Decisions

Русский · About the program · User guide · Console command

Contents

- One program, the neural part by the machine
- No frame interpolation
- The window is the face of the console command
- The picture decides
- The newest VapourSynth, plugins of its interface
- BM3D on any video card
- The denoisers measure the noise
- The levels are kept
- The same file gives the same frames
- Nothing beside the person's file
- A file is whole or it is not there
- A folder is taken whole
- A wrong value is a wrong call
- Inverse telecine refuses a slow source
- FFmpeg for the driver
- PROCESSING where it moves something
- The quality ladder
- Sound

Why the program is the way it is.

One program, the neural part by the machine

There is one program. The three presets on vs-mlrt — DPIR, Real-ESRGAN 2x, Soft HD rebuild 2x — are shown and run by what the machine has, not by a build: an NVIDIA RTX card (`nvidia-smi` names it)

offers the installation of MLRT, and installed MLRT runs them. Without a card and without MLRT they are not in the window: a preset that can never run here would only be a promise. MLRT installed on a machine with RTX keeps working after the folder moves: the preset itself picks TensorRT, DirectML or the processor.

MLRT is installed trimmed to the card (the TensorRT resources of its architecture, the models a preset uses), and its downloads — about 3.5 GB — do not stay in the folder once the check of the backends has passed.

No frame interpolation

RIFE — the network that builds frames between frames (24 into 48 or 60 a second) — is not in the program: none of the ways open to it today gives right frames and a bearable speed at once.

- **TensorRT**, the only fast way of vs-mlrt on an NVIDIA card, builds wrong frames between the frames on an RTX 5070 (the Blackwell architecture, as every RTX 50): a mess instead of a picture with TensorRT 10.16, TensorRT 11.1 and TensorRT-RTX 1.5, in fp16 and in fp32, and in a bare call of `vsmlrt.RIFE` with no preset around it. The frames of the source come through whole: the fault is in what TensorRT builds of this network for this architecture, not in the script. Cards of the earlier generations (RTX 20–40) were not at hand to check.
- **DirectML** through ONNX Runtime gives right frames, but the ONNX Runtime plugin of vs-mlrt (`vsort`) is built only for the old plugin interface, API 3, which VapourSynth R80+ does not load. The builds for the new interface carry TensorRT, TensorRT-RTX, OpenVINO and NCNN.
- **NCNN** on Vulkan crashes on this network ("Unsupported split axis", an access violation).
- **OpenVINO on the processor** gives right frames, slowly: on a Ryzen 9 5900X (12 cores) a new 4K frame takes 1.28 s, a 1080p frame 0.30 s. A minute of 24p doubled to 48p is 1,440 new frames: about 31 minutes in 4K and 7 in 1080p. 24 into 60 is 2,880 new frames, no less than an hour for a minute of 4K.

A fractional multiplier (24 into 60 is 2.5) vs-mlrt counts through the akarin plugin besides, and it exists only for API 3. RIFE comes back when a way on the video card gives right frames: TensorRT fixed for Blackwell, or ONNX Runtime built for the new plugin interface.

The window is the face of the console command

Every button runs an operation of `vs_cli.cmd`, and the first line of every run in the journal is that command. What a form allows — which encoder a card offers, which choice is off and why — is

decided once, in the engine, for the window and the console alike. So a batch file, an AI agent and a person at the window get the same result, and the window cannot do anything the command cannot.

The picture decides

A preset is chosen by eye, so its frame stands under its form. The preview takes the middle frame of a video — the start of a clip is often black or a title — and counts it with the same script and the same values a run takes, by the console command, never by a separate approximation. It is counted when the form has been left alone for a moment, not at every movement of a slider, and a count that a newer change made useless is ended at once: a heavy preset on a 4K frame takes seconds, and the person should not wait for frames nobody looks at.

The newest VapourSynth, plugins of its interface

VapourSynth R80 dropped the old plugin interface (API 3), and the catalogue still offers many plugins only in it. Staying on an old VapourSynth would freeze the program on a version that no longer gets fixes, so the program installs the newest stable VapourSynth and a set of plugins built for its interface (`install\vs_plugins.json`): from PyPI, from the releases of their authors on GitHub, and from vsrepo. The set does not move the engine. Whether the engine really loads every plugin is read from the engine's own log of loading, because a refused plugin is silent to Python. The engine can be removed to the Recycle Bin and installed again.

BM3D on any video card

BM3D runs through Vulkan (`bm3dvk`): an NVIDIA, AMD or Intel card with its ordinary driver, no CUDA Toolkit. Without a Vulkan device it runs on the processor. On the processor two runs give the same frames byte for byte; on a video card they agree within about 65 dB — it sums in a varying order. PROCESSING names one of them.

The denoisers measure the noise

A denoiser tuned for one level of noise smears clean footage or leaves heavy noise in place. Seven denoisers measure the noise of the source before they start: on flat areas of five frames spread over the clip, luma and chroma each their own. Their settings are made for about 2 codes of noise; more raises them — BM3D's sigma and MVTools' thresholds with the noise, DFTTest's sigma with its square — and less never lowers them, so clean footage gets exactly the settings. MVTools' threshold of a

scene change rises with the noise too: on heavy noise every block exceeds the default, and the whole clip would pass as new scenes, unfiltered. `noise=fixed` takes the settings as they are.

The levels are kept

A preset does not change the range of its source. The output carries the range of the source in its frames; a preset that goes through RGB converts back into the same range; FFmpeg is told the range on its input and on its output — on the output alone it would take the untagged input for limited and stretch the levels.

The same file gives the same frames

The grain has a fixed seed, so a render of a look can be compared with another render of it, and a frame of the preview is the frame the run will give.

Nothing beside the person's file

Indexes of the sources (`.lwi`, `.ffindex`) are written into the program's folder, not beside the source: the loaders write them next to the file by default, into the person's folder. The engine, its plugins, MLRT, the caches of the installers live inside the program too; nothing goes into Windows, the registry or PATH.

A file is whole or it is not there

A result is written under a name of its own and takes its name only when complete; STOP ends vspipe and FFmpeg and removes the half-written file. A result already there is passed over rather than written again, so a stopped run over a folder simply starts again and goes on where it stopped; *Overwrite* makes everything anew.

A folder is taken whole

The Source is a file or a folder with every video of its subfolders, and the result keeps those folders under the folder of the palette and of the preset with its encoder. A folder of results names how they were made; two presets never write into one folder.

A wrong value is a wrong call

A number outside its range or a choice that is none of its field's is refused with the reason, not moved quietly to the nearest allowed value: a run with other values than asked is worse than no run.

Inverse telecine refuses a slow source

TIVTC removes the frames that 3:2 pulldown added. On a source under 29 fps there are none, and it would throw away every fifth real frame; the preset refuses such a source.

FFmpeg for the driver

The FFmpeg build is chosen for the NVIDIA driver: a build newer than the driver supports breaks hardware encoding instead of speeding it up. The 610 series of drivers has settled and 610.88 is its LTS: with it, and without an NVIDIA card, the newest stable FFmpeg is installed — never older than 9.0.2, the build the programs are checked with; a newer one that fails the NVENC test goes back to 9.0.2. Below 610.88 an older branch keeps NVENC working, and the program says that not every operation is possible.

PROCESSING where it moves something

The block PROCESSING stands only on the presets where it changes something: the BM3D presets and the neural ones. The other filters always run on the processor, and a choice that does nothing would read as a choice that matters.

The quality ladder

Every encoder has the same three steps — 14, 17, 21 — software and hardware alike: distinguishable steps without overlap, from a near-lossless master to a preview.

Sound

Only the picture is processed. The sound is copied; a ProRes source going into x264 or x265 gets AAC 384 kbps, because its PCM would not fit those containers; MXF gets PCM 24 bit at 48 kHz, the only sound its muxer writes. The program has no hardware decoding: the source loaders of VapourSynth decode on the processor whatever they are asked.