Playback and Review Standards

Abstract

Many reviews, particularly between vendor and production company end up transcoding the source media to a movie format for review. Even after the movie file generation the production company may rename the file to match their own conventions (if they dont transcode it again). Adding metadata to these transcoded files (and provide recommendations on how to carry any metadata through additional transcoding sessions) could help then translate any notes, or annotations back to the source media.

Secondarily it would be good to make recommendations for transcoding for review. During reviews, its common to single frame forward and backwards, color fidelity is often critical to the review process, so we need to know that the resulting media isnt making any color adjustments (this was an issue with h264 in quicktime for a while).

At this point its not clear if this is just a standards process, or if code will need to be developed.

Powerpoint PDF output presentation:

Improving Vendor Reviews.pdf

Metadata Standards

Media definition

These fields are for transcoded media, for original frame sequences it shouldn't be necessary. Metadata that should exist in the media file could include:

First Frame , Last Frame	int	Encoding to movie files typically loses the start frame, making it a pain to identify which frame you are looking at. We could look at doing this with timecode, but sometimes you want both timecode and a frame number.

Source filename	string	Something to track where the encoded media came from.
Source ID	string	Unique ID from vendor creating content. - This could be using: https://proto.school/content-addressing/04
Source frame rate	float	If you are reviewing a proxy, but still want to remap back to the source frame, knowing the source frame rate is required (DO WE NEED THIS AND LAST FRAME?) – useful for high-frame rate media, e.g. 120 fps - (MIGHT MAKE SENSE AS A STRING TO HANDLE 59.94 better?)
image active area	xMin, yMin, xMax, yMax	The bounding box of the picture location within the image. This is used in cases where the image is a re-processed version of the source frame, e..g. where a 2.35 aspect ratio picture has been padded to HD (perhaps timecode is burnt in, etc), this would allow any annotations to be always defined relative to the source frames, so would be able to be correctly overlayed on top.
Watermarking?	String	Document what sort of watermarking has been applied? - invisible, burnin?
Slate Length	Int	Duration of slate length (0 if no slate).
Display Type	Enum	Stereo left/right Stereo top/bottom Long/Lat VR mono Long/Lat VR Stereo top/bottom NOTE: This should be based on existing standards, e.g. https://github.com/google/spatial-media/tree/master/spatialmedia
Color Space	string	Many file-formats do already have options for color spaces, but certainly for internal reviews facilities may decide to encode to a non-standard color space. For media that is crossing facilities we should stick to known embedded colorspaces, and allow existing tools to remap where necessary.

Screen coordinate system could be based on: https://github.com/desruie/OpenTimelineIO/blob/autodesk_desruie_spatial_doc/docs/tutorials/spatial-coordinates.md

Review definition

Metadata that should exist in the file.

Source ID	string	A unique ID for the company generating the media that can be used to get back to the original media. The main use of this is if the filename changes as it goes through different company pipelines. This may only be for reviewed media, rather than all media, and ideally its something reasonably compact and human readible, for example <SHOWCODE>-<REVIEWID> – spy-1234 – where reviewid is an incrementing ID per show. For this document we dont care about its structure, only that it exists.
Source Entity	string	Identifies the shot, asset or entity. Potentially useful as a burn-in.
Source sub entity	string	This would be the way to identify the media within the source entity without versions. e.g. lets say I have a filepath /shows/spy01/bat001/pix/rnd/precomp_v001/precomp_v001.0001.exr spy01 = SHOW bat001 = SHOT = Source Entity precomp = Source subentity This would be used by an asset management system to group versions, without having to guess what the versioning system is.
source sub entity version	float	Version ID for sub-entity.

Task	string	Taskname if known at creation -
date authored	string	The latest date of the original authored content. This would be carried through any transcoding, so we dont end up with the transcoded timestamps.

Media Review Info

For doing external reviews (vendor passing media to production company) you may have additional metadata that needs to be passed with the media for review, this would typically be in an excel file (See VES Delivery Specification) with the following columns:

Date Submitted	date string	repeated on each line for cases where the resulting excel sheet is merged.
Vendor	string	Vendor name, repeated on each line for cases where the resulting excel sheet is merged.
Filename	string	The filename that is being shipped to be reviewed.
Source ID	string	Used to map the following fields to the actual media.
review task name	string	The reviewing company may have their own task names, which could be "comp", "anim"
Review for	string	Notes on why the media is being reviewed, e.g. For Final, For Feedback, WIP.
Notes	string	Notes for the reviewer, so they know what they should be commenting on.

NOTE, some or all of these fields may also make sense embedded.

Implementation

Use existing metadata values where possible, and fall back on XMP where not.

https://www.adobe.com/devnet/xmp.html

or

https://wiki.multimedia.cx/index.php/FFmpeg_Metadata

https://python-xmp-toolkit.readthedocs.io/en/latest/

XMP data can be read by ffprobe, e.g.: ffprobe -v quiet -print_format json -show_format -export_xmp 1 -show_streams "[0000-0119].mov"

Also, exiftool appears to have support for reading and writing xmp data: https://exiftool.org/forum/index.php?topic=8745.0 , see: https://exiftool.org/TagNames/QuickTime.html

Annotations and Notes

The other area that common specifications could be defined is how annotations and notes could be sent back to the vendor, in a format that is ready for ingest into the tracking system.

https://openreviewio-standard-definition.readthedocs.io/fr/latest/README.html

Notes

Notes are often going to be directly ingested into databases, but there will be cases where you want to send them from vendor to vendor. For this we may want to define a neutral excel format that is easy to read for non databases.

Date Reviewed	String	
Reviewer Names	String	Who was doing the reviewing.
Review Location	String	Where it was.

ReviewID	String	A unique ID that can be used to map annotations to a review. Ideally this is something human-readable, e.g. YYYYMMDDHHMM-<Location> but from the file format point of view, its simply a string.
Source ID	String	Reference back to the media source.
Source Entity	String	i.e. the shot
Sub-source entity	String	i.e. the sub-source
Status	String	Approved/Not Approved/CBB
Notes	String	

Logically, the first three columns are really the header and in most cases are simply repeated, but it does simplify everything into a single table.

Annotations

Annotations would need to be in a more computer readable format such as XML or JSON. e.g.:

```
<review reviewid="{REVIEWID}" datereviewed="{DATEREVIEW}" reviewby="{REVIEWBY}" reviewlocation="{REVIEWLOCATION}">
    <media sourceid="{SOURCEID}">
       <notes>
            {NOTES}
       </notes>
       <annotation frame="{FRAMENUMBER}">
          <line thickness="{SIZE}" style="{LINESTYLE}" color="{COLOR}">
               <coord x="{X}" y="{Y}"/>
               <coord x="{X}" y="{Y}"/>
               <coord x="{X}" y="{Y}"/>
          </line>
       <brush style="" color="<COLOR>">
          <coord x="{X}" y="{Y}" thickness="{SIZE}" opacity="{OPACITY}"/>
          <coord x="{X}" y="{Y}" thickness="{SIZE}" opacity="{OPACITY}"/>
          <coord x="{X}" y="{Y}" thickness="{SIZE}" opacity="{OPACITY}"/>
        </brush>
        <text x="{X}" y="{Y}" fontsize="{SIZE}" label="{TEXT TO DISPLAY}" />
        <colorcorrect x="{X}" y="{Y}" size="{SIZE}" area="{AREATYPE}" asccc="{ASC COLOR CORRECT}" />
   </media>
</review>
```

Encoding standards

FFMPEG

Can we define ffmpeg standards for playback, including color space conversion. The default color space for ffmpeg is bt601, which is not typically what we are all using any more, but there are a number of conversion options that we should be considering. Its fairly typical to see the -vf "colormatrix=bt601:bt709" flag in conversions. This is an 8-bit color space conversion, there is apparently a better option, see: https://trac.ffmpeg.org/wiki/colorspace

It would be great to have well documented ffmpeg encoding flags that satisfy:

- Color fidelity.
- Ability to reasonably single step forwards and backwards.
- File size

Shotgun transcoding: https://help.autodesk.com/view/SGSUB/ENU/?guid=SG_Administrator_ar_data_management_ar_diy_transcoding_html

Syncsketch points to this: https://cms.eas.ualberta.ca/dif/case-studies-tutorials/ffmpeg-convert-image-sequence-into-movie/

Reference "Dailies script" https://github.com/jedypod/generate-dailies

EXR

RV has notes on performance here - https://support.shotgunsoftware.com/hc/en-us/articles/219042268-Optimizing-RV-Playback-Performance

Are there recommendations on versions of EXR that are better for review, particularly externally?

References

https://s3.amazonaws.com/software.tagthatphoto.com/docs/mwg_guidance.pdf - Guidelines for handline metadata

https://www.hackerfactor.com/blog/index.php?/archives/552-Deep-Dive.html - reference for XMP History.

http://info.signiant.com/rs/134-QHZ-485/images/Signiant_EG_Metadata_Everywhere_CloudSpeX.pdf