Video Capture from VHS

This guide takes you through all steps required to obtain best possible captures from your analog VHS tapes. It introduces a method to reduce time sync errors without the use of a high quality TBC (Time Base Corrector), which are expensive and difficult to find. As I do my media capturing and digitizing on a Windows 10 platform this is based on my platform, but would work accordingly for Mac or Linux systems. You need to be able to use the command line to benefit from this guide. Use the table of contents to the right to quickly jump to the section you require.

I'm offering a Simple and Sophisticated Workflow. The first 2 steps are the same, after which you need to decide how much effort you want to put into repairing your video file.

Prerequisites

Gear

- JVC *HR-S6711EU* VHS player
- Blackmagic *Intensity Pro* 4K PCI-e capture card, which is an excellent and affordable capture card.

Software

- Blackmagic *Media Express* 3.7 (Info). If you don't own a Blackmagic capture card you need to adapt this guide to the vendor's own software, or to OBS Studio.
- *HandBrake* 1.5.1, download the GUI and the CLI version, we will use *HandBrakeGUI* and *HandBrakeCLI* to differentiate between the versions.
- *SMPlayer* 22.2.0.10060
- *XnView* MP 1.02
- *Audacity* 3.1.3 with FFmpeg library
- *ffmpeg* and *ffprobe* from FFmpeg Builds binaries for Windows
- *MKVToolNix*
- *GIMP* 2.10.32 with Resynthesizer plugin (instructions for Windows)

Connections

- Video: HR-S6711EU (s-video) → Intensity Pro (s-video through Y - green, B-Y - blue)
- Audio: HR-S6711EU (cinch) → Intensity Pro (RCA in)
- Audio monitor: Intensity Pro (RCA out) → Receiver or active speakers / headphones

Note that the HR-S6711EU has S-Video / Cinch Audio output terminals and a SCART terminal (Composite / S-Video / Audio - in / out)

Settings

Desktop Video Setup (Blackmagic)

- Video Input: S Video
- Audio: Analog Audio Input Levels -3.00 dB

Blackmagic Media Express

This is my preferred video recorder for VHS utilizing a Blackmagic video capture card

- Project Video Format: 625i50 PAL or 525i59.94 NTSC
  Select the one which matches your VHS source, tapes will most likely have been recorded for interlaced scanning. Contrary to what many claim in various articles you should later not scale down the video to VHS' genuine resolution, particularly if you are able to do the recording from a Super VHS capable device.
- Capture File Format: QuickTime Uncompressed 8-bit YUV
  Select this lossless codec over any AVI lossless codec because it allows *ffprobe* video and audio length.

Continue with section SMPlayer

Record VHS with OBS Studio

Alternatively, use OBS Studio to record VHS utilizing a different video capture card

Settings

- Output
  Hint: Chose *Simple* or *Advanced*, not both, then save. *Simple* is good for lossless recording of both video and audio. *Advanced* is only necessary for some special use cases.
    - Output Mode: Simple
        - Recording Path
        - Recording Quality: Lossless Quality
    - Output Mode: Advanced
        - Type: Custom Output (FFmpeg)
        - File path or URL
        - Container Format: matroska
        - Audio Bitrate: 256 Kbps
        - Audio Track: 1
        - Audio Encoder: pcm_s16le
- Audio
    - Sample Rate: 48 kHz
    - Channels: Stereo
- Video
  Hint: The base resolution of the capture card is 720×576. Reducing it by 10 pixels gets rid of

the sync lines at the bottom of the video, but this might depend on your VHS player. The value is not selectable, so click into the *Base (Canvas) Resolution* field and enter 720×566 manually. If you reencode the video with HandBrake later then fix sync lines and other cropping in HandBrake.
    - Base (Canvas) Resolution: 720×566
    - Output (Scaled) Resolution: 720×566
    - Downscale Filter: Lanczos (Sharpened scaling, 36 samples)
    - Common FPS Values: 50 PAL

Properties

- Video Connection: Composite
- Audio Connection: Analog RCA
- Channel: 2ch

Advanced Audio Properties

- Tracks: 1

Filters

- Gain: -1.00 dB
- Limiter: Threshold -1.00 dB, Release 60 ms

Deinterlacing

- Yadif 2x
- Top Field First

SMPlayer

- Enable *Show the current time with milliseconds* in **Options –> Statusbar**

HandBrakeGUI

- Dimensions
    - Orientation and Cropping - 4:3
        - Cropping: Custom
        - Left: 16, Right: 16, Top: 4, Bottom: 12[1)]
    - Orientation and Cropping - wide
        - Cropping: Custom
        - Left: 12, Right: 12, Top: 78, Bottom: 82
    - Resolution and Scaling
        - Resolution Limit: Custom
        - Maximum Size: 692 x 564[2)]

        - Anamorphic: Automatic
        - Scaled Size: Optimal Size
    - Border
        - Fill: None
        - Left: 0, Right: 0, Top: 0, Bottom: 0
- Filters
    - Interlace Detection: Default
    - Deinterlace: Yadif, Preset: Default
- Video
    - Video Encoder: H.265
    - Framerate: 25, Constant Framerate
    - Constant Quality: 20
- Audio
    - Codec: AC3
    - Bitrate: 160
    - Mixdown: Stereo

Notes:

1. save these settings into a new preset
2. you might need to test some of your settings to achieve best results, specially the deinterlacing filters.
3. if you encode the movie for archiving I recommend to remove all filters, scaling, the cropping values, and use a video quality setting of 14 or better.

Simple Workflow

1. **Record VHS with *Media Express***
2. **Check quality of streams and decide whether you need to repair the video and stretch or repair the audio**
   Use *ffprobe* on the recorded video file and take a note on the length of the video stream (stream #1) and audio stream (stream #2). Depending on your VHS player and the condition of your tapes you will get dropped frames, cross talk, and other problems when recording. These usually show as an empty or distorted frame in the recorded video. You will also notice that the length of the video and audio streams differ, which is an indication of dropped frames, and takes the video streams out of sync.

   ```
   ffprobe -v error -show_entries stream=duration "title.mov"
   ```

    1. Use this Simple Workflow if the audio and video streams have the exact same length, or if you accept frames not recorded correctly will remain in the digital copy. This method requires limited effort and time and will result in a video file where audio and video are in sync, but faulty frames would not have been removed or repaired.
    2. Use the Sophisticated Workflow to achieve the best possible result. This method usually requires significant effort and time and will result in a video file where audio and video are in sync and faulty frames have been removed or repaired.
3. **Stretch audio**
    1. Check whether video and audio are in sync at several locations in the video file, particularly at the beginning and the end. Find locations with distinct audio, for example something drops to the floor, or where you can see the lips when a person starts talking.

If the audio is not the original audio of the movie (example: a movie originally spoken in English but synchronized to German) try to find a location where the words are the same, for example when a name is spoken (example: New York in English and in German sound similar and the words match length and lip movement). You can proceed to step 3.e. if:
    1. video and audio are in sync throughout the video file
    2. video and audio are off sync but with the same time difference throughout the video file (which we will fix with *MKVToolNix* later)
2. Encode the recorded audio/video with *HandBrakeGUI*. Use a low resolution codec which is fast, we only need the video to verify the sync and will discard it later. This allows us to run several *MKVToolNix* attempts to sync the audio to the video without long multiplexing times.
3. Import the recorded audio/video into *Audacity*.
4. Find the correct audio stretch. This is always a compromise, as the sync deviation will not be linear over the entire length of the video.
    1. Sometimes the audio starts playing advertisement or preview information right after the end credits of the movie. Remove this in *Audacity* first.
    2. Select **Effect → Change Tempo...**; 'Length from' should show the length of the audio stream reported from *ffprobe*, then change 'Length to' to the length of the video stream as reported from *ffprobe*. Hit 'OK' to do the conversion.
    3. Export the modified audio as lossless 16 bit FLAC file
    4. Use *MKVToolNix* to combine the encoded video stream (step a) with the newly created audio FLAC file. De-select the original audio, then multiplex.
    5. Load the newly created file into *HandBrakeGUI* und use **Preview** on different locations in the file to check video and audio are in sync.
    6. Repeat steps II-V until you are satisfied with the result.
5. If your audio is off sync throughout the video file with the same time difference enter a *Delay (in ms)* to the audio track in *MKVToolNix*. Negative numbers will play the audio sooner, positive numbers later relative to the video stream.
6. Use *MKVToolNix* to combine the original, lossless video stream with the newly created FLAC audio file. Enter 'Stereo' as track name, set language and delay for the audio track, then **Start multiplexing**. This will create a lossless master file for further processing with *HandBrake* in the steps below.
7. The resulting file might not be playable but *HandBrake* will be able to encode it.

4. Use *SMPlayer* on the original recorded video file to find start marker in seconds.milliseconds (sec.ms) and length of entire video in seconds.milliseconds (sec.ms)
5. Use *HandBrakeGUI* to find the correct cropping values (hit **Preview** then modify the cropping values)
6. **Re-encode with *HandBrakeCLI*** using a preset with Settings explained above, –start-at is the start position of the encode, –stop-at is the duration

```
HandBrakeCLI -i "H:\Blackmagic\title.mkv" -o "E:\Handbrake\new-
title.mkv" --preset-import-gui --preset "preset name" --audio 1 --
start-at sec.ms --stop-at sec.ms
```

7. **Set Chapters** or **Generate Chapters** in *MKVToolNix*
    1. Load “new-title.mkv” in *MKVToolNix*
    2. Open tab **Multiplexer / Output**
    3. Set Chapters
        1. Set Generating chapters: Don't generate chapters
        2. Chapter file: select a valid chapter xml file (
                                                example

) which should contain chapter names and position in the video
3. Language: set language of the chapter file

4. Generate Chapters (no set chapters)
   1. Set Generating chapters: Chapters in fixed intervals
   2. Set Interval: 00:05:00 or any other value you prefer
   3. Multiplex to your final movie file

Sophisticated Workflow

1. **Record VHS with *Media Express***
2. **Check quality of streams and decide whether you need to repair the video and stretch or repair the audio**
   Use *ffprobe* on the recorded video file and take a note on the length of the video stream (stream #1) and audio stream (stream #2). Depending on your VHS player and the condition of your tapes you will get dropped frames, cross talk, and other problems when recording. These usually show as an empty or distorted frame in the recorded video. You will also notice that the length of the video and audio streams differ, which is an indication of dropped frames, and takes the video streams out of sync.

   ```
   ffprobe -v error -show_entries stream=duration "title.mov"
   ```

   1. Use this Sophisticated Workflow to achieve the best possible result. This method usually requires significant effort and time and will result in a video file where audio and video are in sync and faulty frames have been removed or repaired.
   2. Use the Simple Workflow if the audio and video streams have the exact same length, or if you accept frames not recorded correctly will remain in the digital copy. This method requires limited effort and time and will result in a video file where audio and video are in sync, but faulty frames would not have been removed or repaired.
3. **Repair empty frames, artifacts and cross talk**
   This method is suitable if the length difference of video and audio streams is significant, or if there are dropped frames, artifacts or cross talk to repair in the video stream. You might also need to repair the audio (next step). In case you only find empty frames to repair there is a good chance that after removing those empty frames audio and video get in sync without any audio repair.
   1. Export the entire video to png image files (lossless)[3)]

      ```
      ffmpeg -i "title.mov" "\path\to\images\images%%d.png"
      ```

   2. Open the images folder and trim the video images by removing leading and trailing frames you don't want to include in the final video stream. It might also be necessary to check the audio stream in *Audacity* at the same time, so you don't cut behind the start of audio. Do this first so you do not need to repair parts of the video which will get cut anyway.
   3. Open the images folder, then select *View: Details*, then select the down arrow on the right side of *Size*, and select “Tiny”. This will isolate all images which are smaller than 16kb, which includes images that are black. These are frames which either were added or could not be read by the player, and contain very little color information, hence they are tiny. Black video content in the video might appear the same black, but contain much more color content and are not recognized as tiny.
   4. Now open all tiny images one by one and with *XnView* and move a few frames forward

and backward (PgDn and PgUp) to check whether the tiny images are surrounded by faulty frame images with cross talk or other errors.
5. Remove all tiny images.
6. Find cross talk and artifacts either from checking around tiny images or by visually inspecting all remaining images, then remove images with artifacts and add the same number of images to replace the removed ones. They are often next to the removed tiny images. Use XnView to open the first image, then press *Page-Down* to “play” the images. On a decent PC this will run as fast or faster than playing the video itself. Like in the video you will notice glitches or cross talk or artifacts, then stop “play” and remove the faulty images. Add the last good image and the next good image around the removal area by copying it and then renaming it into the image flow. If this is in a still image sequence (little movement, no visible speech) you can add more images, but don't do this in motion sequences or when the talking character is displayed, as this will break the flow.
7. You can also watch the repaired video before final authoring and take a note at the position in the video file, then go back to the images to find them.
8. **ADVANCED** If you find some images which contain artifacts you can repair them with *GIMP* (use **Filters → Enhance → Heal selection…**).
9. Finding cross talk and artifacts requires a lot of time and patience, if the video has a lot of locations which need to be repaired.
10. Afterwards you first need to rename the remaining files into sequence. Select all files in the folder, then right-click and select *Rename*, and enter a new “filename”. All files will be renamed to “filename (###).png”.
11. Convert the images back to a video file (lossless) with ***FFmpeg***[4] and[5]

```
ffmpeg -i "\path\to\images\filename (%%d).png" -vcodec ffv1 -level
1 -coder 1 -context 1 -g 1 "title.mkv"
```

4. **Repair audio**
   You might need to repair the audio if you repaired dropped or empty frames in the previous step. In rare cases, and if the repairs occurred in consistent time locations throughout the video file, you might be able to simply **stretch** the audio.
   1. Extract the audio stream with *MKVToolNix*.
   2. Trim the audio in *Audacity* to start and duration matching your video. For example, if you are working on a PAL video with 25 fps (frames per second), divide the number of frames you removed from the beginning of the video by 25 which gives you the seconds.milliseconds to remove from the start of the audio. Do the same for the length of the audio. Export the audio to 16 bit FLAC but do not close *Audacity*, so that you can undo the changes and apply again after adjustment. Combine audio and video in *MKVToolNix* and check whether the audio is in sync.
   3. If the audio is not in sync
      1. Define markers in the video where audio sync changes significantly. This is usually around areas where the amount of video to repair increases or decreases. Example: there are little errors in the video within the first 30 minutes, then there is a 10 minute block with heavy artifacts, then a 5 minute block with less artifacts, then rest of the video with very little errors in the video. In this example we would set the markers at 00:30:00, 00:40:00, and 00:45:00. The following steps will use these values.
      2. Check the audio sync delay with *SMPlayer* around position 00:30:00 and verify there is no significant delay, so you can leave this section unchanged.
      3. Find a position where audio “starts” around position 00:40:00, like speech or a notable audio pitch like a bell, a crash, or something like that. Note the exact

position in the video. You can press '.' or ',' in *SMPlayer* to move forward or backward by key frame.
4. Find the audio you identified in the step before around position 00:40:00 in *Audacity*, then select start 00:30:00 to end at the found position, for example select 00:30:00-00:40:12.350.
5. Select **Effect → Change Tempo...**; 'Length from' should show the duration you selected in the audio stream (00:30:00-00:40:12.350) in seconds (612.35), then change 'Length to' to the new duration for which you add or subtract sec.millisec as identified before. Hit 'OK' to do the conversion.
6. The audio after the repaired section will automatically time shift to the new end location of the stretched audio section.
7. Export the audio as 16 bit FLAC and combine it with the video in *MKVToolNix*. Do not close *Audacity* yet.
8. Repeat this for all section you identified the need to stretch the audio. The more sections you divide the audio in the more accurate your result will become, but also the more effort you have to put into repairing the audio track.
9. When you work through the sections in the audio I suggest to do this
    1. create a compressed copy of the video with *HandBrake*, which is much smaller than the lossless version and can be processed by *MKVToolNix* much faster
    2. open *MKVToolNix*, *SMPlayer*, *Audacity*, and the folder where all your files reside at the same time
    3. import the compressed video and the FLAC audio track into *MKVToolNix*
    4. stretch one section in *Audacity* (change tempo)
    5. export to FLAC and overwrite the file you already imported in *MKVToolNix*
    6. multiplex in *MKVToolNix*
    7. check the result in *SMPlayer*. You might need to make a fine correction to the change tempo duration in *Audacity*, then redo steps D-F.
    8. repeat steps D-G for all sections where audio needs to be repaired.
    9. make sure that the audio and video streams now have the same length, and if not make final corrections in *Audacity*.

5. **Create lossless master file**
    1. Once you are happy with the result import the lossless video and the fully repaired FLAC audio track into *MKVToolNix*. Enter 'Stereo' as track name, set language and (if necessary) delay for the audio track. Instead of the delay you can also modify the start sequence of the audio in *Audacity* (e.g. cut an additional 100 ms).
    2. Open tab **Multiplexer / Output** and set or generate chapters
        1. Set Chapters
            1. Set Generating chapters: Don't generate chapters
            2. Chapter file: select a valid chapter xml file (example) which should contain chapter names and position in the video
            3. Language: set language of the chapter file
        2. Generate Chapters (no set chapters)
            1. Set Generating chapters: Chapters in fixed intervals
            2. Set Interval: 00:05:00 or any other value you prefer
    3. **Start multiplexing** to a lossless master file
6. **Re-encode with *HandBrakeGUI*** using a preset with Settings explained above. Since we already trimmed audio and video we do not need to modify start marker and duration of the video. Find the correct cropping values before encoding (hit **Preview** then modify the cropping values).

Multiple part videos

1. If your video consists of more than one part or file repeat the steps described in the Simple or Sophisticated Workflow for each file and save them as “filename.cd1.mkv” and “filename.cd2.mkv”. If you are using Kodi to play the video it will automatically concatenate the two video files into one title.
2. Alternatively, concatenate the audio and video parts during encoding and make one video file out of it, following the Sophisticated Workflow.
    1. **Repair empty frames, artifacts and cross talk** as per step 3 above. Keep the image files of each part of the video in separate folders for now and omit step 3.k.
    2. **Repair audio** as per step 4 above for each part and export to a 16 bit FLAC file.
    3. **Combine the parts**
        1. Rename the images in the first folder with frame images to a-image (###), the images in the second folder to b-image (###), etc. as explained in step 3.j. then move all image files into the same folder.
        2. Select all files in the folder, then right-click and select *Rename*, and enter a new “filename”. All files will be renamed to “filename (###).png”. You now have all frame images of all parts of the video in sequence.
        3. Convert the images back to a video file (lossless) with ***FFmpeg*** (step 3.k.)

            ```
            ffmpeg -i "\path\to\images\filename (%%d).png" -vcodec ffv1 -
            level 1 -coder 1 -context 1 -g 1 "title.mkv"
            ```

        4. Load the repaired FLAC audio files of all parts into the same instance of *Audacity* and align them behind each other. For easier adjustment it is better to keep each audio file in a separate track.
        5. Export the audio again as 16 bit FLAC file and combine it with the newly encoded video file in *MKVToolNix*.
    4. Create the final movie with steps 5. **Create lossless master file** and 6. **Re-encode with HandBrakeGUI** as explained in the Sophisticated Workflow.

Result

This is an excerpt from a Cartoon Video I recorded and then repaired. At position 00:39:00 in the main movie there are a few seconds in the movie with damages, which I have removed. Notice the borders in the recorded movie which were removed with cropping in *HandBrakeGUI*, and the audio which was off sync during the entire movie. There were 18 empty frames which took the audio out of sync by about 720 ms by the end of the movie, but after having removed the empty frames there was no further audio repair necessary except the one to adjust to the removed frames at position 00:39:00.

The repaired movie has a resolution of 692×564, down from the recorded 720×576, and the quality is quite amazing considering the tape is a commercial original standard resolution VHS tape.

Recorded	Repaired

Note: if you open both videos separate them from the wiki either with the “picture-in-picture” selection arrow, or the full screen button. Due to an issue with the html5 video player function the video opened second gets distorted colors.

Links

- Prevent Clipping
- Profiles and Scene Collections
- How to transfer your VHS to Computer
- How to record high quality Wav audio in OBS ?
- Correct settings for capturing VHS
- 5 Best Ways to Deinterlace Video in 2022
- How To Properly De interlace Videos To 50 or 60 Fps Progressive
- HandBrake command line reference
- ffmpeg Documentation
- Use FFMPEG for Video to Frames, then Frames to Video with Original Sound
- FFMPEG- Convert video to images
- FFMPEG An Intermediate Guide/image sequence
- Syncing audio to video of a grabbed VHS video
- Lossless universal video format

Footnotes

1)

You need to check the correct cropping values for every tape, as they always differ. Open *HandBrakeGUI* and check and modify them using **Preview**, then save the preset with **Presets -> Update Current Preset**.

2)

Recording resolution is 720 x 576, set the scaled size to recording resolution minus the cropping values. The correct cropping values depend on your VHS player and the tape used. Some VHS Players do not deliver better quality than the original VHS resolution, in which case you can reduce the scaled size to 50% of the recording values minus cropping. The *HR-S6711EU* is a S-Video capable player which delivers much better quality through the s-video connection, in which case leave the scaled size at recording resolution minus cropped values.

3)

the double % are required by windows to escape the %

4)

get help with the following commands

```
ffmpeg -h
ffmpeg -encoders
ffmpeg -decoders
ffmpeg -h encoder=ffv1
```

5)

Links:
A quick guide to using FFmpeg to convert media files
FFmpeg Codecs Documentation
List of Lossless FFmpeg Video Encoders
FFV1 encoding cheatsheet