
video-compare

release **v20240623**

Split screen video comparison tool written in C++14 using FFmpeg libraries and SDL2.

`video-compare` can be used to visually compare, e.g., the effect of codecs and resizing algorithms on two video files played in sync. The tool is flexible, allowing videos of differing resolutions, frame rates, scanning methods, color formats, container formats, codecs, or durations.

A movable slider enables easy viewing of the difference across any region of interest.

Thanks to the versatility of FFmpeg, it is actually also possible to use `video-compare` to compare two images. The common PNG and JPEG formats have been successfully tested to work.

Installation

Arch Linux

Install via AUR:

```
1 git clone https://aur.archlinux.org/video-compare.git
2 cd video-compare
3 makepkg -sic
```

Homebrew

Install via Homebrew:

```
1 brew install video-compare
```

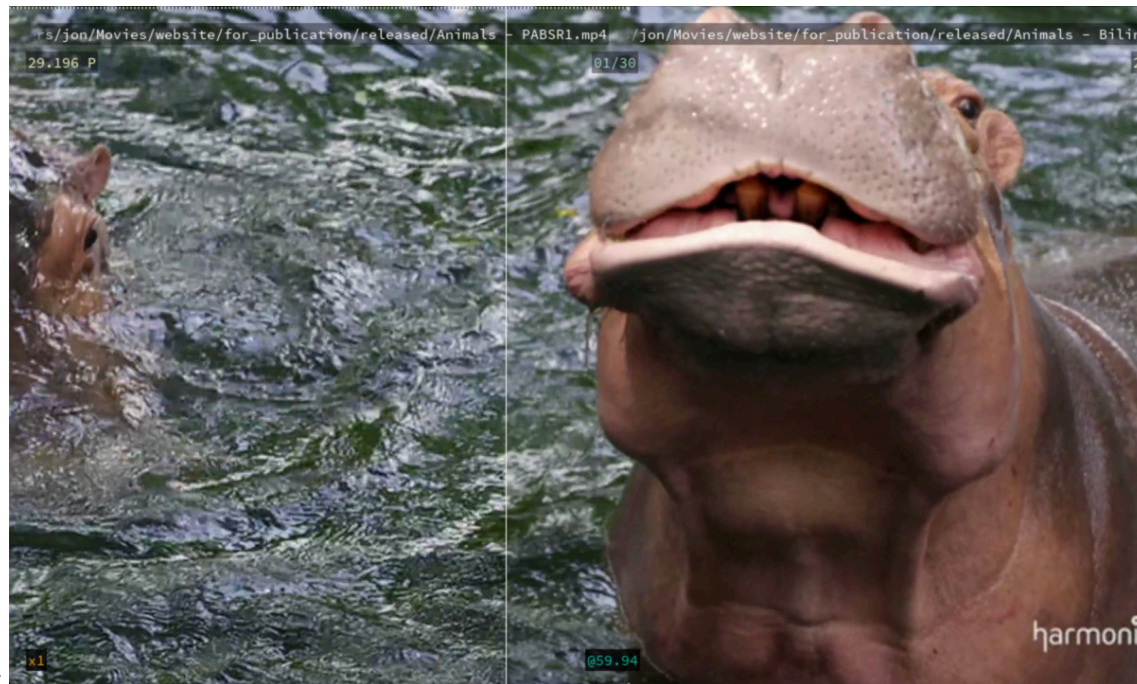
Pre-compiled Windows 10 binaries

Pre-built Windows 10 x86 64-bit releases are available from this page. Download and extract the .zip-archive on your system, then run `video-compare.exe` from a command prompt.

Compile from source

Build it yourself.

Screenshots



Visual compare mode:



Subtraction mode (plus time-shift, 200% zoom, and magnification):



Vertically stacked mode:

Credits

`video-compare` was created by Jon Frydensbjerg (email: jon@pixop.com). The code is mainly based on the excellent video player GitHub project: <https://github.com/pockethook/player>

Many thanks to the FFmpeg, SDL2 and stb authors.

Usage

Launch using the operating system's DPI setting. Video pixels are doubled on devices like a Retina 5K display; therefore, it is the preferred option for displaying HD 1080p videos on such screens:

```
1 video-compare video1.mp4 video2.mp4
```

Allow high DPI mode on systems which supports that. Video pixels are displayed "1-to-1". Useful for e.g. displaying UHD 4K video on a Retina 5K display:

```
1 video-compare -d video1.mp4 video2.mp4
```

Increase bit depth to 10 bits per color component (8 bits is the default). Fidelity is increased while performance takes a hit. Significantly reduces visible banding on systems with a higher grade display and driver support for 30-bit color:

```
1 video-compare -b video1.mp4 video2.mp4
```

Use a specific window size instead of deriving the window size from the video dimensions. The video frame will be scaled to fit. If either width or height is left out, the missing value will be calculated from the other specified dimension so that aspect ratio is maintained. Useful for downscaling high resolution video onto a low resolution display:

```
1 video-compare -w 1280x720 video1.mp4 video2.mp4
```

Automatic in-buffer loop playback, triggered when the buffer fills or end-of-file is reached, streamlines video analysis by eliminating the need for manual replay initiation (bidirectional "ping-pong" mode, `pp`, is also available):

```
1 video-compare -a on video1.mp4 video2.mp4
```

Shift the presentation time stamps of the right video instead of assuming the videos are aligned. A positive amount has the effect of delaying the left video while negative values conversely delays the right video. Useful when videos are slightly out of sync:

```
1 video-compare -t 0.080 video1.mp4 video2.mp4
```

Display videos stacked vertically at full size without a slider (`hstack` for horizontal stacking is also supported):

```
1 video-compare -m vstack video1.mp4 video2.mp4
```

Preprocess one or both inputs via a list of FFmpeg video filters specified on the command line (see FFmpeg's video filters documentation). The Swiss Army knife for cropping/padding (comparing

videos with different aspect ratios), adjusting colors, deinterlacing, denoising, speeding up/slowing down, etc.:

```
1 video-compare -l crop=iw:ih-240 -r format=gray,pad=iw+320:ih:160:0
  video1.mp4 video2.mp4
```

Select a demuxer that cannot be auto-detected (such as VapourSynth):

```
1 video-compare --left-demuxer vapoursynth script.vpy video.mp4
```

Explicit decoder selection for the right video:

```
1 video-compare --right-decoder h264_cuvid video1.mp4 video2.mp4
```

Set the hardware acceleration type for the left video:

```
1 video-compare --left-hwaccel videotoolbox video1.mp4 video2.mp4
```

Perform simpler comparison of a video with itself using double underscore (__) as a placeholder. This enables tasks such as comparing the video with a time-shifted version of itself or testing various sets of filters, without the need to enter the same, potentially long path twice:

```
1 video-compare some/very/long/and/complicated/video/path.mp4 __
```

The above features can be combined in any order, of course. Launch `video-compare` without any arguments to see all supported options.

Controls

- H: Toggle on-screen help text for controls
- Space: Toggle play/pause
- Comma ,: Toggle bidirectional in-buffer loop/pause
- Period .: Toggle forward-only in-buffer loop/pause
- Escape: Quit
- Down arrow: Seek 15 seconds backward
- Left arrow: Seek 1 second backward
- Page down: Seek 600 seconds backward
- Up arrow: Seek 15 seconds forward
- Right arrow: Seek 1 second forward
- Page up: Seek 600 seconds forward
- J: Reduce playback speed
- L: Increase playback speed
- S: Swap left and right video

-
- A: Move to the previous frame in the buffer
 - D: Move to the next frame in the buffer
 - F: Save both frames as PNG images in the current directory
 - P: Print mouse position and pixel value under cursor to console
 - Z: Magnify area around cursor (result shown in lower left corner)
 - C: Magnify area around cursor (result shown in lower right corner)
 - R: Re-center and reset zoom to 100% (x1)
 - 1: Toggle hide/show left video
 - 2: Toggle hide/show right video
 - 3: Toggle hide/show HUD
 - 5: Zoom 50% (x0.5)
 - 6: Zoom 100% (x1)
 - 7: Zoom 200% (x2)
 - 8: Zoom 400% (x4)
 - 0: Toggle video/subtraction mode
 - Plus +: Time-shift right video 1 frame forward
 - Minus -: Time-shift right video 1 frame backward

Move the mouse horizontally to adjust the movable slider position.

Use the mouse wheel to zoom in/out on the pixel under the cursor. Pan the view by moving the mouse while holding down the right button.

Left-click the mouse to perform a time seek based on the horizontal position of the mouse cursor relative to the window width (the target position is shown in the lower right corner).

Hold the SHIFT key while pressing **D** to decode and move to the next frame.

Hold CTRL while time-shifting with +/- for faster increments/decrements of 10 frames per keystroke. Similarly, hold down the ALT key for even bigger time-shifts of 100 frames.

Build

Requirements

Requires FFmpeg headers and development libraries to be installed, along with SDL2 and its TrueType font rendering add on (libSDL2_ttf). SDL2 version 2.0.10 or later is now specifically required for subpixel accuracy rendering capabilities. Users may need to upgrade their existing SDL2 installation before compiling.

On Debian GNU/Linux the required development packages can be installed via [apt](#):

```
1 apt install libavformat-dev libavcodec-dev libavfilter-dev libavutil-  
dev libswscale-dev libswresample-dev libSDL2-dev libSDL2-ttf-dev
```

Instructions

Compile the source code via GNU Make:

```
1 make
```

The linked `video-compare` executable will be created in the source code directory. To perform a system wide installation:

```
1 make install
```

Note that root privileges are required to perform this operation in most environments (hint: use e.g. `sudo`).

Notes

1. Audio playback is not supported.
2. Keep time-shifts below a few seconds for the best experience.
3. Seeks require re-synchronization on the closest keyframe (i.e., I-frame).

Practical tips

Send To integration in Windows File Explorer

You can fire up the tool directly from the File Explorer when you don't need to specify any other arguments than the inputs via Right click -> Send To -> video-compare.

Here is how this integration works:

<https://user-images.githubusercontent.com/8549626/166630445-c8c511b7-005f-48aa-83bc-0eb9676cfa2a.mp4>

You can do that quickly by selecting two files, then right clicking any of them, pressing N (focuses **send to**), then V (selects **video-compare**).

To get video-compare to appear in the **Send To** field you will need to open the **send to** folder, which you can access by typing `shell:sendto` in Run (Windows + R), then simply make a shortcut to `video-compare.exe`.

Thanks to couleurm for the sharing this tip and creating the screen recording above.