Comparing Four 360-Degree Cameras for Spatial Video Recording and Analysis

Jinyue Guo (郭锦岳) (jinyueg@uio.no) [1,2], **Maham Riaz** (mahamr@uio.no) [1,2], and
Alexander R. Jensenius (alexanje@uio.no) [1,2]

[1] *RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion*, **University of Oslo**, Oslo, Norway
[2] *Department of Musicology*, **University of Oslo**, Oslo, Norway

ABSTRACT

This paper reports on a desktop investigation and a lab experiment comparing the video recording capabilities of four commercially available 360-degree cameras: GoPro MAX, Insta360 X3, Garmin VIRB 360, and Ricoh Theta S. The four cameras all use different recording formats and settings and have varying video quality and software support. This makes it difficult to conduct analyses and compare between devices. We have implemented new functions in the Musical Gestures Toolbox (MGT) for reading and merging files from the different platforms. Using the capabilities of FFmpeg, we have also made a new function for converting between different 360-degree video projections and formats. This allows (music) researchers to exploit 360-degree video recordings using regular video-based analysis pipelines.

1. INTRODUCTION

So-called 360-degree video—often also called spherical video—is popular in immersive content creation but is still used relatively little in research. We are exploring 360-degree video recordings for various sound and music projects where sound sources are moving or distributed across an environment. This includes music performances in concert halls, where a 360-degree camera can capture both musicians and the audience. It also includes studies where dancers move around a space. Our recent research is focused on the audiovisual qualities of in-door environments, where capturing the entire scene is essential. Regular video recordings only capture a small portion of space, while 360-degree cameras can capture the entire scene if positioned well.

Even though 360-degree cameras have been on the market for some years, they are still a niche product, and the manufacturers have not agreed on standard settings and formats. Few software solutions are available for editing, and to our knowledge, no analysis tools are available. In sum, while there is a huge potential for using 360-degree video in research, the existing commercial 360 cameras and their software have insufficient support for ad-

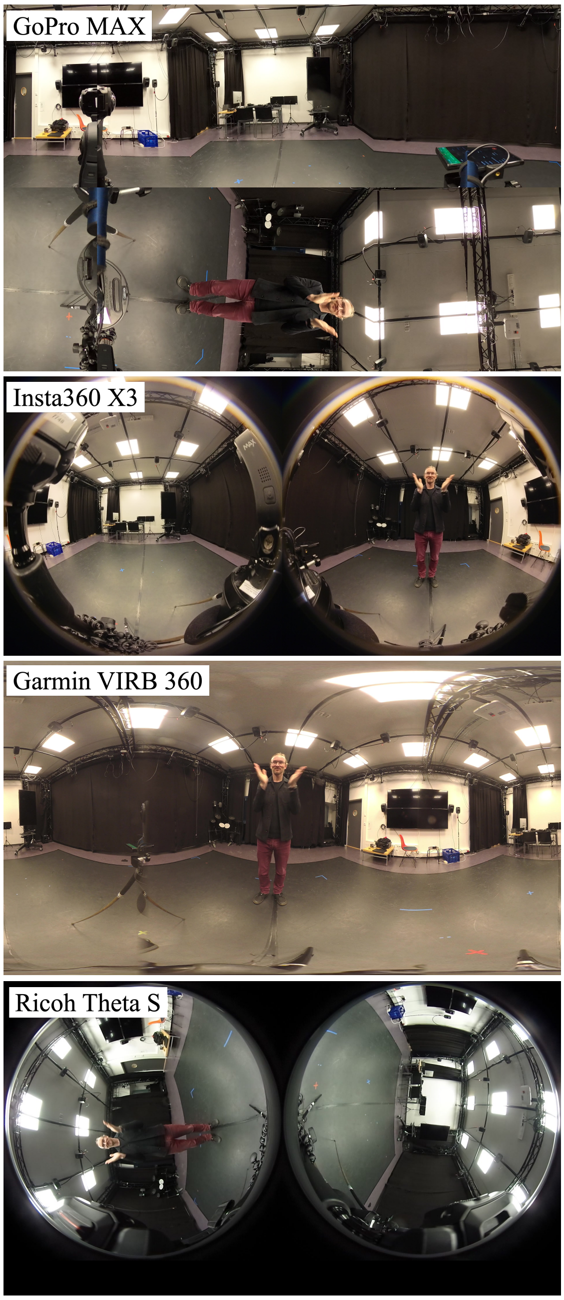

Figure 1. Original projections of the four cameras. GoPro MAX: modified equi-angle cubemap; Insta360 X3: dual-fisheye in two separated files; Garmin VIRB 360: equirectangular; Ricoh Theta S: dual-fisheye in one file.

vanced users and researchers. In this paper, we review four commercially available 360-degree cameras, describe their recording specifications and formats, and present how we implemented functions in the Musical Gestures Toolbox [1] for analyzing the recorded files.

2. ABOUT 360-DEGREE VIDEO

360-degree video is a visual media type that provides an immersive viewing experience. Such videos refer to omnidirectional vision, which entails a full spherical view, including 360 degrees in both horizontal and vertical directions.

Immersive recording and production techniques are becoming popular in the gaming and film industries. In broadcasting, curved monitors and television screens lead the way with an extended field of view. Head-mounted displays (HMDs), like virtual and mixed reality headsets, feature head tracking, enabling dynamic perspective adjustment with natural head motion, thus offering viewers a full 360-degree experience.

On the production end, new techniques have revolutionized set design, with the virtual reconstruction of entire filming locations, using circular screens spanning up to 270 degrees [1]. A 360-degree video is a cost-effective option for immersive recordings. This advanced visual medium has become popular with filmmakers, videographers, athletes, and tech enthusiasts looking for user-friendly recording equipment.

360-degree video recordings can manifest in several representations. Most commercial 360-degree cameras consist of two or more fish-eye cameras and contain internal stitching and processing algorithms that save the videos in a processed format. The saved files differ hugely between models, but most manufacturers provide editing software that allows users to perform post-processing, including trimming, stabilization, denoising, and projection conversion. This makes it possible to render 2D video files for "normal" video editing.

The processing pipeline of 360-degree videos differs from traditional videos mostly due to its spherical property [2]. It contains four steps [3]:

1. Stitching: The video is often captured using multiple cameras, resulting in multiple image views that must be combined ("stitched").

2. Projection: The 360-degree video needs to be projected to a surface and then encoded with compression algorithms for storage.

3. Encoding: The video must then be encoded and transmitted to the encoded media using traditional or customized protocols.

4. Rendering: Finally, the file is decoded and rendered on the viewer's device.

Unlike traditional video processing, the steps of a 360-degree video pipeline cannot be separated from each other [2]. The stitching algorithm will affect the projection, the efficiency of compression methods will change with different projections, and the transmission is adaptive to the viewer's device and viewing habits.

Projection is an important step in the pipeline since all the following steps need to be adjusted according to the projection [2]. For example, the *equirectangular projection* (ERP) [4] is widely adopted and equally projects the spherical video from its center to a flat surface. The projection largely expands the pole areas that are not likely viewed by the viewers, therefore the compression algorithm can utilize a lower bitrate at the upper and lower edge of the equirectangular projection video. The transmission algorithm can also prioritize the middle section around the equator since it is more likely to be viewed.

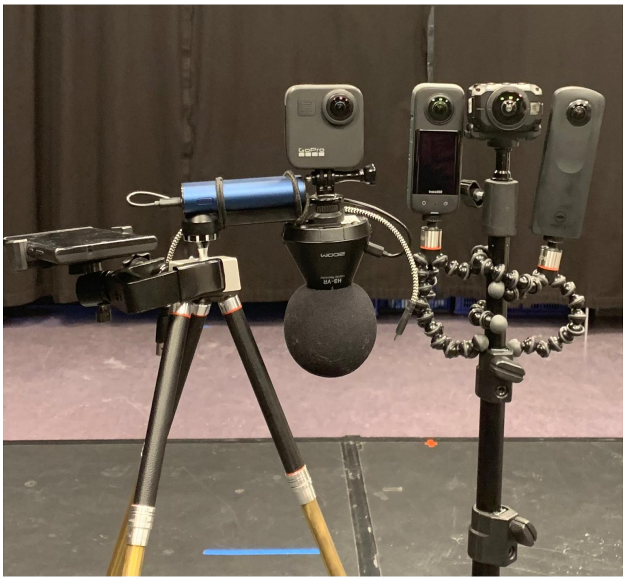

Figure 2. The experimental setup: four 360-degree cameras (left to right: GoPro MAX, Insta360 X3, Garmin VIRB 360 and Ricoh Theta S). A Samsung Galaxy S23 Ulta mobile phone captured sound and light levels.

3. FOUR 360-DEGREE CAMERAS

We have been using 360-degree cameras in our lab for around a decade and decided to conduct a comparative study between four commercially available devices. Four 360-degree cameras were placed next to each other at the same height ($\sim$1 meter), and a mobile phone recorded the "ground truth" light intensity in the lab (Figure 2).

In the following, we will only investigate the video part of the recorded files; the audio part is the subject of another paper [5]. The recordings were done with the best possible settings in each camera. All of them typically allow for reducing the pixel dimensions or frame rates, but we wanted to check the maximum capabilities of each camera.

Each camera has its own software solution for handling the recordings. We investigated the original files saved by each camera, as well as the export options in their corresponding software.

[1] `https://github.com/fourMs/MGT-python/`

Camera	File type	Projection	Codec	Colorspace	Resolution	FPS	Bitrate (kb/s)
GoPro MAX	.360	Equi-Angular Cubemap	H.265	yuvj420p	4096 x (2688)	25	30,002 (x2)
	.LRV	Dual-fisheye	H.264	yuvj420p	1408 x 704	25	2499
Insta360 X3	.INSV	Fisheye (x2)	H.264	yuvj420p	(5760) x 2880	29.97	60,495 (x2)
	.LRV	Dual-fisheye	H.264	yuvj420p	1024 x 512	29.97	3999
Garmin VIRB 360	.MP4	Equirectangular	H.264	yuv420p	3840 x 2160	25	80,008
	.LRV	Equirectangular	H.264	yuv420p	1280 x 720	25	5026
Ricoh Theta S	.MP4	Dual-fisheye	H.264	yuvj420p	1920 x 1080	29.97	15,938

Table 1. A comparison of the original files from four different 360-degree cameras.

3.1 GoPro MAX

The GoPro MAX records in 3K resolution at 50 frames per second (fps) or 5.6K resolution at 25 fps. We chose the latter in this experiment since spatial resolution is more important than temporal in our ongoing project. The GoPro MAX produces many files for each recording. Three file types are stored: a high-resolution file (.360), a low-resolution file (.LRV), and a thumbnail image of the first video frame (.THM). For compatibility reasons, GoPro stores the .360 files in chunks of size 4.0 GB (the limitation of FAT32-formatted drives). It also stores the .LRV files in chunks of similar duration as the .360 files, even though that is unnecessary from a file format perspective.

Each .360 file contains 7 data streams: two video streams, two audio streams, and three data streams. The two video streams contain data that is similar to Google's Equal Area Cubemap (EAC) [2], but optimized to squeeze in as many pixels as possible [3]. The data stream with handler name "GoPro MET" contains metadata in GoPro's open source metadata format GPMF [4]. The "GoPro TCD" stream contains a starting timecode, representing the time since midnight of each frame, and the "GoPro SOS" data stream contains data for file recovery.

The GoPro MAX comes with the GoPro Player software that provides basic editing and export functionalities. The software allows users to edit viewing angles as key frames and export a reframed video, but there is no option to export the video in full 360 degrees. It also contains paid functionalities, such as advanced stabilization algorithms that require a subscription. Apart from the GoPro Player, the company also provides an open-source GPMF-parser [5] that can extract the metadata in their own GoPro Metadata Format. However, the GoPro Player does not have open-source code for stitching and converting their customized Equal Area Cubemap projection into common public formats such as Equirectangular Projection, which makes it hard for advanced users and researchers to process their videos with open-source tools such as FFmpeg.

[2] https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/
[3] https://gopro.com/en/us/news/max-tech-specs-stitching-resolution
[4] https://gopro.com/en/sg/news/gopro-video-metadata-open-source-explained
[5] https://github.com/gopro/gpmf-parser

3.2 Insta360 X3

The Insta360 was set to record in 5.7K resolution at 29.97 fps. It saves two types of video files: two high-resolution single-fisheye files (.INSV) and a low-resolution dual-fisheye file (.LRV). Each file is only 1 minute long, which makes for a lot of files for longer recordings. This design choice may be smart from an action camera perspective, where crashes would not lead to corrupt files, but it is annoying with numerous recorded files for longer recordings. Both .LRV and .INSV files contain only two streams, one video and one audio. There are no metadata streams.

The Insta360 Studio 2024 contains various editing functionalities. Apart from basic trimming and editing, it also supports stabilization, motion ND, object tracking, and color filtering. It is the only software that supports various stitching configurations, including dynamic stitching, optical flow-based stitching, chromatic calibration, etc. Users can choose between a reframed export or a full 360-degree export.

3.3 Garmin VIRB 360

The Garmin VIRB 360 is a legacy product, but it has been a popular camera for many years, and we have used it for numerous concert recordings. It offers 360-degree recordings in 4K, automatically stitched by the camera, and RAW 5.7K recordings, which the user can stitch together using proprietary software. The RAW option records and outputs two separate 200-degree hemispherical videos from the lenses as separate files, with a total resolution of 5.7K. In this experiment, the camera was set to record in 360 at 4K. The manual and packaging of the Garmin VIRB 360 states that it records in 30 fps [6]. However, this functionality is preceded by the camera's regional video format settings (whether set to PAL or NTSC). In Europe, PAL is the standard for video recordings. This sets the frame rate to 25fps by default.

The camera saves three files for each recording: one high-resolution equirectangular recording (.MP4), one low-resolution equirectangular recording (.GLV), and a thumbnail image (.THM). Both the .MP4 and .GLV files contain one video stream and one audio stream. There are no metadata streams.

[6] https://static.garmin.com/pumac/Virb_360_OM-EN.pdf

Camera	Software	Export projection	Export view range	Codec	Export resolution	FPS	Comment
GoPro MAX	GoPro Player	ERP	reframed,full 360	H.265, H.264, ProRes	up to 5376 x 2688 (5.6K)	25	
	GPMF-parser	-	-	-	-	25	exports metadata
Insta360 X3	Insta360 Studio	ERP	reframed, full 360	H.265, H.264, ProRes	1920 x 1080 (reframed), 5760 x 2880 (full 360)	29.97	
Garmin VIRB 360	Garmin VIRB Edit	ERP	reframed, full 360	H.264	1920 x 1080 (reframed), 3840 x 2160 (full 360)	25	
Ricoh Theta S	Ricoh THETA	ERP	full 360	H.264	1920 x 960	29.97	no editing functions

Table 2. A comparison of the export functionalities of different software. EAC stands for Equi-Angular Cubemap, and ERP stands for Equirectangular mapping.

The Garmin VIRB Edit software provides basic editing functionalities, including trimming and adjusting the viewing angle. The software supports the G-Metrix metadata format for more functionalities, such as motion detection and GPS localization. However, the Garmin VIRB 360 camera does not provide any G-Metrix data in its recordings, so this option may be related to compatibility with other hardware devices from the same manufacturer. As for export options, the software supports both reframed and full 360-degree export.

3.4 Ricoh Theta S

The Ricoh Theta S is the oldest camera in our selection and offers the fewest options. It only records one 1080p, dual-fisheye .MP4 file with a single video stream. As such, it is the easiest camera to work with from a file format perspective. However, it uses an unusual dual fisheye projection, with each spherical view shifted 90 degrees.

The Ricoh THETA editing software is basic, only supporting simple playback and conversion functionality. It does not offer editing options but only allows the user to import a single dual-fisheye file and export it to a 360-degree file with an equirectangular projection.

4. VIDEO QUALITY COMPARISON

Since 360-degree videos have a long processing pipeline, different distortions might be introduced, even if we do not consider the intrinsic differences between the cameras. Moreover, the distortions vary drastically for different camera models, and these differences are usually not mentioned in their user manuals. These differences might cause undesired analysis errors if researchers compare camera recordings. In this section, we will compare the distortions and the quality of different camera models so that researchers can avoid some of the errors in their experiments.

4.1 Stitching distortion

The recordings were compared using equirectangular projection to understand the stitching distortion in the cameras' output. A common effect in 360-degree cameras is the distortion of objects in the near field. All our cameras have two lenses: one in the front, the other in the back. The field-of-view (FOV) of one lens in any of the cameras in this experiment is unknown. Indeed, any FOV value exceeding 180 degrees will inevitably result in a parallax error at certain points in the combined image. The lenses cannot physically be precisely at the same point, just like human binocular vision. Both eyes have slightly different viewing angles; focusing on an object far ahead produces a clear image, but as the object draws near, the image gets distorted [6]. In equirectangular projections from 360-degree videos, this distortion manifests at the stitching points.

Similar to how the human nose masks a part of the visual field, a camera's body blocks its own view. Image stitching attempts to compensate for this loss. Distortion can appear as misalignment, ghosting, and disappearance of objects as the views from a camera's front and back lenses overlap at the stitching points.

Figure 3 shows one frame in equirectangular projection at the same time stamp for the four cameras—all display stitching distortion. The GoPro shows misalignment and disappearance but the least amount of ghosting. The Insta360 shows all three kinds of distortion. The Garmin has some misalignment and ghosting and the greatest amount of disappearance. The Ricoh also features a sizable level of all three distortions and is worse than the Insta360.

4.2 Camera response to varying light intensity

To get a sense of how the cameras handle different light conditions, we used a dimmer to gradually reduce the light level in the laboratory. The light level was adjusted and

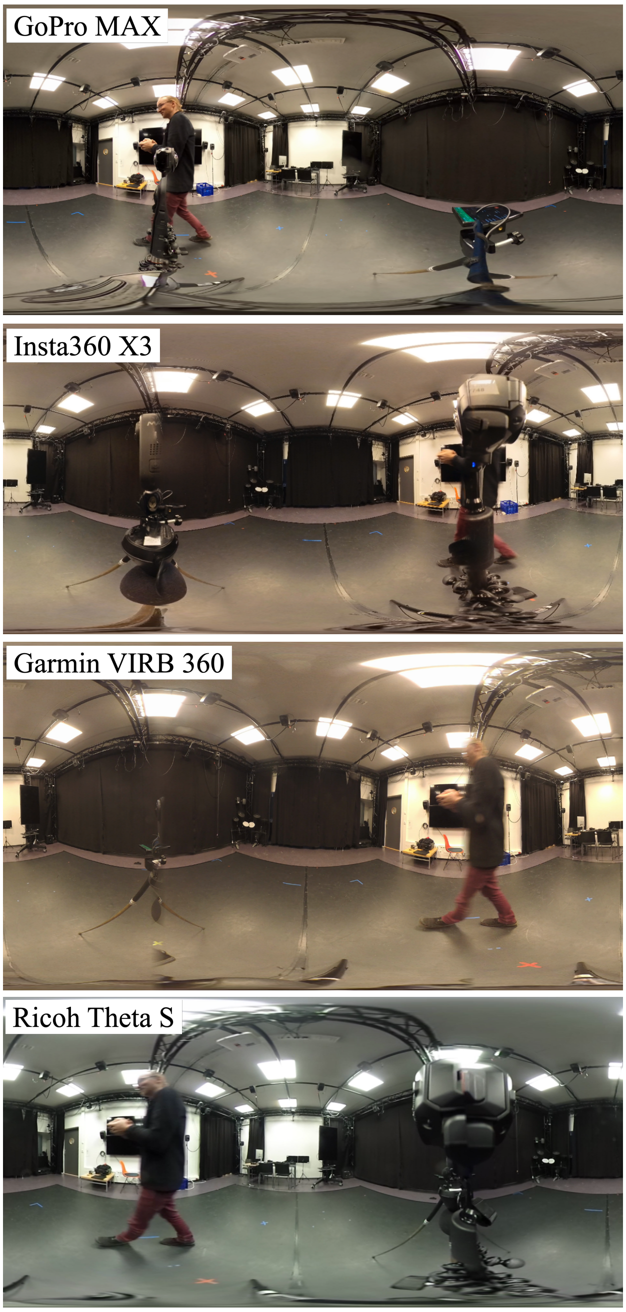

Figure 3. Equirectangular projections of the four camera recordings in the same frames, showing the different ways of stitching distortions at different positions.

held for a few seconds at each intensity, resulting in eight light intensity levels. A Samsung Galaxy S23 Ultra mobile phone was set to record the light intensity in the room with the Android app Physics Toolbox Sensor Suite 1, developed by Vieyra Software (ver. 2023.01.21). The RGB and Brightness graphs for each camera are plotted in Figure 4. The graphs demonstrate the differences in the cameras as they adjust to eight lighting levels. The RGB adjustments manifest in different levels of red, blue, and green in the graphs in any lighting condition. The plot shows blue as the highest intensity, followed by green and red. The Ricoh only does a little RGB color adjustment with bright light and virtually nothing when the light intensity decreases.

Cameras manipulate RGB levels to allow for automatic white balance adjustment as the lighting changes. This ensures that the white parts of the captured frame appear neutral regardless of the lighting level, which helps maintain an accurate representation of colors. Some cameras have an additional automatic color enhancement feature to produce aesthetically pleasing, vibrant videos. Besides automatic white balance and color enhancement, video recordings often undergo exposure compensation [7]. This feature, again through RGB adjustment, produces an optimal brightness level by preventing over- or under-exposure in sub-optimal lighting conditions. The solid black line in Figure 4 shows the brightness level, essentially a function of the RGB values first converted to grayscale. All four cameras respond to all lighting levels. As a general trend, the RGB and brightness values dip first and then increase to a new steady state.

4.3 Projection and compression distortion

While all four cameras use a dual-fisheye lens setup, the manufacturers designed different algorithms to save their files as described in Chapter 3. The projection choices and the compression algorithms that work on the selected projection will create different artifacts. Figure 1 shows a frame of the original files from the cameras.

The GoPro MAX uses a modified equi-angular cubemap, making two 4096 x 1344 streams of video, each compressed to a bitrate of about 30,000 kbps. However, since the two streams have different fields of view (FOV), the equivalent per-angle bitrate of the two streams is different.

The Insta360 X3 saves the two fisheye lenses in separate files, with a rectangular resolution of 2880 x 2880 each. This makes it the highest equivalent full-360 resolution among the four cameras. Combined with the fps of 29.97, it also has the highest equivalent full-360 bitrate.

The Garmin VIRB 360 uses a standard equirectangular projection, with the second-highest bitrate of 80,000 kbps. However, due to the polar expansion of the equirectangular projection, some redundant pixels near the poles are encoded with more than enough bits. Compared to the EAC projection without poles or the dual-fisheye with easily compressible black edges, it is a less compact method for file storage.

The Ricoh Theta S saves the two fisheye lenses into a single video file. Not only does it have the lowest bitrate, but it also saves the file as a standard 1080p 16:9 rectangular video, which is not in the 2:1 ratio of the dual-fisheye image. This resulted in the extra black edge at the bottom and the abnormal 1920x960 resolution in the software-exported equirectangular video, as shown in Figure 3.

5. EXTENDING THE MUSICAL GESTURES TOOLBOX

The overview above shows that all cameras record video with different projections, file formats, pixel dimensions, and frame rates. This makes it challenging to establish uniform workflows when handling recordings from different cameras. We have, therefore, implemented tools in our

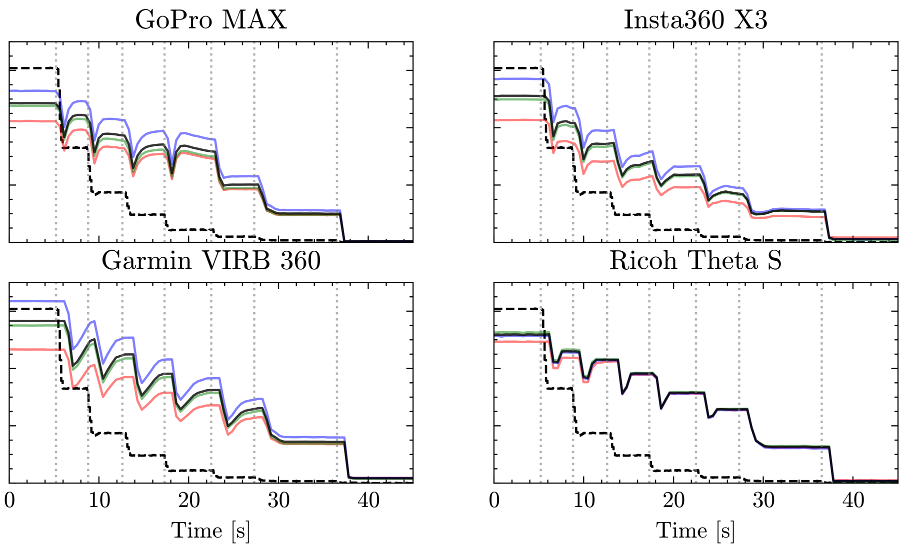

Figure 4. RGB (colored) and summed (solid black) light intensity of the four cameras compared to light intensity captured by the phone (dashed line). The units are different: the phone captured the intensity in lux, while the cameras' intensity value is the averaged pixel value ranging from 0 to 255. The cameras are calibrated by their internal algorithms.

Musical Gestures Toolbox (MGT) [8] for handling the different file types so that they can be used for further analysis. This is implemented in a new class, `Mg360Video`, that contains utility functions for 360-degree video merging, projection conversion, etc. The class inherits from the main video class `MgVideo`, meaning all the current demonstration and analysis functions in the toolbox can now be used directly on 360-degree videos.

5.1 File merging

Both the Insta360 camera and GoPro MAX cameras store high-quality video in multiple files using proprietary formats (.INSV and .360, respectively). These segments needed to be concatenated into one file to facilitate subsequent analysis. Fortunately, both the .INSV and .360 files can be read by FFmpeg. Since they contain video streams that use standard video compression inside (H.264 and H.265, respectively), storing the streams in a standard container is possible. In the updated `MgVideo` and `Mg360Video` classes, we now support inputting a list of filenames that will automatically be merged using FFmpeg's `concat` function. We have decided to export files in the Matroska file format (.MKV), an open format that supports unlimited video, audio, picture, or subtitle tracks in one file. This is a versatile video container widely compatible with various software applications. This approach ensures that the exported content maintains the original video and audio specifications with minimal alteration.

5.2 Projection conversion

The `Mg360Video` class supports projection conversion with the `v360` filter function of FFmpeg [7]. This filter supports conversion between several common 360-degree video formats. When creating a new object, the original projection information is needed. Then, it can be converted into more than twenty different projections using the `Mg360Video.convert_projection()` function, as shown in the first row of Figure 5.

[7] `https://ffmpeg.org/ffmpeg-filters.html#v360`

5.3 Motion analysis

The Musical Gestures Toolbox supports various motion analysis functions that are now compatible with 360-degree videos, including motion average/history images/videos, videograms, motiongrams, self-similarity video matrices, etc. The second row of Figure 5 demonstrates a motiongram in the vertical direction and an extracted motion video frame.

6. CONCLUSIONS

Through desktop investigations and a lab experiment, we have found that the four 360-degree cameras in question all have different video qualities, file formats, projections, and software support. Given their (relative) commercial success, we are surprised by the lack of technical documentation and file conversion support. The GoPro MAX has the best overall video quality, good software, and detailed metadata. Still, its proprietary .360 file format with a customized video projection makes it hard to work with the files from a video analysis perspective. The Insta360 X3 has the highest original resolution, good software support, and unique stitching adjustment functionality. The fish-eye files can be merged and stitched using the Musical Gestures Toolbox, thus making it easier to do batch processing and analysis. Given their lower visual quality, we would not recommend the two legacy cameras (Garmin VIRB 360 and the Ricoh Theta S) for research purposes. However, since we (and others) have numerous recordings from these cameras in our archives, we must also find solutions for handling these files.

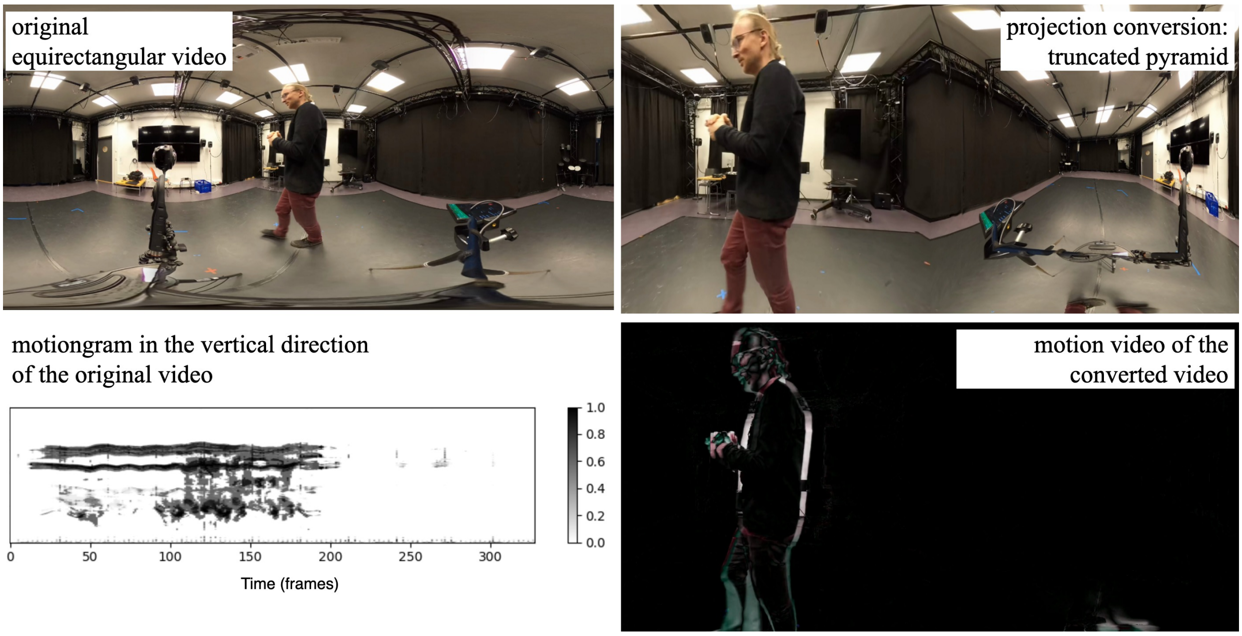

Figure 5. Examples of the new functionality of the extended Musical Gestures Toolbox: projection conversion, motiongram analysis, and motion video generation.

Support for 360-degree videos in the Musical Gestures Toolbox for Python allows researchers to perform file merging and simple projection conversions in a package that already supports numerous video visualization and analysis pipelines. This is crucial for our sonic environment research and will hopefully inspire others to add 360-degree recordings to their data collection.

Moving forward, we hope that manufacturers can work together to support a standardized, open-source representation of 360-degree videos that contain necessary metadata, including projection and stitching information. This can hopefully be the starting point for an open software ecology that supports recording, processing, editing, and analysis from all 360-degree cameras. We will also keep updating the 360-degree video-related functionalities in the Musical Gestures Toolbox, such as 360 video cropping and support for raw camera output formats.

Acknowledgments

This project is supported by the Research Council of Norway through projects 262762 (RITMO), 250698 (MICRO), 324003 (AMBIENT), and 322364 (fourMs).

7. REFERENCES

[1] Netflix Production Technology Resources, "Into the Volume: A Behind-the-Scenes Look into the Virtual Production of 1899," Dec. 2022. [Online]. Available: https://www.youtube.com/watch?v=ZMynJCgJIQk

[2] F. Chiariotti, "A survey on 360-degree video: Coding, quality of experience and streaming," *Computer Communications*, vol. 177, pp. 133–155, Sep. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036642100253X

[3] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, "Visual Distortions in 360° Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2524–2537, Aug. 2020, conference Name: IEEE Transactions on Circuits and Systems for Video Technology. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8756213

[4] D. Salomon, *Transformations and projections in computer graphics*. London: Springer, 2006, oCLC: 209949994.

[5] M. Riaz, J. Guo, and A. R. Jensenius, "Comparing Spatial Audio Recordings from Commercially Available 360-degree Video Cameras," submitted.

[6] F. Cutolo, N. Cattari, U. Fontana, and V. Ferrari, "Optical see-through head-mounted displays with short focal distance: Conditions for mitigating parallax-related registration error," *Frontiers in Robotics and AI*, vol. 7, p. 572001, 2020, publisher: Frontiers Media SA.

[7] S. Bianco, A. R. Bruna, F. Naccari, and R. Schettini, "Color correction pipeline optimization for digital cameras," *Journal of Electronic Imaging*, vol. 22, no. 2, pp. 023 014–023 014, 2013, publisher: Society of Photo-Optical Instrumentation Engineers.

[8] B. Laczkó and A. Jensenius, "Reflections on the Development of the Musical Gestures Toolbox for Python," in *Proceedings of the Nordic Sound and Music Computing Conference*, Copenhagen, 2021. [Online]. Available: https://www.duo.uio.no/handle/10852/89331