postproduction and the mastering of DCPs. Its strength lies in realtime capabilities and includes features such as encryption, decryption and realtime transcoding of different formats. This kind of equipment is usually in large postproduction facilities in charge of rollouts and large theatrical distribution.

Except Clipster all other solutions create subtitles in file formats aiming at consumer distribution such as streaming subtitles, DVD[5] or Bluray Disks[6]. Many file formats exist for subtitles, each of them serving different purposes and designed for different devices.

Table 2. Common Subtitle Formats

file suffix	format name	standard maintainer	year
.srt	SubRip	SubRip Software	2000
.sub	MicroDVD	Tiamat Software/Kotus	2000
.ssa	Sub Station Alpha	Sub Station Alpha	1996
.smi	Synchronised Accessible Media Interchange	Microsoft	1998
.vtt	Web Video Text Tracks	W3C	2010
.ttml/DFXP	Timed Text Markup Language	W3C	2010
.xml	DCinema ST 428-7:2014 DCDM Subtitles	SMPTE	2014

Many of the old formats where originally designed by proprietary applications such as .ssa for Karaoke files or .srt which became a standard due to its popularity. However, in order to support interoperability between applications and devices many efforts were made by standardization organizations. The W3C consortium focusses on standards related to media and internet applications [7], the SMPTE historically develops and maintains standards for the broadcast and cinema industry [8], [9].

The digital cinema standards specify SMPTE ST 428-7:2014 for subtitle creation inside a DCP. This standard defines how all kinds of text fonts and graphics have to be authored in order to be properly rendered at the moment of projection in the cinema. The details of a DCP are beyond the scope of this paper[10]. The standard in its latest revision from 2014 defines methods for stereoscopic subtitles. However, the feature is rarely used, and most applications mentioned above do not support it. The creation of stereoscopic subtitles for Digital Cinema relies on the implementation of the SMPTE ST 428-7:2014 standard. In order to implement such a file and to provide all the information it requires it is necessary to understand the construction of the file.

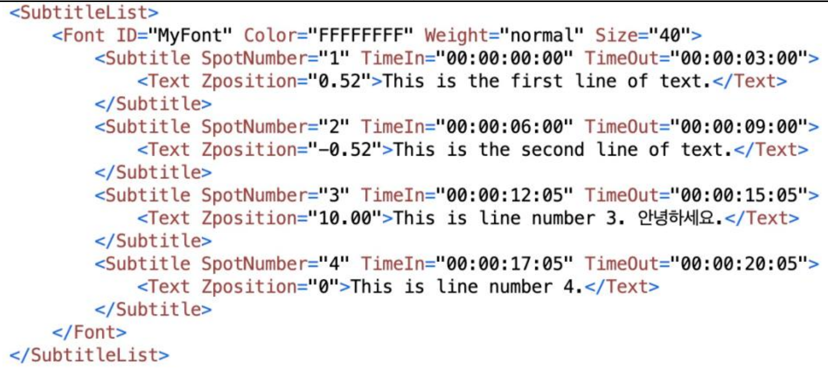
1.3 Anatomy of a stereoscopic subtitle file for Digital Cinema

The standard SMPTE ST 428-7:2014 is an extensive documentation containing all references and recommendations for the creation of a subtitle file that can be part of a digital cinema distribution master (DCDM) which contains all the elements that can be incorporated in a DCP [11]. The file itself is structured using the extended markup language (XML) as defined by the W3C [12]. To make this file format unique and fingerprint the latest version, the file uses a namespace in the shape of the uniform resource locator (URL):

https://www.smpte-ra.org/schemas/428-7/2014/DCST

This is the location where the XML schema definition (XSD) can be found on the internet. This is used to validate the proper construction of the document [13]. The subtitles have as a XML root the *Subtitle Reel* Element. Similar to the other assets of a DCDM the beginning of the subtitle document contains tags containing information about the technical parameters: *Id*, *Content Title Text*, *Annotation Text*, *Issue Date* and *Language* are used to uniquely identify the subtitles document. *Reel Number*, *Edit Rate*, *Time Code Rate*, *Start Time* contain the necessary information to ensure proper synchronization and playback along with the main picture track. *Display Type* specifies the purpose of the subtitles and can either be *Main Subtitle* or *Caption*. *Load Font* specifies a default font. The *Subtitle List* element contains the *Subtitle* elements.

The *Subtitle* element in its simplest from contains a *Time In* and *Time Out* attribute with the appropriate time code. Its child *Text* element contains the actual subtitle text. Alternatively the standard allows subtitles to be represented by images using a child *Image* element. In these child elements optional attributes are responsible for stereoscopic rendering of the subtitles. Optional attributes can further define additional formatting if desired. With the optional attribute *Z position* the depth positioning of the subtitle can be specified. The following code in Fig. 2 shows a simple example of an element *Subtitle List*:



```
<SubtitleList>
    <Font ID="MyFont" Color="FFFFFFFF" Weight="normal" Size="40">
        <Subtitle SpotNumber="1" TimeIn="00:00:00:00" TimeOut="00:00:03:00">
            <Text Zposition="0.52">This is the first line of text.</Text>
        </Subtitle>
        <Subtitle SpotNumber="2" TimeIn="00:00:06:00" TimeOut="00:00:09:00">
            <Text Zposition="-0.52">This is the second line of text.</Text>
        </Subtitle>
        <Subtitle SpotNumber="3" TimeIn="00:00:12:05" TimeOut="00:00:15:05">
            <Text Zposition="10.00">This is line number 3. 안녕하세요.</Text>
        </Subtitle>
        <Subtitle SpotNumber="4" TimeIn="00:00:17:05" TimeOut="00:00:20:05">
            <Text Zposition="0">This is line number 4.</Text>
        </Subtitle>
    </Font>
</SubtitleList>
```

Figure 2. code sample Subtitle List

Z position represents the parallax to render the specific subtitle. The value can be negative, which is supposed to display the subtitle in front of the cinema screen floating in cinema space as shown in the previous example of Fig. 1b. In case of a positive value the subtitle is positioned behind the screen. The value of zero renders with no parallax on the plane of the screen as if it was a monoscopic subtitle as displayed in Fig. 1a. The value of *Z position* is a decimal and represents a percentage of the width of the main picture. This method allows the positioning of static subtitles in space during the projection of a movie. However, in certain cases for technical, physiological or artistic reasons it can be necessary to change the position of the subtitles dynamically. In this case, the Image or Text element can contain an additional *Variable Z* attribute. This attribute contains the id of a *Load Variable Z* element that is a child of the *Subtitle* element. The text child of the *Load Variable Z* element contains a series of key and value pairs, representing a specific *Z position* value and a duration. With these values stereoscopic subtitles can be animated during the projection at specific time code positions.

The authoring of static or dynamic subtitles for Digital Cinema is the goal of this paper. It is the purpose of this research to help filmmakers and independent distributors in their efforts to achieve their success. The following section exposes a novel method and workflow. The following section shows by which methods this goal can be achieved.

II. METHODS

After the overview of the available formats and tools for subtitle creation and of the specification for stereoscopic subtitles for Digital Cinema, this section covers methods and workflows for creating a Digital Cinema subtitle file.

The following considerations guide the choice of methods and tools. It is not the purpose of creating a new tool from the scratch. As explained in the section above, many tools already exist with comfortable and established features for creating subtitles. Instead, the aim is to develop a workflow on top of an existing tool, to take advantage of all its existing features and only add the functionality needed to create stereoscopic subtitles. It can also be an advantage to work with an existing tool that is familiar to the industry. Among the previously discussed software, Davinci Resolve Studio seems to be the most advantageous platform to implement a stereoscopic subtitling solution. Mainly for two reasons:

First, the software provides already many features for handling stereoscopic media: media management, editing, color correction and rendering are designed for stereoscopic workflows. Additionally, the software not only handles monoscopic and stereoscopic video output on external devices such as 3D televisions or cinema projectors, but also connects with various input devices. Being able to observe stereoscopic adjustments on a proper display while using rotation knobs to fine tune the stereoscopic parallax settings facilitate the authoring process.

Second, Davinci Resolve offers a variety of methods for programmers to interact with the software. Application programming interfaces (API) [14] exist for common languages such as Python, Lua or Javascript. Additional functionality can be integrated as Workflow Plugins.

These considerations and features favor Davinci Resolve Studio as a development platform for the objective of this research.

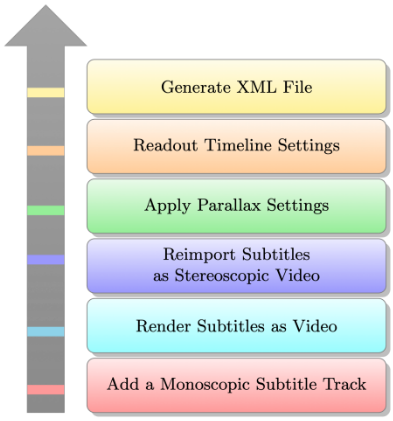

Generate XML File
Readout Timeline Settings
Apply Parallax Settings
Reimport Subtitles as Stereoscopic Video
Render Subtitles as Video
Add a Monoscopic Subtitle Track

Figure 3. Stereoscopic Subtitle Workflow

Stereoscopic subtitles in Davinci Resolve can be created with the workflow describes in Fig. 3. The starting point requires a timeline with a stereoscopic video track. This could be a master consisting of a left and and a right eye video. The first step is to input the subtitle text.

II.1 Add a Monoscopic Subtitle Track

Adding a monoscopic subtitle track on the timeline offers all possibilities for writing, editing and retiming of the subtitles. The software also offers a function to automatically create subtitles using voice analysis from selected audio tracks. Additionally, subtitles created with other software can be imported using for example the .srt file format [15]. This task is common to subtitle authoring in monoscopic projects.

II.2 Render Subtitles as Video

Since subtitles in Davinci Resolve are only monoscopic, this steps transforms the subtitles to a video file which can be displayed in stereoscopic mode. For this purpose, the timeline is rendered only with the subtitles enabled as a monoscopic movie on a transparent background with alpha channel. Since the text does not occupy a lot of the image, the rendering process is fast and the resulting file size relatively small.

II.3 Reimport Subtitles as Stereoscopic Video

In this step, the previously generated subtitle video file is reimported as an asset in Davinci Resolve Studio. By changing the file attribute in the context menu, it can be changed from monoscopic to stereoscopic before it can be added on a new video track above the stereoscopic movie on the timeline. However, since the subtitle track has become a single clip of the duration of the whole movie, the individual subtitles are not visible anymore as time segments. In order to visualize the individual subtitle cuts it is necessary to re-conform the subtitle movie using the in- and out- points of the subtitle track as an editing decision list (EDL). This EDL can easily be generated using the Workflow Plugin. Fig. 4 shows the timeline with the monoscopic subtitle track on top and the re-conformed rendered subtitles which are now visible and editable as stereoscopic video.

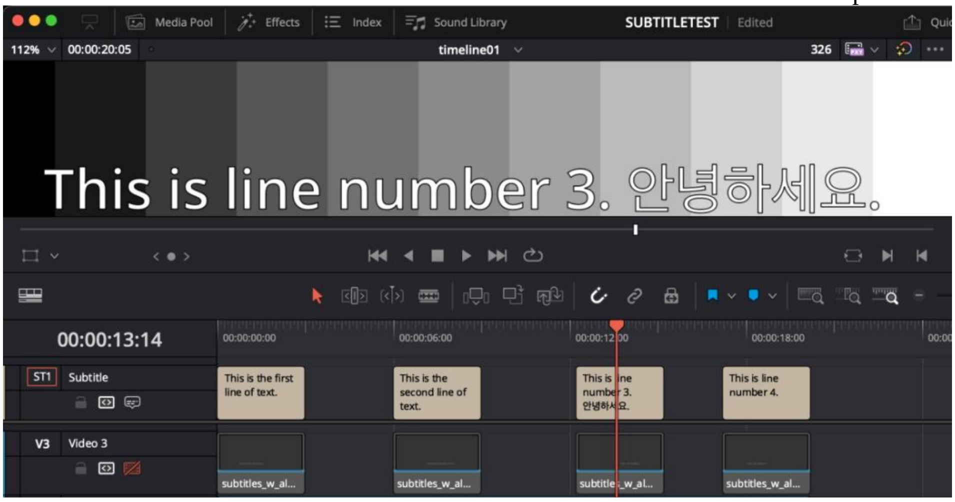


Figure 4. subtitles as stereoscopic video in Davinci Resolve Studio

II.4 Apply Parallax Settings

Once the monoscopic subtitle track is deactivated, identical subtitles appear on top of the stereoscopic video. As Davinci Resolve Studio performs stereoscopic compositing, the result can be observed using for example anaglyph mode for red and cyan glasses or with the appropriate hardware the video can be output to a 3D television or an attached 3D projector. Using the 3D tab of the color page, each subtitle can individually be repositioned in space. The convergence parameter adapts the parallax. If necessary keyframes can be used for animation.

II.5 Readout the Timeline Settings

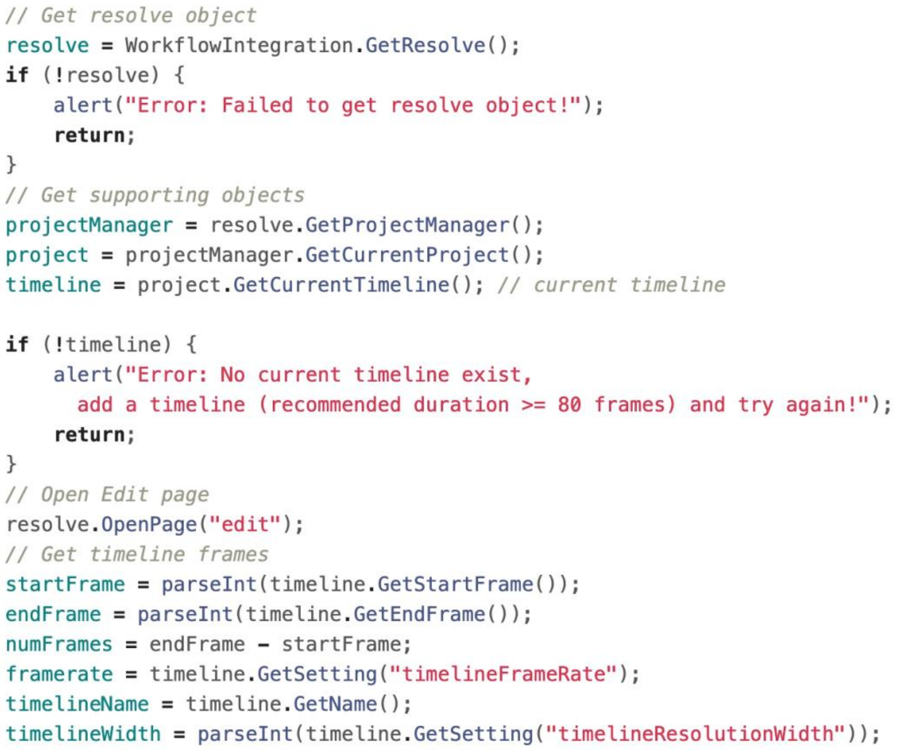
After all subtitles are moved to the desired positions a stereoscopic subtitle file can be created. The objective of this process is to use the text information from the subtitle track and inject the data about the stereoscopic parallax settings created in the previous step. Unfortunately it is not possible to export the timeline information in a humanly readable format. Davinci Resolve files are written in a proprietary binary format which makes it difficult to access the desired information. However, it is possible to retrieve the stereoscopic parallax settings from each clip on the timeline using the API. This task can also be achieved with a Workflow Plugin.

II.6 Generate XML File

In a final step, the subtitle text, the positions on the timeline together with the individual stereoscopic parallax values have to be assembled following the specifications of the SMPTE ST 428-7:2014 standard. The subtitle parallax values have to be transferred to the *ZPosition* attribute. In case of animated stereoscopic subtitles the individual keyframes have to be converted to match the *VariableZ* format. As Davinci Resolve stores the parallax in pixels the timeline image width has to be taken into account to create the required percentage value of the main picture in the DCP. The simple transformation can be calculated according to equation 1.

$$ZPos\,\% = 100\frac{convergence}{imagewidth} \qquad (1)$$

The calculations and formatting can be achieved using the Davinci Resolve API. The code shown in fig. 5 demonstrates how to access values and settings from the timeline.



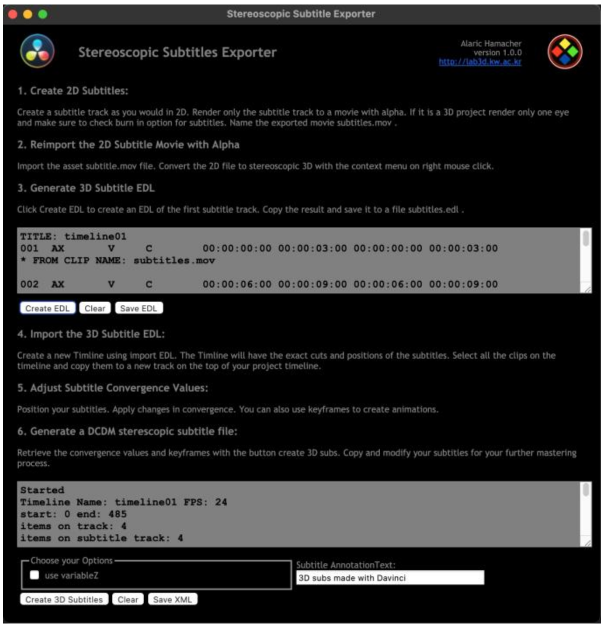
```
// Get resolve object
resolve = WorkflowIntegration.GetResolve();
if (!resolve) {
    alert("Error: Failed to get resolve object!");
    return;
}
// Get supporting objects
projectManager = resolve.GetProjectManager();
project = projectManager.GetCurrentProject();
timeline = project.GetCurrentTimeline(); // current timeline

if (!timeline) {
    alert("Error: No current timeline exist,
      add a timeline (recommended duration >= 80 frames) and try again!");
    return;
}
// Open Edit page
resolve.OpenPage("edit");
// Get timeline frames
startFrame = parseInt(timeline.GetStartFrame());
endFrame = parseInt(timeline.GetEndFrame());
numFrames = endFrame - startFrame;
framerate = timeline.GetSetting("timelineFrameRate");
timelineName = timeline.GetName();
timelineWidth = parseInt(timeline.GetSetting("timelineResolutionWidth"));
```

Figure 5. Code example to read out timeline information using API

A function iterating over all elements of the monoscopic subtitle track can read out one by one the text and format information together with the stereoscopic parallax values of the stereoscopic video subtitle track. Using the document object model (DOM) [16] the XML document can be created by adding elements, nodes and attributes according to the specification. Once the subtitle document is created in memory it can be saved to the user's file system as XML file.

III. RESULTS

Using the tools and methods described in the previous sections, a Workflow Plugin for Davinci Resolve is developed. The Plugin works like an independent application that can be started from a menu inside the application. The Plugin uses the Electron framework [17] to display a custom user interface with interactions using hypertext markup language (HTML) cascading style sheets (CSS) and Javascript. Fig. 6 shows a screenshot of the Workflow Plugin in action.



Stereoscopic Subtitle Exporter
Stereoscopic Subtitles Exporter
Alaric Hamacher
version 1.0.0
http://lab3d.kw.ac.kr

1. Create 2D Subtitles:

Create a subtitle track as you would in 2D. Render only the subtitle track to a movie with alpha. If it is a 3D project render only one eye and make sure to check burn in option for subtitles. Name the exported movie subtitles.mov .

2. Reimport the 2D Subtitle Movie with Alpha

Import the asset subtitle.mov file. Convert the 2D file to stereoscopic 3D with the context menu on right mouse click.

3. Generate 3D Subtitle EDL

Click Create EDL to create an EDL of the first subtitle track. Copy the result and save it to a file subtitles.edl .

```
TITLE: timeline01
001  AX       V     C        00:00:00:00 00:00:03:00 00:00:00:00 00:00:03:00
* FROM CLIP NAME: subtitles.mov

002  AX       V     C        00:00:06:00 00:00:09:00 00:00:06:00 00:00:09:00
```

4. Import the 3D Subtitle EDL:

Create a new Timline using import EDL. The Timline will have the exact cuts and positions of the subtitles. Select all the clips on the timeline and copy them to a new track on the top of your project timeline.

5. Adjust Subtitle Convergence Values:

Position your subtitles. Apply changes in convergence. You can also use keyframes to create animations.

6. Generate a DCDM sterescopic subtitle file:

Retrieve the convergence values and keyframes with the button create 3D subs. Copy and modify your subtitles for your further mastering process.

```
Started
Timeline Name: timeline01 FPS: 24
start: 0 end: 485
items on track: 4
items on subtitle track: 4
```

Choose your Options
☐ use variableZ

Subtitle AnnotationText: 3D subs made with Davinci

Figure 6. Stereoscopic Subtitle Workflow Plugin

The user interface displays the workflow instructions: The upper part of the window displays the automatically created EDL to re-conform the subtitle video file described in step three of the proposed workflow. The lower part displays the output activity and contains an additional input field to provide a custom value to the element Annotation Text. A checkbox enables the creation of variable Z elements. The save button prompts for a filename and exports the XML file to the users file system. This file is the stereoscopic subtitle file for Digital Cinema. The Workflow Plugin can be installed on all installations of Davinci Resolve Studio. This brings the possibility to create stereoscopic subtitles to most users already using Resolve for stereoscopic postproduction.

IV. FUTURE RESEARCH

This paper mainly focuses on the generation of the XML subtitle file. How this file can be further used in the DCP packaging process, and how the stereoscopic subtitles are actually displayed in the 3D cinema can be topics for further research.

V. CONCLUSION

The present paper has exposed the constraints and necessity for appropriate tools for the creation of stereoscopic subtitles. At present, such tools are not available for independent filmmakers or festivals wishing to create additional language versions for stereoscopic movies. As the specifications for digital cinema require a special subtitle format, the generally available subtitle authoring tools cannot be used. This paper has therefore proposed a workflow for the stereoscopic capable and widely used software Davinci Resolve Studio. Using the Davinci Resolve API this paper has successfully demonstrated how a Workflow Automation Plugin can be created and used to add the feature of stereoscopic subtitling to the application. As the success of stereoscopic movies relies on distribution, the ability to create additional language versions with stereoscopic subtitles is crucial. This research aims to support this process and to broaden distribution of stereoscopic movies.

VI. CONFLICTS OF INTEREST

The authors declare no conflict of interest.

VII. ACKNOWLEDGEMENTS

This research was conducted with a Research Grant from Kwangwoon University in 2023. This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2020S1A5B8102066).

REFERENCES

[1]. Z.-S. Wang *et al.*, “A Digital Cinema Playback system compliant with the DCI specification,” in *Picture Coding Symposium, 2009. PCS 2009*, IEEE, 2009, pp. 1–4. Accessed: Mar. 22, 2016. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5167454

[2]. J. Schröter, *3D: History, Theory and Aesthetics of the Transplane Image*. Bloomsbury Publishing USA, 2014.

[3]. G. Yue, C. Hou, J. Lei, Y. Fang, and W. Lin, “Optimal region selection for stereoscopic video subtitle insertion,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 11, pp. 3141–3153, 2017, Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8010450/

[4]. N. N., “CLIPSTER Software Guide 5.1.” Rohde & Schwarz DVS GmbH, 2014.

[5]. M. Ely and P. De Lancie, *DVD production: a practical resource for DVD publishers*. Routledge, 2012. Accessed: Feb. 08, 2024. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.4324/9780080504803/dvd-production-mark-ely-phil-de-lancie

[6]. M. D. Smith, “Ultra HD Blu-ray Format Video Characteristics,” *SMPTE Motion Imaging Journal*, vol. 125, no. 5, pp. 14–27, 2016, Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7514362/

[7]. A. K. Al-Harbi, “Understanding the Regulatory Context of the Internet.,” *Journal of Law/Magallat al-Huquq*, vol. 41, no. 2, 2017, Accessed: Feb. 08, 2024. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=10296069&AN=124086025&h=s3hfzxH26eE3VoBpQSGNT%2Bc63ucoyYAOysCwpQviXBBiSk%2FAAsYhm8q9ii6UIOrNLKaehk%2BDBM49TlC3DD11Dg%3D%3D&crl=c

[8]. C. E. Anderson, “Organization of SMPTE Engineering Program: A Report of the SMPTE Standards Committee,” *Journal of the SMPTE*, vol. 83, no. 7, pp. 600–604, 1974, Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7233648/

[9]. R. M. Rast, “SMPTE Technology Committee on Digital Cinema—DC28: A Status Report,” *SMPTE journal*, vol. 110, no. 2, pp. 78–84, 2001, Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7257385/

[10]. J. Watts, M. Scoll, S. LLamb, and A. Lampard, “D-Cinema Packaging—SMPTE DCP Bv2. 1 Application Profile,” 2020, Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9161348/

[11]. A. Samčović, “CREATING DCP WORKFLOW FOR DIGITAL CINEMA SYSTEM.,” *Carpathian Journal of Electrical Engineering*, vol. 14, no. 1, 2020, Accessed: Feb. 08, 2024. [Online]. Available: http://cee.cunbm.utcluj.ro/wp-content/uploads/1843-7583-14-1.pdf#page=19

[12]. T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible markup language (XML).,” *World Wide Web Journal*, vol. 2, no. 4, pp. 27–66, 1997.

[13]. G. J. Bex, F. Neven, and J. Van den Bussche, “DTDs versus XML Schema: A practical study,” in *Proceedings of the 7th international workshop on the web and databases: colocated with ACM SIGMOD/PODS 2004*, 2004, pp. 79–84. Accessed: Feb. 08, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1017074.1017095

[14]. J. Bloch, “How to design a good API and why it matters,” in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, 2006, pp. 506–507. Accessed: Feb. 08, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1176617.1176622

[15]. J. Tiedemann, “Building a multilingual parallel subtitle corpus,” *Proc. CLIN*, vol. 14, 2007, Accessed: Feb. 08, 2024. [Online]. Available:

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0d0eb34ab56f7b48b6d611b5d0767bc59ba8b9fc

[16]. J. Marini, *Document object model*. McGraw-Hill, Inc., 2002. Accessed: Feb. 08, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/863219

[17]. M. Jasim, *Building Cross-Platform Desktop Applications with Electron*. Packt Publishing Ltd, 2017.