

"good" denoising	Run 1	Run 1	Run 1	Run 2	Run 2	Run 2
	UNSTACKED	UNSTACKED	UNSTACKED	UNSTACKED	UNSTACKED	UNSTACKED
	VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS	FFMPEG reported FPS	VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS	FFMPEG reported FPS
VHS video #1	569.78	402.75	378	569.11	407.08	383
VHS video #2	576.66	411.95	388	567.03	409.52	386
VHS video #3	575.95	407.71	385	569.37	400.93	378
VHS video #4	569.66	407.94	385	574.65	409.08	386
VHS video #5	563.91	404.40	382	574.81	405.70	383
VHS video #6	563.43	407.41	383	567.25	403.05	379
VHS video #7	561.43	402.91	378	565.67	399.23	375
VHS video #8	578.64	406.25	383	570.76	399.32	378

(the "before" clip is not used when unstacked)

UNSTACKED only the 720x576 VHS processed "AFTER" video is set\_output  
Stack Horizontal the "before" and "after processed" videos are stacked horizontally and then set\_output i.e. the width is doubled

Conclusions: no significant variability in unstacked "vspipe to null" fps, nor in ffmpeg reported fps  
significant variability in STACKED "vspipe to null" fps ... ON SEPARATE COMMANDLINE RUNS when presumably Windows 11 has control between process runs  
one should expect less fps from the STACKING process, however NOT the SIGNIFICANT variability between test runs of STACKED

NOTE: the variability has been observed in many test runs, nto just in this set

QUERY: is the variability caused by Windows memory management or some other factor ?  
noting that memory is 32Gb total and task manager reports that 2/3 of memory is "Available" continuously

ENVIRONMENT: CPU: AMD 3900X 12-core  
VGA: RTX nvidia 2060 Super 8Gb  
Mem: 32Gb  
Disk: system: 256Gb Samsung SSD, data: 7200rpm Black 6Tb  
Video Files: 8x VHS PAL captures, all mpeg2 720x576 25i and very noisy  
VapourSynth: R66 portable  
DG: release 252 2024/03/14

.vpy # use DGSource to open the same video file as "before" and "after" and process the "after" video and then stack them  
import vapoursynth as vs # this allows use of constants eg vs.YUV420P8  
from vapoursynth import core # actual vapoursynth core  
core.std.LoadPlugin(r'G:\HDTV\DGtest\Vapoursynth-x64\DGIndex\DGDecodeNV.dll') # do it like gonca https://forum.doom9.org/showthread.php?p=1877765#post1877765  
core.avs.LoadPlugin(r'G:\HDTV\DGtest\Vapoursynth-x64\DGIndex\DGDecodeNV.dll') # do it like gonca https://forum.doom9.org/showthread.php?p=1877765#post1877765  
before\_video = core.dgdecodenv.DGSource( r'G:\HDTV\DGtest\RESULTS\00\_PostcardsFromMannum\_sample-unprocessed\_interlaced\_CUT.DGI', deinterlace=1, use\_top\_field=True, use\_pf=False )  
after\_video = core.dgdecodenv.DGSource( r'G:\HDTV\DGtest\RESULTS\00\_PostcardsFromMannum\_sample-unprocessed\_interlaced\_CUT.DGI', deinterlace=1, use\_top\_field=True, use\_pf=False,  
dn\_enable=1, dn\_quality="good", dn\_strength=0.06, dn\_cstrength=0.06, dn\_tthresh=75.0, dn\_show=0, sh\_enable=1, sh\_strength=0.3 )  
stacked\_video = core.std.StackHorizontal([ before\_video, after\_video ] )  
stacked\_video.set\_output()

Example Commndlines: VSPipe.exe --filter-time --container y4m "G:\HDTV\DGtest\RESULTS\00\_PostcardsFromMannum\_sample-unprocessed\_interlaced\_CUT.vpy" --  
  
VSPipe.exe --container y4m --filter-time "G:\HDTV\DGtest\RESULTS\00\_PostcardsFromMannum\_sample-unprocessed\_interlaced\_CUT.vpy" - |  
ffmpeg.exe -hide\_banner -v verbose -nostats -f yuv4mpegpipe -i pipe: -probesize 200M -analyzeduration 200M -fps\_mode passthrough -sws\_flags lanczos+accurate\_rnd+full\_chroma\_int+full\_chroma\_inp -strict experimental  
-c:v h264\_nvenc -pix\_fmt nv12 -preset p7 -multipass fullres -forced-idr 1 -g 25 -coder:v cabac -spatial-aq 1 -temporal-aq 1 -dpb\_size 0 -bf:v 3 -b\_ref\_mode:v 0 -rc:v vbr -cq:v 0 -b:v 6000000 -minrate:v 500000 -maxrate:v 12000000 -bufsize 12000000  
-profile:v high -level 5.2 -movflags +faststart+write\_colr -y "G:\HDTV\DGtest\RESULTS\00\_PostcardsFromMannum\_sample-unprocessed\_interlaced\_CUT.result.mp4"

Hmmm ... however, when we use denoise "best" instead of "good" and leave the rest alone, the results change and the variability is masked or goes away ...

TESTS using DGSource dn\_enable=1, dn\_quality="best" # rather than "good"

"best" denoising	Run 1	Run 1	Run 2	Run 2	Run 2
	UNSTACKED quality=best	UNSTACKED quality=best	UNSTACKED quality=best	UNSTACKED quality=best	UNSTACKED quality=best
	VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS		VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS
VHS video #1	189.94	160.38		187.64	163.95

Run 1	Run 1	Run 1	Run 2	Run 2	Run 2
Stack Horizontal	Stack Horizontal	Stack Horizontal	Stack Horizontal	Stack Horizontal	Stack Horizontal
VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS	FFMPEG reported FPS	VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS	FFMPEG reported FPS
249.31	90.96	90	302.13	124.24	122
235.45	83.75	83	428.80	122.09	120
314.52	114.79	113	105.31	105.31	79
205.36	170.64	167	278.36	145.86	143
282.62	219.61	212	353.66	210.05	204
60.80	94.08	93	190.62	132.28	130
222.14	78.67	78	126.82	163.08	159
316.87	84.21	83	463.66	112.23	110

VHS video #2	185.61	162.12			187.47	161.69			174.04	150.07			171.76	149.05
VHS video #3	187.15	162.45			186.95	161.59			174.28	150.51			172.53	149.82
VHS video #4	187.40	162.87			186.00	160.14			172.61	147.94			170.03	150.63
VHS video #5	184.34	161.54			184.51	160.90			172.15	147.10			168.76	147.83
VHS video #6	183.64	161.86			181.95	160.29			168.65	147.85			171.43	133.53
VHS video #7	182.05	159.12			184.56	160.05			169.34	148.91			169.62	150.31
VHS video #8	182.02	160.20			182.11	159.25			171.33	146.17			170.00	149.15

OK, let's go back to using DGSource dn\_enable=1, dn\_quality="good" # rather than "best"  
... and try INTERLEAVE instead of stackhorizontal, just to see if a road may possibly point back to the 2x DGSource opens of the same file

(when the "before" clip is used, which it is not when unstacked)

"good" denoising	Run 1		Run 1		Run 2		Run 2		Run 1		Run 1		Run 2		Run 2	
	Not		Not		Not		Not		Interleaved		Interleaved		Interleaved		Interleaved	
	Interleaved		Interleaved		Interleaved		Interleaved		quality=good		quality=good		quality=good		quality=good	
	VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS			VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS			VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS			VSPiPE TO NULL FPS	VSPiPE TO FFMPEG FPS		
VHS video #1	567.58	405.11			570.48	410.16			966.75	589.00			971.98	579.59		
VHS video #2	567.80	408.60			575.56	414.49			979.19	601.98			990.39	602.32		
VHS video #3	571.08	394.54			575.76	397.11			968.86	607.44			980.18	608.46		
VHS video #4	566.83	407.38			574.95	410.28			981.18	597.23			983.61	591.17		
VHS video #5	571.77	407.12			568.30	407.75			920.11	583.99			983.82	584.26		
VHS video #6	570.79	394.65			566.69	396.28			960.16	586.46			969.13	577.27		
VHS video #7	562.76	388.26			562.66	394.76			978.91	553.72			975.17	561.12		
VHS video #8	567.46	395.83			570.85	397.74			973.81	581.65			968.67	582.84		

WTF ?? 900+ fps from denoising plus interleaving VS plain denoising alone (not using the "before" clip) ???

```
import vapoursynth as vs # this allows use of constants eg vs.YUV420P8
from vapoursynth import core
core.std.LoadPlugin(r'G:\HDTV\DGtest\Vapoursynth-x64\DGIndex\DGDecodeNV.dll') # do it like gonca https://forum.doom9.org/showthread.php?p=1877765#post1877765
core.avs.LoadPlugin(r'G:\HDTV\DGtest\Vapoursynth-x64\DGIndex\DGDecodeNV.dll') # do it like gonca https://forum.doom9.org/showthread.php?p=1877765#post1877765
before_video = core.dgdecodenv.DGSource( r'G:\HDTV\DGtest\RESULTS\00_PostcardsFromMannum_sample-unprocessed_interlaced_CUT.DGI', deinterlace=1, use_top_field=True, use_pf=False )
after_video = core.dgdecodenv.DGSource( r'G:\HDTV\DGtest\RESULTS\00_PostcardsFromMannum_sample-unprocessed_interlaced_CUT.DGI', deinterlace=1, use_top_field=True, use_pf=False, dn_enable=1, dn_quality="good", dn_strength=0.06, dn_cstrength=0.06, dn_tthresh=75.0, dn_show=0, sh_enable=1, sh_strength=0.3 )
interleaved_video = core.std.Interleave( [before_video, after_video] )
interleaved_video = core.std.AssumeFPS( interleaved_video, fpsnum=25, fpsden=1 )
interleaved_video.set_output()
```