**20230811 – Release 2.7.46-e.02**

11 Aug 18:00 [DTL2020](https://github.com/DTL2020) [r.2.7.46-e.02](https://github.com/DTL2020/mvtools/tree/r.2.7.46-e.02)  [4e87fa2](https://github.com/DTL2020/mvtools/commit/4e87fa29255e172b5f4ea37dedbd34cc2c0ce4f2) [Release 2.7.46-e.02](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-e.02) [Latest](https://github.com/DTL2020/mvtools/releases/latest)

Small bugfix for Auto-thSAD for bitdepth > 8.

**20230811 – Release 2.7.46\_e.01 (small features extracted)**

11 Aug 10:12 [DTL2020](https://github.com/DTL2020) [r.2.7.46-e.01](https://github.com/DTL2020/mvtools/tree/r.2.7.46-e.01) [a5d25eb](https://github.com/DTL2020/mvtools/commit/a5d25eba3d6d2ed07da90222494ee5379c661f34)

[Release 2.7.46\_e.01 (small features extracted)](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-e.01)

Added **SuperCurrent** input to MAnalyse to feed prefiltered super clip at prefiltering or multi-generation MVs refining use cases.

Added Auto-thSAD for MDegrainN.

New params to MDegrainN:  
**thSADA\_a** (float), default = 0. Multiplier proportional to estimated nosie level  
**thSADA\_b** (float), default = 0. Offset to calculated Auto-thSAD.

If both thSADA\_a and thSADA\_b = 0 - Auto-thSAD is disabled.

Used Auto-thSAD is scaled and offsetted arithmetic mean of blocks SAD values below thSCD1 (noise\_estimate). Next is applied adjusting params:  
Auto\_thSAD = thSADA\_a \* noise\_estimate + thSADA\_b

thSAD2, thSADC, thSADC2 calculated proportionally to provided old params values.

Version based on pinterf last commits from 2021 after 2.7.45 release and expected to be stable enough.

**20230730 – Release 30.07.2023**

30 Jul 18:22 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.24](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.24) [71d50fb](https://github.com/DTL2020/mvtools/commit/71d50fbad2f4e00d6593d426490cb867163a84bd) [Release 30.07.2023](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.24)

Added Auto-thSAD for MDegrainN.

New params to MDegrainN:  
**thSADA\_a** (float), default = 0. Multiplier proportional to estimated noise level  
**thSADA\_b** (float), default = 0. Offset to calculated Auto-thSAD.

If both **thSADA\_a** and **thSADA\_b** = 0 - Auto-thSAD is disabled.

Used Auto-thSAD is scaled and offsetted arithmetic mean of blocks SAD values below **thSCD1** (noise\_estimate). Next is applied adjusting params:  
Auto\_thSAD = **thSADA\_a** \* noise\_estimate + **thSADA\_b**

**thSAD2**, **thSADC**, **thSADC2** calculated proportionally to provided old params values.

**20230617 – Release 17.06.2023**

17 Jun 20:17 [DTL2020](https://github.com/DTL2020) r.2.7.46-a.23 [f4b30d1](https://github.com/DTL2020/mvtools/commit/f4b30d1ea2d9a7e3ec13808beb68c7dbf349ae23) [Release 17.06.2023](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.23)

Added denoise mask clip input into MDegrainN. Work only on block-based mode. Must be Y8 format with frame size equal to blocks number to process (including any used overlap mode).

New param to MDegrainN:  
**dnmask** - clip. 0 is full standard denoise, 255 is no denoise (so positive Y-channel can be used as mask to degrain only low brightness levels).

Example script (for **IntOvlp=3**):  
dn\_mask1=ConvertToY8()

blksize=8  
#int\_ ovlp=3  
dn\_mask\_x=dn\_mask1.width/blksize  
overlap\_size=blksize/2  
dn\_mask\_y=(dn\_mask1.height-overlap\_size)/(blksize-overlap\_size)  
dn\_mask1=BilinearResize(dn\_mask1, dn\_mask\_x, dn\_mask\_y)

dn\_mask1=Levels(dn\_mask1, 0, 1, 100, 0, 255, coring=false)

dn\_masked=MDegrainN(.., **IntOvlp=3**, **dnmask**=dn\_mask1)

Added update MEL memory with best (lowest sum of DM table row) block and memory for sum of current stored in IIR memory block.

**20230330 – Release 30.03.2023**

30 Mar 10:00 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.22](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.22) [0a6b093](https://github.com/DTL2020/mvtools/commit/0a6b093507bc757457783d537bc6cfaaa273989d) [Release 30.03.2023](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.22) Pre-release

Added computing and displaying of residual noise bits count per frame to MCompensate.  
Compute sum of log2 of the samples absolute difference between source and motion compensated output frame of MCompensate. For complete static frame sequence RNB=0 bits/frame. For noise bitrate per second - value should be multiplied to frame rate.

New param to MCompensate (bool): **showRNB** (default = false).

Usage example:  
super=MSuper()  
mv=MAnalyse(super)  
MCompensate(super, mv, showRNB=true)

**20230315 – Release 15.03.23**

15 Mar 17:29 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.21](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.21) [a64d653](https://github.com/DTL2020/mvtools/commit/a64d653b48151e995ba6caf255bac69502f4318f) [Release 15.03.23](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.21) Pre-release

Added new processing mode to MDegrainN: MEL (Most Equal Looking) search mode for TTH (Temporal Thresholding).  
New params to MDegrainN:

**pmode=0** - standard blending, **pmode=1** - MEL search and TTH only.

**TTH\_DMFlags** - dismetric flags for estimating blocks difference at TTH compare. Flags 0x1 to 0x20 valid.

**TTH\_thUPD** - integer threshold for selection: keep output old block from memory or update block in memory and output new block.

**TTH\_chroma** - use chroma in TTH dismetric analysis (slower, better quality) or not (faster).

Fixed performance issue with double processing of chroma planes in combined YUV processing with no overlap.

**20220308 – Release 08.03.2023**

08 Mar 19:15 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.20](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.20) [9141a08](https://github.com/DTL2020/mvtools/commit/9141a0838c82f474f4ae89c49e38e48a6cd5a0d0) [Release 08.03.2023](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.20) Pre-release

Fixed possible bug with **trymany** in MAnalyse.  
Added **trymany** into **optPredictorType=1** mode (zero, global and median predictors only).

Added partial fix for 4:2:x formats processing chroma shift issue for MAnalyse, MDegrainN, MCompensate (may also MRecalculate). With the curernt pel-precision from MSuper.

**20230308 – Fix02 for 4:2:x formats**

08 Mar 10:27 [DTL2020](https://github.com/DTL2020) [r.2.7.45-fix02](https://github.com/DTL2020/mvtools/tree/r.2.7.45-fix02) [fa9eb74](https://github.com/DTL2020/mvtools/commit/fa9eb74afb28a8d9ca48ba2ec48befd608bab0e5) [Fix02 for 4:2:x formats](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.45-fix02) Pre-release

Uses sources from 2.7.45 pinterf branch. Fix partial chroma shift for chroma-subsampled formats (MAnalyse, may be MRecalculate, and MDegrainX/N and MCompensate). Fix error from fix1 with MAnalyse for 4:2:2 sources.

**20230307 – Fix for 4:2:x formats**

07 Mar 19:36 [DTL2020](https://github.com/DTL2020) [r.2.7.45-fix1](https://github.com/DTL2020/mvtools/tree/r.2.7.45-fix1) [26105ad](https://github.com/DTL2020/mvtools/commit/26105adff31020d537ebd4c70fda594fb2a1731c) [Fix for 4:2:x formats](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.45-fix1) Pre-release

Uses sources from 2.7.45 pinterf branch. Fix partial chroma shift for chroma-subsampled formats (MAnalyse, may be MRecalculate, and MDegrainX/N).

**20221024 – Release 24.10.22**

24 Oct 19:20 [DTL2020](https://github.com/DTL2020) [7fdb122](https://github.com/DTL2020/mvtools/commit/7fdb122d05d7378386e315a558f5786f84499783) [Release 24.10.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.19) Pre-release

Added **MPB\_MVlth** param to MDegrainN. Limit allowed length of MV for weight correctionby MPB processing. Can decrease possible 'ghosting' with too extreme **MPB\_SPCsub**/**add** params values. Not scaled by **pel** currently. Recommended values - about 2..3 squared **pel** value. Valid working range from 0 to squared frame size (unlimited). Zero may be disable MPB weight adjusting completely.

Added reduced **tr** blending mode for MPB controlled currently by **MPB\_SPCadd** > 10.

Added **MPBtgtTR** param to MDegrainN:  
In standard MPB mode controls initial number of weights used for calculate initial blend estimation (may be 0 - only current block used).  
Valid range - from 0 to **tr**.  
In reduced weights MPB mode - controls number of ref frames (2 \* **tr**) used for blending without any other weights adjustment by MPB.

Added **MPB\_DMFlags=64** flag. Uses covariance metric only. Can be used only with MDegrainN.

Added VIF (DWT- based) metric. Controlled by 0x10 (VIF-Approximation) and 0x20 (VIF-Edges) flags. 16 and 32 decimal. Full VIF is 16+32=48. Can be used in both MDegrainN MPB flags and MAnalyse.

**20221009 – Release 09.10.22**

09 Oct 17:23 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.18](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.18) [4b92af1](https://github.com/DTL2020/mvtools/commit/4b92af10d6e1bf281b3d32a1535fc7276428aec2) [Release 09.10.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.18) Pre-release

Added SSIM metric in MAnalyse and MPB MDegrainN processing. Now possible dissimilarity metric flags (as bit mask):  
bit 0 - SAD,  
bit 1 - SSIM luma only,  
bit 2 - SSIM contrast and structure.  
Examples:

1. SAD only = 1
2. SSIM luma only = 2
3. SSIM contrast and structure = 4
4. Full (standard) SSIM = 6
5. SAD + SSIM contrast and structure = 5

Selecting of dissimilarity metric supported only in part of **optSearchOptions** and **optPredictorType** of MAnalyse.

New param for MDegrainN - **MPB\_DMFlags**. Integer any of dissimilarity metric bitmask, default=1.  
New param for MAnalyse - **DMFlags** Integer any of dissimilarity metric bitmask, default=1.  
New param for MRecalculate - **DMFlags** Integer any of dissimilarity metric bitmask, default=1.

Current release have only C-reference SSIM calculation functions so very slow. For quality check mostly.

**20221006 – Release 06.10.22**

06 Oct 17:52 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.17](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.17) [a5f27e6](https://github.com/DTL2020/mvtools/commit/a5f27e64d4f63608b89035c1722f03955fa9973c) [Release 06.10.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.17) Pre-release

Added **MPB\_PartBlend** param to MDegrainN to check real partial blend vs subtraction of block. Default false (use subtraction). If set to **true** - perform full block blending with test block removed (slower but a bit more accurate in SAD).  
Separated subtractive and additive coefficients to different params **MPB\_SPCsub** and **MPB\_SPCadd** for better flexibility at experiments of finetuning.  
Added isMVsStable function to check if MVs in current tr-pool for current block are enough coherent (stable) - to try to make MPB processing only at areas with stable enough motion search in tr-pool of frames.

New MDegrainN param **MPBthIVS** - threshold to compare current calculated measure of non-stabilily of MVs (sum of accelerations multiplied to sum of vectors angle difference). Param is internally scaled to squared pel value but may significantly depend on **tr** and other settings. To help adjust this threshold - use IVS-mask display with **showIVSmask=true**.

Added protection to MDegrainN against too low padding (now need to be at least blocksize in size) and not equal temporal radius param for MAnalyse and MDegrain - display error messages instead of corrupted output.

Added **showIVSmask** param to MDegrainN to mark blocks detected as stable enough MVs with black. Default false. Black blocks with are detected as ready for MPB processing.

Added **mvmultivs** param to MDegrainN as option to provide separate MVclip with different search source or options for IVS mask creating. Provided clip must be equal to **mvmulti** in block number, overlap mode and recommended to use **truemotion=false** preset of MAnalyse to show noise-moved blocks as best as possible, not recommended to make from prefiltered clip. **mvmulti** clip may use any required params for best denoising, can be created from prefiltered source and so on.

**20220924 – Release 24.09.22**

24 Sep 18:30 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.16](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.16) [f85b1d3](https://github.com/DTL2020/mvtools/commit/f85b1d335951f9ef800f42c242a0c2d3b0e9c4aa) [Release 24.09.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.16) Pre-release

Added **SuperCurrent** param to MAnalyse, clip param Allow to provide differently processed clip as current source for search. May be useful to use with prefiltering use cases.

Added **SearchDirMode** param to MAnalyse, int param.   
0 - search standard direction (current frame to ref frame MVs).   
1 - reverse search.

Fixed bug with not-selecting combined luma+chroma processing modes when thSADC=thSAD (and thSADC2=thSAD2).

Added Multi-Pass Blending mode in MDegrainN. New params:  
MPBthSub, int (10), threshold for subtracted blocks.  
MPBthAdd, int (20), threshold for std blended blocks (additively).  
MPBNumIt, int (0), number of iterations. 0 - MPB processing mode not used.  
MPB\_SPC, float (1.5), multiplier and divider for weight adjustment at each iteration if SAD of curent blending result vs subtracted or ref block is above threshold.

**20220803 – Release 03.08.22**

03 Aug 08:32 [DTL2020](https://github.com/DTL2020) [d035ba2](https://github.com/DTL2020/mvtools/commit/d035ba2ee089ab593bcbe2a6bf21b728f38fbae6) [Release 03.08.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.15) Pre-release

Added diagonal interpolated overlap mode to MDegrainN of 2x blocks number to process. **IntOvlp=3** with SAD re-check and **IntOvlp=4** with interpolated SAD.  
Added more error messages if non-compatible options provided for MSuper/MAnalyse/MDegrainN.  
Updated documentation with new options.  
Added meander scan in the combined luma+chroma overlapped processing - may be better reuse of cached ref planes data.

**20220726 – Release 26.07.22**

26 Jul 17:45 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.14](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.14) [281ac78](https://github.com/DTL2020/mvtools/commit/281ac78337e70078329584cf37792a0a17f2d2b9) [Release 26.07.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.14) Pre-release

Added mode **2** for **IntOvlp** for MDegrainN: It do not check real SAD of the interpolated blocks positions. So it is faster but may be lower in quality.

Fixed buffer overrun bug in InterpolateOverlap in MDegrainN.

Added AVX2 (8 bit output), SSE2 and SSE4 ( >8 bit output) second pass processing to output format into MDegrainN.

Disabled loading of shader file Compute.cso in **optSearchOption=6** mode of MAnalyse.

Added different builds - for Win10 and later with DX12, for Win7 and others without DX12. Also some IntelC++ builds available for AVX2 CPUs.

**20220720 – Release 20.07.22**

20 Jul 12:30 [DTL2020](https://github.com/DTL2020) [1e85f89](https://github.com/DTL2020/mvtools/commit/1e85f89e78485f9886374ddbd5af004c79ffdd60) [Release 20.07.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.13) Pre-release

Added interpolated overlap mode to MDegrainN. Only 'max' blocksize/2 mode currently implemented.

New param of MDegrainN: **IntOvlp** (int).  
Values:  
0 - standard mode (default).  
1 - internally interpolate input MVs to blocksize/2 overlap mode.

Added block size 16x16 for subshifting with AVX2 implementation. Fixed bug with **chroma=false** in MDegrainN no copy of chroma planes to output from previous release.

**20220710 – Release 10.07.22**

10 Jul 18:47 [DTL2020](https://github.com/DTL2020) [ba3307f](https://github.com/DTL2020/mvtools/commit/ba3307f6e703d37d8df9fc12b16b06946fcc56b8) [Release 10.07.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.12) Pre-release

Added single pass colour overlapped processing in MDegrainN. Fixed regression of not using thSADC/thSADC2 in single pass processing.  
Added tweaking param **adjSADLPFedmv** to MDegrainN to adjust SAD of MVs passed thSAD check after filtering. Float param. Default 1.0 - no correction. Recommended value about 0.8.

Added **optSearchOption=6** to MAnalyse. In this mode DX12-ME only used for getting MVs from HW accelerator and SAD calculation performed on host CPU. Compute.cso shader is not used. Also for 8x8 8bit block available **UseSubShift=1** for MAnalyse to use sub-shifting (allow to run with **pelrefine=false** at MSuper and save RAM).  
May be faster at some combinations of host/accelerator. Also the SAD calcultation of shader for **pel=2** and **pel=4** still not completely correct (higher in compare with original mvtools).

**20220702 – Release 02.07.22 \_2**

02 Jul 18:39 [DTL2020](https://github.com/DTL2020) [666b46b](https://github.com/DTL2020/mvtools/commit/666b46b6fcb3e268846b7baad6d3234f29d6090d) [Release 02.07.22 \_2](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.11) Pre-release

Added **UseSubShift** param to MDegrainN and **pelrefine=true**/**false** to MSuper. Default = 0, set to 1 to enable.

If **pelrefine=false** in MSuper - all other filters must use **UseSubShift=true** (or **optSearchOption=5** for MAnalyse).

Also redesigned MDegrainN no-overlap processing to single pass YUV formats processing - looks like also added to performance.

\_2 updated release with fixed bug of MDegrainN crash.

**20220512 – Release 12.05.22**

12 May 07:32 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.10](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.10) [f0da1d9](https://github.com/DTL2020/mvtools/commit/f0da1d9174e65867486d1321ba330471b3e4ee64) [Release 12.05.22](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.10) Pre-release

Added **MVLPFGauss** MVs low-pass filtering mode to MDegrainN as single control-param adjustment. MVLPF implemented in all processing modes of MDegrainN (chroma enabled and overlap enabled). Default = 0 (disabled), float param. Expected adjustment range 0.5..3.0. Added usage of **scaleCSAD** param defined in MAnalyse in the secondary SAD check after MVLPF processing in MDegrainN.  
Fixed number of bugs.

**20220410 – Example of MVLPF processing**

10 Apr 18:41 [DTL2020](https://github.com/DTL2020) [ca8af8f](https://github.com/DTL2020/mvtools/commit/ca8af8f8c6befea37128982ae2858a7f744127f9) [Example of MVLPF processing](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.09) Pre-release

First working example of motion vectors low pass filtering to decrease conversion of luma noise to additional phase noise after MDegrainN processing.  
New control params for MDegrainN: **MVLPFCutoff**, **thMVLPFCorr**.

**MVLPFCutoff**: cut off frequency of the low pass filter for motion vector's components (dx,dy) in temporal (**tr**) axis.  
Default 1.0 additional processing disabled.  
Valid range 0..1. Estimated working range when enabled: 0.05 to 0.5.

**thMVLPFCorr**: Maximum difference between original and filtered vector's dx,dy components and new filtered vector (any of component) for correction. If difference above this value (not internally scaled to **pel** value) - the original vector from MAnalyse is used.  
Value =0 (default) disables correction. May be useful to fix some bugs at the footage with lots of different movement and noise.  
Expected good value: **pel**\*(4..10).

**20220209 – First fully in-accelerator processing of pel 1,2,4**

09 Feb 18:39 [DTL2020](https://github.com/DTL2020) [5267086](https://github.com/DTL2020/mvtools/commit/526708604c725d4a1a7f7890ba97d29ad36e0206)

[First fully in-accelerator processing of pel 1,2,4](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.08) Pre-release

MAnalyse can now process all supported **pel** (1,2,4) SAD computing inside accelerator. Using run-time sub-sample shifting with kernel size 8 (close to Lanczos3..4) of single uploaded full-frame plane. Both **chroma=false** and **true** supported. Block size only 8x8 and 16x16.

**20220130 – Added pel 2 and 4 for SO=5**

30 Jan 16:52 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.07](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.07) [b44058c](https://github.com/DTL2020/mvtools/commit/b44058c9569af387069cfcc63d5a96e811e0e874) [Added pel 2 and 4 for SO=5](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.07) Pre-release

Fixed bug with **pel**>1 in MDegrainN and added support for **pel=2** and **pel=4** processing with SO=5. The **thCohMV** param is not auto-scaled internally with pel adjustment so it is recommended to scale at about **xPel** value before fine-tuning.  
The SAD processing for pel 2 and pel 4 still onCPU so expected better performance in next versions.

Added new params to MDegrainN:

1. **adjSADzeromv** (1.0 - default, no op) - possible SAD multiplier for zero-move blocks (before thSAD processing for getting block's weighting value). Float value. Recommended values: 0.9..0.4-. Possible medium values 0.75..0.5. Allow to increase degraining at static areas.

Example: setting 0.5 result thSAD for zero move blocks (static) will be thSAD\*2.

1. **adjSADcohmv** (1.0 - default, no op) - possible SAD multiplier for blocks in coherent moving areas (before thSAD processing). Float value. Recommended values: 0.9..0.4-. Possible medium values 0.75..0.5. Allow to increase degraining at big enough coherent moving areas of much larger 1 block\_size size (like camera pan movement over non-changing scene).
2. **thCohMV** (-1 default, no op) - threshold to detect if block's move vector is equal to surround blocks (top,left,right, down) move vectors. -1 - disables this part of processing (faster), 0 - lowest working value. Recommended values 0..4. Possible range - 0..unlimited int. Too high values will create error-blended blocks (like with too high thSAD value).

**20220120 – Build 20.01.2022 (DX12 chroma SAD)**

20 Jan 18:49 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.06](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.06) [f0cd1f6](https://github.com/DTL2020/mvtools/commit/f0cd1f6a9afbd402887e587ddf3404239e203371)

[Build 20.01.2022 (DX12 chroma SAD)](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.06) Pre-release

Build 20.01.2022. Added chroma support in shader SAD calculation and **scaleCSAD** param passing to SAD calculation. Added optimization of resource upload to accelerator (direct pointing to Y buffer from AVS environment).

**20211231 – First working DX12\_ME**

31 Dec 13:16 [DTL2020](https://github.com/DTL2020) [r.2.7.46-dx12\_me.a01](https://github.com/DTL2020/mvtools/tree/r.2.7.46-dx12_me.a01) [f80451f](https://github.com/DTL2020/mvtools/commit/f80451f4f992b10b030e3ff00b77b7d7a5e6c143) [First working DX12\_ME](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-dx12_me.a01) Pre-release

testbuild. **optSearchOption=5** with **levels=1** is DX12\_ME mode of MAnalyse. Only tested with 1920x1080 frame size, only accept data YV12 format. Only tested with MDegrainN, may crash other MDegrains with invalid vectors.

**20211123 – Testbuild 23.11.21**

23 Nov 19:56 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.05](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.05) [c64e1e7](https://github.com/DTL2020/mvtools/commit/c64e1e7ba4e67141bad6923cf1330a269aac4f43) [Testbuild 23.11.21](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.05) Pre-release

Added **optPredictorType** 3 and 4.  
3 - only check the SAD of the predictor from level 1 at level 0. SAD value should be typical.  
4 - only use interpolated vectors from level 1 (SAD value is reduced typically and thSAD in MDegrainN need to be lower).  
Should work with any other options - no special requirements.

**20211121 – Mostly finished SO2**

21 Nov 18:22 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.04](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.04) [9d14181](https://github.com/DTL2020/mvtools/commit/9d141819bb7e509a0b959ac93bd9e0aba06c9877) [Mostly finished SO2](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.04) Pre-release

Mostly finished AVX2 version of **optSearchOption=2** and fixed **optPredictorType=2**. Intel C++ builds and MSVS2019 executables with intel-dlls for run. The \*sf.dll version contain test of special sad return value from non-finest level (only with **optSearchOption=2** and **optPredictorType=1**) for possibly better denoising (in selected cases). \*sf.dll typically require (allow) lowering thSAD(2) to about half of 'standard' or less. **optPredictorType=2** mostly usable with levels=1 and 2 and may cause severe distortions with levels > 2. All executables may require minimum AVX2 CPU to run.

**20211113 – Build 13.11.2021**

13 Nov 20:38 [DTL2020](https://github.com/DTL2020) [r.2.7.46-a.03](https://github.com/DTL2020/mvtools/tree/r.2.7.46-a.03) [02c98d7](https://github.com/DTL2020/mvtools/commit/02c98d756202f597d046aa591bd178154c662003) [Build 13.11.2021](https://github.com/DTL2020/mvtools/releases/tag/r.2.7.46-a.03)

SSE2 and AVX2 executables with latest version of FetchPredictors\_sse41 (only activated with **optSearchOption** > 0).

**20211027 – Test for speed and denoise quality**

27 Oct 09:13 [DTL2020](https://github.com/DTL2020) [r2.7.46-pre.a.01](https://github.com/DTL2020/mvtools/tree/r2.7.46-pre.a.01) [e434d05](https://github.com/DTL2020/mvtools/commit/e434d05556e5adb9747f377a1ff97e00a4904cdb)

[Test for speed and denoise quality](https://github.com/DTL2020/mvtools/releases/tag/r2.7.46-pre.a.01) Pre-release

Test for speed of early zero weight skip vs old method.  
Test for degrain quality of equal-weighting vs sad-wise old method of weights calculation. May contain OpenMP auto threads enabled so disable avstp MT (**mt=false**) in MAnalyse().