

Synchronizing physiological data and video in a telemedicine application: A multimedia approach

Laurent Lambert, Khalil Hachicha, Zahid Syed Ahmed, Andrea Pinna, Patrick Garda

▶ **To cite this version:**

Laurent Lambert, Khalil Hachicha, Zahid Syed Ahmed, Andrea Pinna, Patrick Garda. Synchronizing physiological data and video in a telemedicine application: A multimedia approach. Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE, Aug 2015, Milan, Italy. pp.181-185, 10.1109/EMBC.2015.7318330 . hal-01256638

HAL Id: hal-01256638
https://hal.science/hal-01256638

Submitted on 15 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synchronizing Physiological Data and Video in a Telemedicine Application: a Multimedia Approach

Laurent Lambert, Khalil Hachicha, Syed Zahid Ahmed, Andrea Pinna and Patrick Garda
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606,
CNRS, UMR 7606, LIP6, F-75005, Paris, France
firstname.lastname@lip6.fr

***Abstract*— Several medical examinations require that multiple modalities of exams are stored in a synchronized manner. For instance, an EEG exam needs that several physiological signals along with video of the exam are acquired synchronously to aid the neurophysiologists to perform their diagnostics. Furthermore support for telemedicine for such exams have become important in recent years. The existing EDF standard that is used for physiological signals makes it difficult to provide integrated support of adding video and compressed component data, however due to widespread use of EDF standard in the domain, cross compatibility with EDF standard for physiological data is also essential. We present in this work a novel idea to solve these issues. Our approach uses standard multimedia containers in which physiological data are embedded alongside video and audio. This paper provides our analyses of the state of the art of multimedia containers EDF, AVI, ASF, MPEG and MKV and their potentials for a telemedicine application and outlines how MKV stands out as an interesting option in this regard, allowing also capability of compression of physiological data if needed.**

I. INTRODUCTION

Telemedicine, in recent years, has emerged as a major new trend in medicine that exploits advancements in technologies such as communications, high speed internet and multimedia. The neurophysiological exams such as Electroencephalogram (EEG) are now often done with video recording of the exam, in order to facilitate remote diagnostic via telemedicine. This leads to two key issues. Firstly the equipment has to record all the exam modalities (physiological signals, audio, video, exam annotations, etc.) in a highly synchronized manner. The word synchronization refers to time and more precisely the temporal relations between media objects in a multimedia system. Secondly the data container format that ensures that all the gathered information are stored in a format that preserves all the synchronization information and makes the data exploitable by end user applications for diagnostics whether in-house, in same hospital or somewhere else via telemedicine.

Smart-EEG project [1], [2] aims to resolve both issues and it envisions a solution that provides highly synchronized data acquisition of all the exam components and stores them into a format that uses existing standards adapted to store this timed exam data. This paper is oriented to this second part and outlines the contributions of this work by proposing an MKV [3] based solution.

This project is supported by Conseil régional d'Ile-de-France

The EDF standard [4] is widely used in hospitals to store the physiological data, all the medical tools used by the hospitals use EDF format or are EDF format compliant. This ensures interoperability among several hospitals or doctors even if they are not using the same tool. Hence, remaining compliant with EDF standard due to legacy reasons is an important aspect. However EDF standard was not designed to incorporate multimedia signals such as video, moreover it does not support compressed signals. This leads to a challenge to find an innovative manner that gives upgraded facilities that are needed to store timed information of multimodes, is streaming friendly as we are targeting telemedicine and yet also provides facility to keep EDF compliance of neurophysiological data if needed.

This paper is outlined as follows. Section II describes the state of the art in existing video-EEG exams and the limitations it has in terms of synchronization. Section III highlights two possible models of storing timed information data that can lead to multiple implementations that are discussed in Section IV. Section V presents our key innovations and how they help in resolving the problem. Finally Section VI concludes the discussions and mentions future outlook.

II. STATE OF THE ART

Current synchronization between physiological data and video are performed by saving the physiological data inside an EDF file and video in a multimedia container. A container is a file or stream which contains data of different kinds. It provides features such as temporal synchronization, encryption, tags, chapters, description of contained streams. The EDF file stores raw polygraphic recording while the video file usually contains compressed video and audio data. This encapsulation hierarchy is depicted in the figure 1.

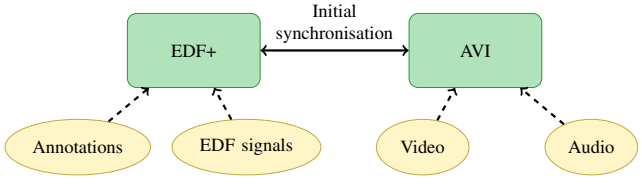


Fig. 1. Current synchronisation between EDF and video

When playing back both type of signals, the two files are started synchronously and are assumed to stay synchronous

during the entire sequence. It is easy to understand that one of the playbacks may drift in time and therefore loses synchronization. This methodology to store timing information is unsuitable for telemedicine usage where timing relationship between video and physiological signals is important. The drawbacks of this methods are :

- Playback synchronization cannot be guaranteed because the EDF file and the multimedia file does not have a common time base.
- Physiological data is stored uncompressed. This limitation impacts both transmission bandwidth for telemedicine usage as well as storage for commodity tests. EEG tests performed over night are usually several Gigabytes and may benefit from a compression scheme.
- Adding or removing information inside the EDF file involves recreating a complete file.

Many Previous works have attempted to synchronize video and physiological data. In [5], Zeng et al introduced an acquisition system record allowing to acquire biomedical data along with video. Gackowski et al [6] developed internet based tele Dicom to make the teleconsultation of medical image possible. Real time voice communication, visualization of synchronized digital imaging and communications is offered. Guerri [7] propose to use Java media framework (JMF) to synchronize audio, video , electromyograph data and lumbar device data. To achieve synchronization, the authors uses a glue file which maps timing from one file to the other. Campos [8] has demonstrated the use of EEG by remotely viewing the acquisition screen. Even if this method gives close to perfect synchronization, it generates only a video file. This method makes hard to retrieve signals from the received stream. An alternative solution to provide compression alongside synchronization is to perform a data hiding method [9]. In this method, one modality becomes a container for others. Usually, each image of a video sequence embed the data of the relative modalities. This way, when the decoder reads an image, it retrieves audio and physiological data. This scheme provides good synchronisation but has some drawbacks:

- One error in an image discard the related window of time.
- Removal of any track leads to decoding the main track, removing appropriate data and re-encoding the main track. Also, the removal of the main track is problematic and a new main track needs to be elected.
- Using a video codec to encoded audio or physiological data lead to less efficient solution compared to specialized codec for every modalities.

III. Container Models

A stream of multi-modal data can be represented in a hierarchical manner. At the lowest level are samples. Samples are timed indivisible units of information. This represents the value of a physical quantity at a given time. For a video sequence, an image is a sample whereas for one dimensional signals such as audio, it is a scalar. In the latter case, samples are packed together temporally into Logical Data Units [10] (LDU) which are the second hierarchical level. LDUs are the atomic data which gets processed by a data container. Samples inside an LDU may be compressed to form a binary sequence. For that reason, LDUs are usually media agnostic that is it does not make any assumptions on the contained data. Those LDUs are categorized by their type to form the last hierarchical level: tracks (video, audio, physiological data). They are interleaved by the container handling synchronization. Often, they are ordered in chronological order. For completeness, a higher level exists which multiplexes channels or streams. This is, for example, used in TV broadcasting to stream multiple channel at once. Figure 2 exposes this encapsulation hierarchy.

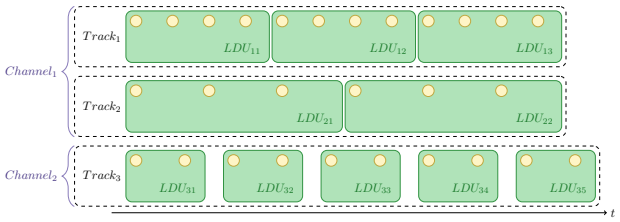


Fig. 2. Multimedia encapsulation, this example showcases two channels containing 2 tracks and 1 track respectively. In track one, each LDUs contain 4 samples, in track two each contains 3 samples each and in track three each contain 2.

A. Constant sample rate

The simplest model assumes that every LDU in a track is of constant rate. To calculate the maximum precision to place a LDU we use the following equation:

$$\mathcal{E} = \frac{1}{2 * LDU_{rate}}$$

The LDU rate can be tricked by inserting dummy samples containing unreachable values or duplicated samples. In doing such, the LDU rate increases and therefore decreases the maximal error of the model. However, inserting those extra samples increases the file size. It is a trade off between precision and file size. Figure 3 illustrates two tracks synced by a constant rate container.

B. Timestamps based

The second model assumes non constant LDU rate. Therefore, LDUs are placed on a virtual time line referred to as

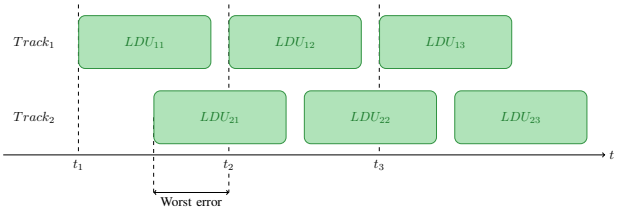


Fig. 3. Example of a constant sampling rate model. First track contains perfectly synced LDUs with the expected timings. The second track exhibits the worst precision, LDUs are exactly in between two expected timings.

the reference clock, which is of constant frequency. At each period of this virtual clock, a virtual counter is incremented. This counter represents the number of periods since the beginning of the media. Its value is conveyed alongside the LDU. The frequency at which the reference clock is incremented defines the finest grain at which you can place a LDU and is closely related to the maximum precision. Maximum error is when an event occurs right in between two ticks.

$$\mathcal{E} = \frac{1}{2 * F_{refclk}}$$

The width of the counter defines the time before the counter overflows. Usually, the counter is taken big enough so that this case does not happen. Figure 4 illustrates two tracks synced by a constant rate container.

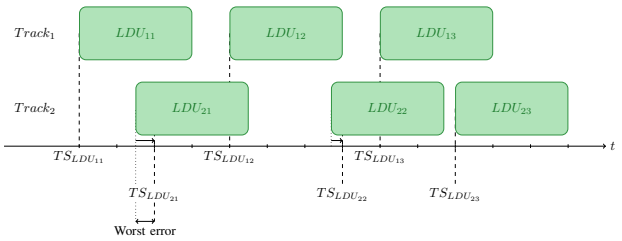


Fig. 4. Example of a timestamp model. First track contains perfectly synced LDUs with the reference clock. Second track exhibits the worst precision when LDU_{21} is exactly in between two reference clock ticks. Arrows show the rounded timestamp.

IV. Container implementations

A. Constant sample rate implementations

*1) **EDF+**:* An EDF+ file is composed of two parts, the header which contains information about the file and its signals. It is followed by data records containing actual data samples. One data record contains samples of every channel in a window of time. The size of this window is defined in the header. The size of the place holder for each sample is 16 bits. The LDU rate can be calculated as follows:

$$LDU_{rate} = \frac{N_{samples}}{T_{datarecord}}$$

Where $T_{datarecord}$ is the minimum time of a data record, $N_{samples}$ is the number of samples in one data record, the ratio of the two being the sample rate. The error is therefore directly proportional to the sample rate. $T_{datarecord}$ and $N_{samples}$ are both 8 ASCII characters. The error is minimised when $T_{datarecord} =$ "0.000001" and $N_{samples} =$ "99999999" which gives $\mathcal{E} = 50fs$. This case would lead to a file growing at a rate of 20 Tbytes/seconds to store physiological signals only.

EDF+ has been proposed to add annotations support and non-contiguous recording by introducing a metadata channel. The place reserved for this channel is constant in every data record whether or not it is used to store annotations. This space is often unused and therefore wasted as it contains no information. The company BioSemi [11] has specified a BDF and BDF+ file to increase the size of a sample from 16 bits to 24 bits.

*2) **AVI**:* Audio Video Interleaved [12] is a media container introduced by Microsoft. It is only designed to convey compressed video, audio and subtitles. The rate at which LDUs are presented is defined in the header of the file. Those rates are constant throughout the file. The sampling rates are written on 32 bit words and therefore the maximum expressible sampling rate is 2^{32}. Thus, The maximum worst error is $\mathcal{E} = \frac{1}{2*2^{32}} = 0.11ns$.

This elementary container was later replaced by ASF.

B. Timestamp implementation

Most timing based implementations share the same hierarchical design. They are based on serialised objects which are identified by unique ID (GUID). An object may contain specific data specified by its ID or sub-object. For example, an object LDU is created which contains both a timestamp object, a data object and other meta data objects (specific track information).

*1) **ASF**:* Advanced Systems Format [13] is an object oriented container developed by Microsoft. The primary goals of ASF are efficient streaming capabilities and independence of any particular operating system and communications protocol.

It supports the inclusion of text streams, scripts and user defined data structure. LDUs are expressed in unit of 100ns which means that placing an LDU is in the worst case 50ns off timing.

*2) **The MPEG family**:* The MPEG systems family (MPEG1 systems [14], MPEG2 systems [15], MPEG4 systems [16]) specifies how encoded LDUs are identified, synchronized and how meta-data are added to a stream. It is not to be mistaken with MPEG compression standards, which defines how data gets processed to reduce the size of an LDU.

In the MPEG1-systems specification, the multiplexing layer handles only video, audio and subtitles. Its synchronization layer uses a reference clock of 90 kHz, which induces a worst case error of 5.5us.

MPEG2-systems specification, improves those features by adding a user defined tag to the multiplexing layer and uses a 127Mhz reference clock. It also separates the usage of a single stream (program stream) used in DVDs for example and the transport of multiple streams (transport stream) used for broadcast applications.

MPEG4-systems gives the user the ability to configure the reference clock frequency. This frequency is at maximum 2^{32} Hz, meaning that the best maximum error induced is 0.12ns. MPEG4-systems does no assumptions on the data it contains.

*3) **MKV**:* Matroska [3] is an open standard container format. It was developped to offer a patent free alternative to the MPEG-systems family. It supports both streaming and file storage. MKV was designed to be flexible and extendible therefore no supposition are done on the LDUs. Timestamps are multiples of 1ns.

Furthermore, the format is based on a extendible language called EBML which enables it to have features such as seeking, chapters, meta-data inclusion, menu and encryption. This last part is useful in the field of telemedicine where, medical data secrecy is crucial.

*4) **SMIL**:* Synchronized Multimedia Integration Language [17] is an XML based synchronization language, which is actively used in Multimedia Messaging Service and for the SVG file format. It was designed for web applications by the World Wide Web Consortium. Its goal is to provide a temporal and spatial description of a scene containing heterogeneous media. The SMIL description specifies the relationships between media and provides hyperlinks to the actual data. It is not a container strictly speaking. This actually involves having multiple other files containing the actual data.

Although, SMIL represents time using timestamp, temporal information of an object is specified relatively to others. This usually means that the size needed to store temporal information is smaller. These relationships are described as h:min:s.ms., in which the number of decimals of milliseconds is unlimited. In theory, an error of zero may be achieved.

However, since the file is a textual description, its memory footprint is very large compared to a binary description. Moreover, SMIL is hardly extendible to use media that are not images, videos, audios and texts. Those two factors make it a bad candidate for our solution.

Table I summaries this section by comparing all implementations. Non timestamp implementations are not considered in our solution, as they involve increasing the amount of data to adjust the needed precision. Furthermore, SMIL does not meet our need of compactness since it has a textual representation. We choose MKV in our application because it's implementation is open and has no royalties unlike MPEGs container and ASF. However, those containers may be used if this criteria is not mandatory in other contexts.

V. Our Solution

Our solution is based on the MKV implementation. In order to incorporate physiological data, we have introduced a new track alongside the video and audio, which contains the physiological samples. Physiological data are chopped into windows of time to produce packets of data. In fact, each physiological LDU is created by copying a data record from an EDF+ file. By doing so, MKV provides resynchronization points between each data record. EDF+ is used as a codec inside of MKV. These LDUs are interleaved with video and audio packets by the MKV layer. We are using FFmpeg's implementation [18] of MKV to manage the synchronization layer. On the decoder side, LDUs are demultiplexed using their track ID and are placed temporally using the timestamp of each LDUs. Temporal relationship of samples inside a LDU is deduced by the sample rate.

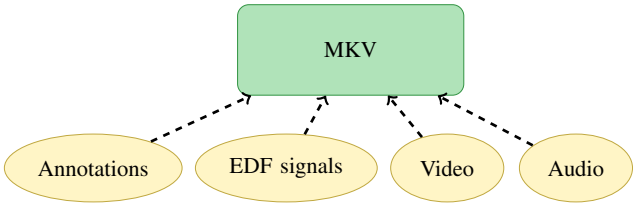


Fig. 5. Our solution to synchronize physiological data and Video

An implementation of this method composed of two modules is currently functional:

- An encoder side: encodes individual modalities with specialized codecs (video, audio, exg). From them, encoded LDUs are interleaved using MKV. The stream is sent on a TCP/IP connection.
- A decoder side: reads the mkv stream from a TCP/IP connection. MKV provides the demultiplexing capabilities and the synchronisation layer for each modality. LDUs are sent to the appropriate decoder and are then displayed synchronously.

Along side this experiment, the data which travels on the TCP/IP connection is dumped in a file. The decoder is capable of reading this file and playing it back. This showcase the possibility of using this method for streaming application or for storage.

Our application uses this synchronisation scheme during an EEG exam, it includes the acquisition of Electroencephalogram, Electrocardiogram, Electromyogram, Electrooculogram as well as video and audio. This recording is afterward completed with the annotations of a technician. As a whole, this exam is sent to an expert for tele-diagnosis. It is also used for long term storage in an archive server.

Embedding physiological data inside a timestamp based container has multiple advantages:

- **Resynchronization**: Since every timing relationship is expressed as a common fine-grain time base, the capacity to correct the slight drift that one track may suffer is preserved.
- **Compression**: Because this scheme may be used in a telemedicine context, the usage of a compression algorithm for physiological data is taken into consideration. Even if our implementation does not compress physiological data yet, the synchronization scheme does not restrain the capability of holding compressed data.
- **Insertion and withdrawal**: Using multimedia container also simplifies the insertion or withdrawal of one track, as each track is independent. To remove a track, a simple filtering of LDUs of this type is necessary. To insert a new track, it may be accomplished by creating a new stream and interleaving the new LDUs. This operation can be achieved on the fly when reading a stream. Both operations do not require to decompress other media data.
- **Non-contiguous data**: non-contiguous tracks, such as annotations, are easily integrable and have low overhead since no place is reserved for potential data. The annotations are present in the stream only when needed.

Finally, as EDF is used as a codec inside the multimedia container, extracting the original EDF file is as simple as

TABLE I

SUMMARY TABLE OF THE PERFORMANCE AND FEATURES OF EACH CONTAINER

Container	Maximum error	Support compressed data	Timestamp model	Free of usage	Agnostic	Binary format
EDF+	50fs	N	N	Y	N	Y
AVI	0.11ns	Y	N	N	N	Y
ASF	50ns	Y	Y	N	N	Y
SMIL	unlimited	Y	Y	Y	N	N
MPEG 1 systems	5.5us	Y	Y	N	N	Y
MPEG 2 systems	4ns	Y	Y	N	Y	Y
MPEG 4 systems	0.1ns	Y	Y	N	Y	Y
MKV	0.5ns	Y	Y	Y	Y	Y

concatenating every LDUs of the physiological track. This is important since most hospital may not own the required software to read this new file format.

VI. CONCLUSION

Physiological signals coupled with synchronized video helps in improving interpretation of exams in which ordering of events is critical. In this paper, we have presented how we have integrated physiological data inside a multimedia container for streaming and storage purposes. This scheme is flexible enough to provide compression of physiological data if the need arises. Moreover, deleting or inserting new correlated data in a stream is simple enough as each modality remains independent. Furthermore, retrieving EDF file from our stream is trivial. These qualities make this solution a good candidate in the field of telemedicine.

Future works include integrating this file format as an exchange file in the Smart-EEG project. This synchronization mechanism will be used in a clinical evaluation of telemedicine applied to the EEG tests in France.

ACKNOWLEDGMENTS

This paper is based upon the work in the Smart-EEG project that is funded by the Conseil régional d'Ile-de-France. This project is conducted jointly with the companies CIRA, Partelec, ACACIA, 2CSI, the hospitals Georges Pompidou and Lariboisière, the ETIS labs and LIP6. The authors thank all the project members for discussions

REFERENCES

[1] Laurent Lambert, Imen Dhif, Mohammed Shaaban Ibraheem, Syed Zahid Ahmed, Bertrand Granado, Khalil Hachicha, Andrea Pinna, and Patrick Garda. Smart-eeg : A new platform for tele-expertise of electroencephalogram. *Colloque du GDR SOC-SIP*, 2012.

[2] Syed Zahid Ahmed, Yuhui Bai, Imen Dhif, Laurent Lambert, Imen Mehdbhi, Patrick Garda, Bertrand Granado, Khalil Hachicha, Andrea Pinna, Fakhreddine Ghaffari, Aymeric Histace, and Olivier Romain. SmartEEG : a multimodal tool for EEG signals. In IEEE, editor, *Proceedings of IEEE FTFC Conference*, pages 1–4, Monaco, Monaco, May 2014. FUI 15.

[3] Steve Lhomme. Mkv. `http://www.matroska.org/index.html`.

[4] Bob Kemp, Alpo Vrri, Agostinho C. Rosa, Kim D. Nielsen, and John Gade. A simple format for exchange of digitized polygraphic recordings. *Electroencephalography and Clinical Neurophysiology*, 82(5):391 – 393, 1992.

[5] Shengke Zeng, J.R. Powers, and H. Hsiao. A new video-synchronized multichannel biomedical data acquisition system. *Biomedical Engineering, IEEE Transactions on*, 47(3):412–419, March 2000.

[6] Andrzej Gackowski, ukasz Czekierda, Anton Chrustowicz, Jacek Caa, Micha Nowak, Jerzy Sadowski, Piotr Podolec, Mieczysaw Pasowicz, and Krzysztof Zieliski. Development, implementation, and multicenter clinical validation of the teledicomadvanced, interactive teleconsultation system. *Journal of Digital Imaging*, 24(3):541–551, 2011.

[7] J.C. Guerri, C.E. Palau, A. Pajares, A. Belda, J.J. Cermeno, and M. Esteve. A multimedia telemedicine system to assess musculoskeletal disorders. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, volume 1, pages I–701–4 vol.1, July 2003.

[8] C. Campos, E. Caudevilla, Alvaro Alesanco, Nelia Lasierra, O. Martnez, J. Fernndez, and Jos Garca. Setting up a telemedicine service for remote real-time video-eeg consultation in la rioja (spain). *I. J. Medical Informatics*, 81(6):404–414, 2012.

[9] Raul Pea, Alfonso vila, David Muoz, and Juan Lavariega. A data hiding technique to synchronously embed physiological signals in h.264/avc encoded video for medicine healthcare. *Article ID 514087*, in press.

[10] G. Blakowski and R. Steinmetz. A media synchronization survey: reference model, specification, and case studies. *Selected Areas in Communications, IEEE Journal on*, 14(1):5–35, Jan 1996.

[11] Biosemi. `http://www.biosemi.com/faq/file_format.htm`.

[12] Microsoft. Audio video interleaved. `https://msdn.microsoft.com/en-us/library/windows/desktop/dd318189(v=vs.85).aspx`.

[13] Microsoft. Windows media video. `http://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=14995`.

[14] ISO/IEC. 11172-1:1993 - information technology – coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s – part 1: Systems. Technical report, RFC Editor, 1993.

[15] ISO/IEC. 13818-1:2013 - information technology – generic coding of moving pictures and associated audio information – part 1: Systems. Technical report, RFC Editor, 2013.

[16] ISO/IEC. 14496-1:2010 - information technology – coding of audio-visual objects – part 1: Systems. Technical report, RFC Editor, 2010.

[17] Philipp Hoschka (ed.). Synchronized multimedia integration language (smil) 1.0 specification, june 1998.

[18] Ffmpeg. `https://www.ffmpeg.org/`.