
About

BF is a video manipulation tool that can:

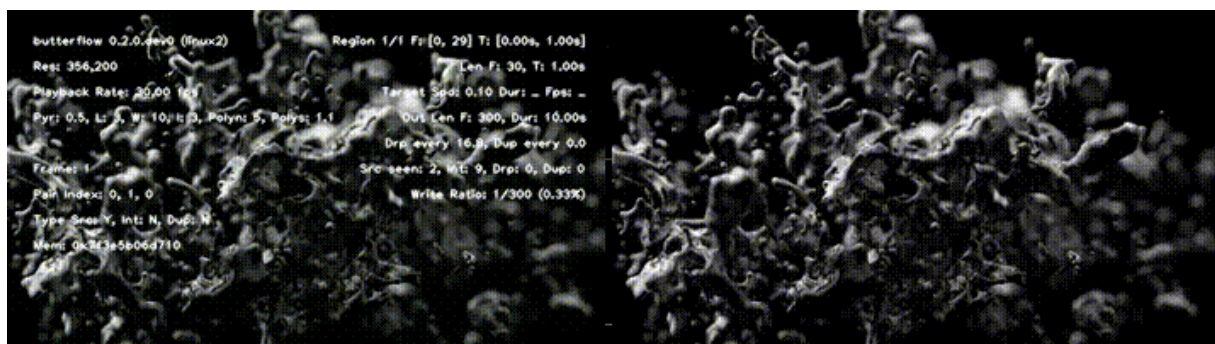
1. Make **motion interpolated videos** (increase a video's frame rate by rendering intermediate frames based on motion using a combination of pixel-warping and blending).
2. Make **smooth motion videos** (do simple blending between frames).
3. Leverage interpolated frames to make **fluid slow motion videos**.

Demonstration

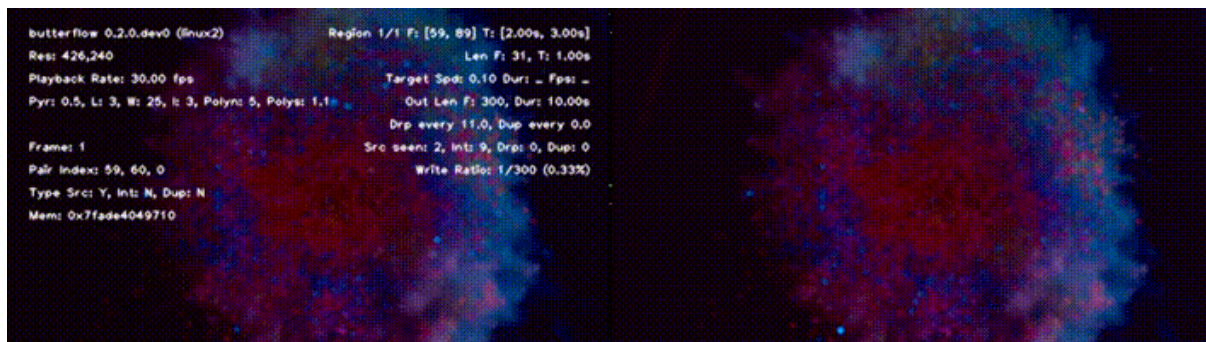
BF works by rendering intermediate frames between existing frames using a process called motion interpolation. Given two existing frames, **A** and **B**, this program can generate frames **C . 1**, **C . 2...C . n** positioned between the two. In contrast to other tools that can only *blend or dupe* frames, this program *warps pixels based on motion* to generate new ones.

The addition of interpolated frames gives the perception of more fluid animation commonly found in high frame rate videos, an effect most people know as the “soap opera effect”.

Besides creating motion interpolated videos BF can leverage interpolated frames to make fluid slow motion videos:



In these examples BF slowed a 1sec video down by 10x. An additional 270 frames were interpolated from 30 original source frames giving the video a smooth feel during playback. The same video was slowed down using FFmpeg alone, but because it dupes frames and can't interpolate new ones the video has a noticeable stutter (shown on the right-hand side).



Install

Requirements: A 64-bit system with a compatible graphics device.

- **Windows 10 (Portable):** Download the latest releases.
 - **Preview:** butterflow-0.2.4a4-win64.zip
 - ★ Sha256: 856ad05bafaeb9d084d6b1cbf0bdec17defeef438e610f6a3a772087acea301b
 - **Stable:** butterflow-0.2.3-win64.zip
 - ★ Sha256: a77fbbdbdd0d85bb31feac30e53378a36ff4fb2a8a98ba15a6fa06def4b36ad1
- **macOS and Linux:** See the Install From Source Guide for instructions.

Setup

BF requires no setup to use but it's too slow out of the box to do any serious work. BF can take advantage of hardware accelerated methods to make rendering significantly faster, but this requires that you have a functional OpenCL environment on your machine.

You can check if you have a compatible OpenCL device by running `butterflow -d`. If BF detects your device no other set up is necessary, otherwise continue reading below.

Windows

First make sure that you have the latest version of your graphics driver installed and then try running `butterflow -d` again. If BF still fails to detect your device, try specifying the location of your hardware's OpenCL client driver in the registry at `HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors` by adding a key with the full path to the DLL as a `REG_DWORD` type with a data value of 0. NVIDIA users should look for `nvopenc164.dll`, AMD: `amdocl64.dll`, and Intel: `IntelOpenCL64.dll` by searching system folders, then add the keys with Registry Editor.

You may still need to add 32-bit keys even if you're on a 64-bit system. Do the same for 32-bit drivers `nvopengl32.dll`, `amdocl32.dll`, ..., at `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Khronos\OpenCL\Vendors`.

Example for NVIDIA:

```
1 [HKEY_LOCAL_MACHINE\SOFTWARE\Khronos\OpenCL\Vendors]
2 "C:\Windows\System32\nvopengl64.dll"=dword:00000000
3
4 [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Khronos\OpenCL\Vendors]
5 "C:\Windows\System32\nvopengl32.dll"=dword:00000000
```

When finished, run `butterflow -d` again to check if your device is detected.

macOS and Linux

No setup on macOS is necessary because Apple provides OpenCL support by default on all newer Macs. If you're on Linux, please seek other sources on how to satisfy the OpenCL requirement.

Usage

Run `butterflow -h` for a full list of options. See: Example Usage for typical commands.

License

MIT License