



# Deshaker (Latest version: [2.5](#))

Deshaker is a video stabilizer for VirtualDub. It eliminates camera shakiness and makes panning, rotation and zooming smoother.

## Features:

- Stabilizes horizontal/vertical panning, rotation and zooming.
- Works in 2 passes for optimal simulated camera motion.
- Can use previous and future frames to fill in "black areas" caused by the stabilizing.
- Optional motion vector video output during first pass.
- Lots of settings for the advanced user.
- Human guidance is possible on a frame per frame basis.

## External Deshaker links:

[[english](#)] A good but slightly dated Deshaker guide by John Meyer:

[http://www.sundancemediagroup.com/articles/deshaker\\_guide.htm](http://www.sundancemediagroup.com/articles/deshaker_guide.htm)

[[deutsch](#)] The Deshaker manual (this page) and the guide above translated to German by Armin Schreiter:

[http://www.fhc-hildesheim.de/tipps\\_tricks.htm](http://www.fhc-hildesheim.de/tipps_tricks.htm)

[[español](#)] The Deshaker manual (this page) translated to Spanish by Ramón Cutanda:

<http://www.videoedicion.org/documentacion/index.php?article=140>

[[русский](#)] Detailed Russian article comparing Deshaker to other video stabilizers, by Eugene Vasiliev:

<http://www.ixbt.com/divideo/digital-video-guide/5-2-1-stabilization.shtml>

[[français](#)] The Deshaker manual (this page) and the John Meyer guide translated to French by Andre Milliner:

<http://letransfert.xooit.fr/t215-Deshaker-traduction-en-Francais-de-la-DOC.htm>

## Installation

Download [Deshaker](#) and place *Deshaker.vdf* under the VirtualDub plugins directory and (re)start VirtualDub.

I strongly recommend using VirtualDub version 1.9.9 or later for Deshaker version 2.5 or later.

There's no 64-bit version of Deshaker yet, so you'll need the 32-bit version of VirtualDub to run Deshaker.

## The Short Guide

These are the minimum number of steps you must make in order to stabilize a video clip:

1. Open your video clip in VirtualDub (menu: *File/Open video file*).
2. Add the Deshaker filter (menu: *Video/Filters*, click: *Add*, double-click: *Deshaker*).
3. In the Deshaker configuration dialog that appears, select the correct *Source pixel aspect* and *Video type*. Also, if your camcorder has a CMOS sensor instead of a CCD sensor, check *Camcorder has a rolling shutter*, and if possible, find out the correct rolling shutter amount (see settings below). Then, click *OK* to close.
4. Step through the video in some way (for example using *File/Preview filtered* in VirtualDub). No useful video is created during this first pass.
5. Open the Deshaker configuration dialog again (menu: *Video/Filters*, double-click: *Deshaker* line), and select *Pass 2* by clicking on it.
6. Encode the video the usual way (menu: *File/Save as AVI*).

The default settings will however result in black borders caused by moving the frames around to compensate for shaky video. There are various settings that can help limit the size of the borders, remove them completely or make them less ugly. Read about these pass 2 settings below.

## The Long Guide

This section describes all the settings in detail.

The default parameter settings are chosen to work pretty good under most circumstances. However, you can always get better results by adjusting some settings. For example, some complicated clips might require that you change some pass 1 settings, or you will get jerky video. Also, you'll need to change some pass 2 settings if you want to take care of the black borders.

Remember that all settings have a purpose, or they wouldn't be there. Most people I hear from seem to use the default settings or change very few of them. And many clips I've seen from other users are far from as good as they could have been. I know the settings may seem a little hard to understand and maybe a little too many, but if you want perfect looking video you should at least try.

To understand these settings it is necessary for you to understand how the filter works, so here comes a brief description.

## How the Filter Works

During **Pass 1** the filter tries to find the panning, rotation and zoom that, when applied to the current image, makes it look like the previous image (almost). It does this by taking blocks of pixels in the two images and finding the shift that makes them match as good as possible. First, to find the "big" motion, the images are reduced a lot in scale and the initial shift is calculated from the whole (scale-reduced) images. This shift is then finetuned in a scale twice as big and with four times as many blocks, and so on. Finally we have the block shifts (or motion vectors) in the full image. The filter then finds the best values for panning, rotation and zoom based on these motion vectors and those values are written to the log file.

At the start of **Pass 2** the values in the log file are read and optimal camera motion is calculated based on the smoothness settings. The filter then transforms each image of the video clip according to this motion.

## The Settings

| Parameter                       | Description   |
|---------------------------------|---|
| Source pixel aspect             | Tells the filter what pixel aspect ratio the source clip has. It's very important that you set this correctly, or the rotation (for example) won't be made correctly. You may enter any value if you can't find the preset you're looking for.  |
| Video type                      | Selects between progressive video and interlaced video (for input and output). If the video is interlaced you must also determine if the video has the first or last field on the first line. In general, DV uses lower field first and all other formats (including HDV) use upper field first. You can easily check that your setting is correct by looking at the motion vectors for the two fields when there's motion in the clip. If the vectors point in opposite directions most of the time the setting is incorrect.  |
| Camcorder has a rolling shutter | <p>Most modern camcorders (those with a CMOS sensor instead of a CCD sensor) don't capture the whole frames at once, but instead use a "rolling shutter". Here the lines of a frame are captured one after another at slightly different (although usually overlapping) times, which can cause some ugly effects. For example, vertical lines will get slanted when doing a fast horizontal pan, and when panning up/down everything will get squeezed and stretched.</p> <p>For Deshaker, a rolling shutter means that the camcorder shake parameters (panning, rotation and zoom) aren't necessarily constant over the entire frame, but can change slightly from one line to the next. If you enable this setting, Deshaker will therefore allow these parameters to change linearly in the vertical direction during pass 1 (and you will see a range of values under the output video). This makes extracting these parameters more reliable and it also makes it possible for pass 2 to remove the rolling shutter distortions if you keep the setting enabled during that pass too. Although the actual camcorder shake isn't linear during the capture of a frame, it's usually a good enough approximation to make the distortions disappear almost completely. But I still strongly recommend using the internal camcorder stabilizer too.</p> <p>If you have one of those compact CMOS cameras with bad (or no) internal stabilization, and the clip has very fast shakes (for example if the camera is mounted on a bike or similar), you probably won't be able to get a very good result from Deshaker, since the distortions will be far from linear in this case. Get a CCD camera capable of high shutter speeds for these kind of videos.</p> <p>Different camcorder models may have different rolling shutter speeds (how fast the shutter rolls downwards). And a certain model could possibly even have different speeds under different conditions. To be able to handle all speeds properly, you can enter an amount from 0 to 100%. You can try which amount setting looks best, or you can measure it more precisely yourself using the following procedure:</p> <p>Put the camcorder on a tripod (or something flat) and point it towards some vertical</p> |

line (or bar) that covers the whole height of the frame. Now, using a fast shutter speed (to make the line sharp), record some video where you pan left and right (direction doesn't matter) at a pretty fast constant speed, as well as when it's completely still. Then, pick a still frame and a panning frame (where the vertical line is slanted). If you recorded interlaced video, the panning frame will contain two fields (and the line will appear twice), which is what you need. If you recorded progressive video, you'll need two consecutive frames instead. Then, find the horizontal position (x-pos) of the vertical line at 5 places, namely at  $St$  = top of still frame,  $Sb$  = bottom of still frame,  $Ft$  = top of first panning field/frame,  $Fb$  = bottom of first panning field/frame,  $Nt$  = top of next panning field/frame. Now, the rolling shutter amount should be calculated as:  $100 * (Fb - Ft + St - Sb) / (Nt - Ft)$ .

The default rolling shutter amount is 88%, which was measured from the Sony HDR-HC1(E) camcorder. (Deshaker versions before 2.4 had 88% fixed.)

If you measure the amount for another camcorder, please let me know so I can publish the values for other users. Here are the ones I've got so far:

- **Canon EOS 500D**: 71%
- **Canon EOS 550D**: 66%
- **Canon EOS 5D Mark II**: Probably somewhere between 66% - 72%
- **Canon HV20**, at 1440x1080 25p: 38%
- **Casio Exilim EX-F1 Pro**, at 1280x720 30fps: 22%
- **Casio Exilim EX-F1 Pro**, at 1920x1080 60fps: 72%
- **HD Hero**, at 1280x720 60fps: 82%
- **iPhone 4**: 97% (+/- 2)

(Note: Deshaker assumes that the rolling shutter rolls downwards over the frames. For this to happen, the iPhone must record video in landscape mode with the lens in the upper part of the phone. Otherwise you'll probably need to rotate the video before Deshaker, and rotate it back afterwards.)

- **Kodak Zi8**: Somewhere between 80% - 100%
- **Oregon Scientific ATC5K**: 100%
- **Panasonic HDC-SD300 EG-K**: 73%
- **Pentax K7**, at 1536x1024 30fps: 75%
- **Sanyo Xacti HD1010**, at 1280x720 60fps: 48%
- **Sanyo Xacti HD1010**, at 1920x1080 30fps: 28%
- **Sony HDR-FX7**: 86.4% (+/- 2.0)
- **Sony HDR-HC1**: 88%
- **Sony HDR-HC1E**: 88%
- **Sony HDR-V1U**: 86.4% (+/- 2.0)
- **Sony HVR-A1E**: 82%

Note that only rolling shutter distortions that are caused by moving/shaking the camcorder can be removed. Fast moving objects within the frame will sadly still look as strange as before.

You cannot use the deep analysis algorithm with rolling shutter video.

|                |  |
|----------------|--|
| Log file       | The name of the log file to write in pass 1 and read in pass 2.  |
| Append to file | If checked, it appends to the log file instead of overwriting it. This is useful if you want to process certain frames again.  |
|                |  |
| Video output   | What kind of output to generate during pass 1. If you select <i>Motion vectors</i> you will see the previous image with arrows showing how to move the blocks to match the current source image. Only white arrows are used to calculate panning, rotation and zoom factor. For interlaced video both of the fields are shown, the first field above the last. |
| Block size     | The width and height of the blocks used during the image matching.   |
| Scale          | The maximum scale to do the image matching on. <i>Full</i> is best but very slow.  |
| Use pixels     | How many pixels in the blocks to actually use when matching. <i>All</i> is best but very slow.   |
| Color mode     | The image matching can be made in color or grayscale. Grayscale is a little faster but the matching is worse. This setting has nothing to do with the color of the final output of pass 2.   |

|  |  |
|--|--|
| Initial search range   | In the most reduced scale this percentage gives the maximum shift while matching. Lower = slightly faster.   |
| Differential search range  | When going from a reduced scale to twice the size, the filter already has approximate shifts. The differential search range is how many pixels up, down, left and right from these approximate shifts to look when matching. Lower values make processing a lot faster but when objects move quickly over the frames, or if you have very fast/big rotations and/or zooms, matching can fail for some blocks.  |
| Detect rotation/zoom   | If Deshaker has problems finding a stable background to match on, and you know there is no notable rotation/zoom in the problematic frames, you can turn off detection of rotation and/or zoom (individually) for these frames (or the entire clip). This will make Deshaker assume there is no rotation/zoom and therefore not accept groups of motion vectors that indicate rotation/zoom. And this should make it easier for Deshaker to find the background. But as I said, there mustn't be any rotation/zoom in the analyzed frames for this to work. In fact, there might be problems even when the camera is moving forward quickly, because it looks rather similar to zooming.   |
| Discard motion of blocks that have match value less than X                 | If the best match value found while matching a certain block is below X this block is discarded. The blocks in higher scales that depend on this block are also discarded. These discarded blocks show no motion vector in the output video of pass 1.   |
| Discard motion of blocks that have 2nd best match larger than best minus X | If the second best match value found while matching a certain block is close to the best match value, this means that the best shift found isn't very reliable. It could just as well be this second shift that's the correct one. For example a blue sky has good match values for all shifts. Setting this parameter correctly discards those unreliable blocks. The blocks in higher scales that depend on these blocks are also discarded. These discarded blocks show no motion vector in the output video of pass 1.   |
| Discard motion of blocks that move more than X pixels in wrong direction.  | When calculating the best values for panning, rotation and zoom based on all motion vectors, the filter soon discovers that some blocks have motion that don't quite fit in with the "main motion". If a block has a motion vector that is more than X pixels off this main motion it is discarded in order to get more precise values. The motion vectors for these blocks are drawn in red in the output video of pass 1 and are not used in any way. Since the panning, rotation and zoom are all done in only 2 dimensions, it's possible to get a lot of these blocks that don't seem to fit in with the main motion when the effect of perspective gets big. This happens when the camera is set to wideangle and not just rotating (along any axis), but actually moving. Here you should use high X values (or maybe limit the matching area to the most distant parts). When zoomed in a lot or when the camera is stationary, the perspective effect is low and you should use low X values (even 1) to discard motion of objects moving even very slightly.   |
| Discard motion of blocks that move more than X pixels (absolute motion)    | Simply discards all blocks that have a motion vector that is longer than X pixels. Might be useful in some cases.  |
| Remember discarded areas to next frame                                     | When enabled, this feature makes Deshaker try to ignore approximately the same areas from one frame to the next. Deshaker will then become a lot more successful in ignoring moving objects. As long as they enter the scene rather slowly (by not covering too much of the background), Deshaker will usually be able to ignore those objects even if they eventually grow to cover most of the frame.  |
| Deep analysis if less than X % of vectors are ok                           | <p>When set to 0% (which is the default value) this does nothing. To enable it, set it to 25% or something. Then, if less than 25% of the found vectors get white using the standard algorithm, this deep analysis algorithm kicks in. It does a much better job finding an object to lock on where the standard algorithm can be confused and lock on two separate objects at the same time, usually resulting very few white vectors and weird values. Use this in combination with the discard setting above to control how much the vectors in the object you want to lock on (usually the background) may vary. For example, the blocks on a moving person usually don't move "coherently" as the background usually does (if the camcorder is stationary). So by using a low value for the discard setting above (maybe &lt; 1) the person would probably get very few white vectors and this algorithm would therefore hopefully lock on the background instead, even if it's a rather small part of the frame. This algorithm is very good, but also VERY slow (depending on other settings and how complicated the video is).</p> <p>You cannot use the deep analysis algorithm with rolling shutter video.</p> |

|   |   |
|---|---|
| Skip frame if less than X percent of all blocks are ok. | If less than X percent of all blocks are ok (i.e. the rest of the blocks either have no acceptable match or have been discarded), skip this frame (set panning and rotation to 0 and zoom scale to 1)   |
| Ignore pixels outside <i>area</i>                       | Only <i>area</i> is used during matching. If you check <i>Let area follow motion</i> , the area will move along with the motion vectors between different frames. You can use it if you want to stabilize on a moving object instead of the background. Just don't expect this to work perfectly :)   |
| Ignore pixels inside <i>area</i>                        | <i>area</i> is not used during matching.  |
| Ignore pixels darker than X % brightness                | Most camcorders seem to use some kind of temporal smoothing internally to reduce noise. This means that the noise that is left will look very similar between two successive frames, and may cause Deshaker to incorrectly match on it. This problem gets worse when the camcorder is forced to use high gain (when filming in dark places) and it's also a lot worse in dark areas of the frame. By using this option you can ignore dark pixels during matching. What threshold to use depends on your camcorder and the gain used in the clip. The default 15% seems fine for my camcorder in daylight but may not be ideal for your camcorder/clip.   |
|   |   |
| Same destination properties as source                   | If checked, the destination video will get the same size and pixel aspect ratio as the source and the two settings below will be ignored.   |
| Destination pixel aspect                                | Tells the filter to generate video with a certain pixel aspect ratio. If you plan to watch the clip on a computer monitor you'll probably want to use square pixels. You may enter any value if you can't find the preset you're looking for.   |
| Destination video size                                  | The size of the output video you want from pass 2.  |
| Generate "interlaced progressive" video                 | <p>Usually, Deshaker will generate video with the same video type (interlaced/progressive) as the source video. But if you enable this setting, it can turn interlaced video into progressive video with twice the framerate. Earlier versions of VirtualDub didn't allow for filters to change the frame rate, so the two progressive frames are interlaced into the same output frame. In order to get the true progressive video, you must separate the fields of this video afterwards. In later versions of VirtualDub you can do that by adding a <i>Bob doubler</i> filter after Deshaker, using the deinterlacing method <i>None - alternate fields</i>. You could also do it with AVISynth and its <i>SeparateFields</i> function (and a preceding <i>ComplementParity</i> for "upper field first" video). You should never deinterlace this video in any other way than to simply separate the fields.</p> <p>The difference between true interlaced output and this interlaced progressive output is that the video becomes twice as high (in order to fit two frames into one) and more importantly that the lower field (or rather "frame" in this case) isn't calculated with a one line offset as in interlaced video.</p> |
| Resampling  | The algorithm to use when calculating pixel colors from the source image. Note that none of these are very good when the destination size is a lot lower than the source. Instead, add a separate 2:1 reduction filter (or something) after the Deshaker to reduce scale a lot.   |
| Edge compensation                                       | <p>A side-effect from stabilizing video is black or strange looking borders caused by moving the image around to compensate the shaking. To reduce or eliminate this effect you can zoom into the image. The following options do this in different ways:</p> <ul style="list-style-type: none"> <li>• <i>None</i>: No zoom is added.</li> <li>• <i>Adaptive zoom</i>: Determines, for each frame, how much zoom is needed to avoid borders. This additional zoom is then smoothed together with the stabilizing zoom changes using the zoom smoothness parameter, so the borders still become visible sometimes.</li> <li>• <i>Adaptive zoom only</i>: Same as <i>adaptive zoom</i>, but the zoom smoothness parameter is only applied to this adaptive zoom and not for stabilization. So, with this setting no smoothing will be made to the zooming made in the clip.</li> <li>• <i>Fixed zoom</i>: Deshaker will add a constant zoom to completely eliminate any</li> </ul>  |

|   |   |
|---|---|
|   | <p>borders. This zoom may be different for each scene of the clip.</p> <ul style="list-style-type: none"> <li>• <i>Adaptive+fixed</i>: First, <i>adaptive zoom</i> will be used. Then, a smaller fixed zoom will be added on top of this to completely eliminate any borders. This will usually result in less zoom than <i>fixed zoom</i>.</li> <li>• <i>Adaptive+fixed only</i>: First, <i>adaptive zoom only</i> will be used. Then, a smaller fixed zoom will be added on top of this to completely eliminate any borders.</li> </ul> <p>If you're using any edge compensation (i.e. anything but <i>None</i>), you can limit the amount of zoom needed to avoid the borders by using lower values for <i>Motion smoothness</i> and especially <i>Max. correction limits</i>.</p>   |
| Previous and future frames to fill in borders | <p>This is a kind of additional edge compensation. When the current frame doesn't contain any image data in the "real world area" we want in our output, it can search in past and future frames and use that image data instead. As long as that area is part of the object (or background) that was stabilized on in pass 1, this usually works perfectly. But when other objects move in these areas, it can look a bit strange. It's usually better than black anyway, so I strongly recommend using this feature. Just make sure you crop any borders from the source clip before Deshaker. Otherwise this won't work very well at all.</p> <p>If the current frame is number 50, it searches frames in the following order to find the image data it needs: 50, 49, 51, 48, 52, 47, 53, 46, 54 etc. So for each pixel the closest frame (chronologically) is always used. You can set how many frames to store internally for this feature. More is always better but keep in mind that each frame needs up to 8 MB of memory (for full HD) and it can take a long time to search through all frames for image data.</p> <p>Try using this feature on a clip with lots of panning, and zoom out by setting <i>Extra zoom factor</i> to something like 0.6. Then you'll get a panorama like effect. It's slow, but fun. :)</p> |
| Soft borders                                  | <p>If enabled, <u>all</u> relevant previous and future frames that Deshaker has stored in memory will be used to calculate the border (as opposed to just using the closest frame in time). For each pixel, it takes an average of the corresponding pixels in all these frames, weighted so that the frames closest to the current frame (in time) get higher importance.</p> <p>This option will also allow you to make the transitions between a frame and the frame(s) beneath it smoother. The transition width value you supply is how far in from the edge of a frame the transition will start. The transition will be applied also to the current frame, so you'll actually lose some information in the edges of the main frame by using this option. But I'd say it's worth it. If you don't like it, just set the transition width to 0.</p> <p>Enabling 'Soft borders' improves the appearance of the border quite a lot, especially when there are objects moving there, but it is also very slow when there are lots of border pixels.</p>   |
| Extrapolate colors into border                | <p>If enabled, any black border pixel that couldn't be fixed in another way, is replaced by a color extrapolated from the edge of the known video pixels. This option is very slow when there are lots of border pixels, and I strongly recommend using 'Previous and future frames to fill in borders' in combination with this option.</p>  |
| Extra zoom factor                             | <p>An additional zoom factor to apply to the video. This can for example be used together with <i>fixed zoom</i> edge compensation to zoom out just enough for the borders to become slightly visible but still remain hidden in the overscan area of the TV. Then you will see more of the video while still not seeing any borders on the TV. Or you can use it to zoom in slightly to get less borders. This setting won't cause another resampling, so using a little extra zoom to finetune your results is perfectly safe, quality wise. (Btw, all output pixels are always resampled directly from the original source pixels exactly once no matter what settings you use. So, no double resampling ever. This includes areas taken from previous or future frames.)</p>  |
| Motion smoothness                             | <p>These values determine how smooth the motion will be. You can set a parameter to 0 to turn off the smoothing completely. This can be useful if you don't want to stabilize zoom for example. But you must use a rather high zoom smoothness if you use adaptive zoom as edge compensation, or the video will zoom in and out very fast to avoid the borders. The motion smoothness calculation is not time-based but rather</p>  |

|                        |  |
|------------------------|--|
|                        | based on frame count, so you'll probably want to use larger values for video with high number of frames per second. And in this case, the fps for interlaced video will count as twice as high, since each of its frames contains two "sub frames" (fields). If you enter -1 as smoothness you will get "infinite smoothness", meaning the camera will appear to be stationary at the position of the first frame that was processed in pass 1. This feature usually doesn't work perfectly, though. |
| Max. correction limits | To keep the corrections (that cause the black borders) small during fast pans (for example), you can limit the maximum corrections. Especially when using <i>fixed zoom</i> as edge compensation these settings keep the zoom from being too large. What these settings really do is automatically lower the motion smoothness values in certain parts of the video when it becomes necessary.   |

## Things to Think About

Deshaker works best for video shot using high shutter speeds (at least 1/200 sec or so). Otherwise you might get motion blur without motion in the deshaked video. Also, the matching will work better.

Always try to make sure the white vectors end up on the background in the frames for best results, i.e. try to make Deshaker completely ignore objects moving in front of the background. If the white vectors jump around between different objects from frame to frame, you need to do something to get a good result. Either use the "ignore area" settings or something else. There are many things you can do. Experiment with the settings!

Don't forget to set the source video pixel aspect ratio. For example if your source video has resolution 720x480, 704x480 or 352x240, it's probably NTSC, and if it has 720x576, 704x576 or 352x288, it's probably PAL.

If there is no zooming in the video clip, use *Adaptive zoom only* as edge compensation or if you don't want any edge compensation, set zoom smoothness to 0. This is because the zoom stabilizing can't distinguish between zooming due to the camera moving towards something or if it's actually zooming. You may think it's the same thing, but it isn't. At least not for wideangle shots.

Often there's conflicting motion of large objects in the video clip. In this case the filter tries to "follow" the motion of the biggest object (usually the background). The setting that discards blocks that move more than X pixels in wrong direction can be used to some extent to control this. A low value makes it follow only the biggest object, while a larger value makes it follow the average motion of all objects (that aren't moving too much in the "wrong" direction). You can also use the image matching area to control which object to follow or try enabling the "deep analysis", which is designed specifically for this problem.

Video with a high degree of barrel or pincussion distortion will not look good when deshaked. If you get a lot of distortion in your video, try avoiding the extreme zoom positions of your camcorder. Extreme wide angles will unfortunately always result in a less than perfect looking stabilized video due to some geometrical effects that Deshaker doesn't deal with.

If you're a perfectionist (like me), always use the slowest settings. This makes it *\*really\** slow! :)

## Human Guidance

Sometimes it can get nearly impossible to automatically match two successive images correctly (or rather 'the way you want to'). This usually happens when the images have very little in common or when there's conflicting motion of large objects (see 'Things to Think About' above). In order to make a complete video clip look perfect, you might want to use different settings during pass 1 for different parts of the clip (or even single frames). You can do this in two ways:

- You can use the *Append to file* option to continue writing to the same log file. You can then change the settings, step a few frames, change again, step some more etc. The frame numbers are written to the log file so you can even go back and redo some frames. If a frame exists more than once in the log file, the last one is used in pass 2.
- You can edit the log file manually. The values on each line in the file are (from left to right) frame number, x- and y-panning (in pixels), rotation (in degrees) and zoom factor. When "rolling shutter" is enabled the four values for panning, rotation and zoom are given twice; first for the middle line and then for the first line of the frame. If a frame is the first frame of a new scene, the word 'new\_scene' should be at the end of that line. You can delete lines that got completely wrong (and that you don't care to try to fix in a better way). Gaps in the file are treated as zero-panning, zero rotation and no scaling.

## Stabilizing Interlaced Video

Deshaker can deshake interlaced video, but the resulting video will look slightly softer than for progressive video since the fields of interlaced video are resampled individually and they only have half the vertical resolution of a progressive frame. The resampled fields are then reinterlaced in the output video. There is really no sense in trying to detect non-moving pixels and keep them "un-deinterlaced" since you wouldn't be deshaking video shot on a tripod anyway, and without a tripod the pixels will almost always move slightly. (Deinterlacing with motion estimation in some way might give a better result but I'll probably not try implementing that anytime soon. And for certain reasons I don't like the idea of it.)

If you want, here's another way to stabilize interlaced video that might look better when some parts of the video are completely non-moving.

1. Use my smooth deinterlacer (or Xesdeeni's AVISynth port of it, or other advanced deinterlacers) to generate 50 or 60 fps full-sized video.
2. Feed it into Deshaker as progressive video (in two passes).
3. Run the resulting video through this little AVISynth script (that I got from Xesdeeni):

```
AVISource("deshaked.avi")
SeparateFields()
SelectEvery(4, 1, 2) (or SelectEvery(4, 0, 3) depending on the output field polarity)
Weave()
```

If you stop after the second step you'll have smooth 50 or 60 fps full sized video that will look great when played on a computer. That video can't be played on a TV however.

## Version History

### Version 2.5 (2010-07-18)

- When using future frames to fill in borders, you no longer need to manually pre-collect frames and delay the audio etc. Everything is now taken care of automatically, mostly by VirtualDub. This doesn't work for very old versions of VirtualDub, though. In fact, I strongly recommend version 1.9.9 or later.
- You can now turn off detection of rotation and/or zoom during pass 1. Read more under the '[Detect rotation/zoom](#)' setting above.
- When the setting *Remember discarded areas to next frame* is enabled, the discarded areas are now remembered for all processed frames, so that when you want to reprocess some frames with other settings during pass 1, Deshaker will know which blocks were discarded in the frame prior to the frame you're starting reprocessing at. This only works as long as the total number of blocks (horizontally and vertically) don't change, which can happen if you, for example, change the block size.
- The default edge compensation type was changed from *Adaptive zoom* to the more basic *None*, since I want you to decide for yourself if and how the borders should be taken care of. Also, the default smoothness values were changed from 400 (1500 for zoom) to 1000.
- The C++ and MFC libraries are now linked statically. (This means you shouldn't need to install the C++ Redistribution Package any longer.)
- Other smaller changes and bugfixes. For example, I fixed a bug which caused Deshaker to sometimes ignore large areas during pass 1 for certain (unusual) settings.

### Version 2.4 (2008-11-15)

- Added two new edge compensation zoom options; *adaptive+fixed* and *adaptive+fixed only*. Read more under the '[Edge compensation](#)' setting above.
- The constant zoom that is added for the fixed zoom edge compensation options is now constant only for each scene in a clip. It used to be constant for the entire clip.
- If you have a rolling shutter camcorder, you can now specify how severe this effect is for your camcorder model. Read more under the '[Camcorder has a rolling shutter](#)' setting above.
- The default location of the log file has been changed (due to write access limitations).

### Version 2.2 (2007-05-21)



- The most time consuming parts of Deshaker are now multi-threaded, which allows for significantly faster processing on systems with more than one cpu.
- Added an option to help ignoring large moving objects during matching. Read more under the '[Remember discarded areas to next frame](#)' setting above. This new option is enabled by default (but not if you load old settings files).
- Added an option to ignore motion vectors that are simply too long. Read more under the '[Discard motion of blocks that move more than X pixels \(absolute motion\)](#)' setting above.
- Improved handling of new scenes. When matching fails for a certain frame, the word 'new\_scene' is written to the logfile, which allows pass 2 to take appropriate action.
- Fixed a bug where some settings would change by themselves under some circumstances.
- Fixed a bug that was introduced in version 2.0 where the borders would always look a bit bad if you used previous and future frames to fill in the border but didn't enable soft borders.

## Version 2.0 (2006-08-12)

- Renamed the feature for improved border appearance that was added in version 1.9 to "Soft borders" and improved it by allowing soft edge transitions between frames in the border, which greatly reduces any sharp ugly edges there. Read more under the '[Soft borders](#)' setting above.
- Added an option to ignore dark pixels during matching. Read more under the '[Ignore pixels darker than X % brightness](#)' setting above.
- Deshaker should no longer crash when being used from AviSynth. But no motion vectors or info texts will be shown in the video due to a limitation in AviSynth.
- The info text in the output video during pass 1 now includes the percentage of ok blocks and whether or not deep analysis was used for the frame.
- The text of the deep analysis option was changed from 'Deep analysis if > X % of blocks discarded' to 'Deep analysis if < X % of vectors are ok', because it's really a percentage of all found vectors (not a percentage of all blocks). Note also that the value is treated differently. If you used value X before, you should now enter 100-X. This is taken care of automatically if you load old processing settings.

## Version 1.9 (2006-06-18)

- Added an option to improve appearance of the border when using previous and future frames.
- Added an option to extrapolate colors into the border. Read more under the '[extrapolate colors into border](#)' setting above.
- Added an option to always produce video with the same properties as the source video. Read more under the '[same destination properties as source](#)' setting above.
- Other minor changes, fixes and speed improvements.

## Version 1.8.1 (2006-02-11)

- Fixed a bug (that was introduced in version 1.8) where gaps in the log file would cause black frames.

## Version 1.8 (2005-08-30)

- Added correction for distortions caused by camcorders with a rolling shutter. Read more under the '[camcorder has a rolling shutter](#)' setting above.
- Added a new option for generating progressive video from an interlaced source. Read more under the '[generate interlaced progressive video](#)' setting above.

## Version 1.7.1 (2005-07-28)

- Fixed an old bug that suddenly showed itself (by crashing) when I tried VirtualDub-MPEG2 with an HDV clip.

## Version 1.7 (2005-07-26)

- Improved the algorithm that extracts panning, rotation and zoom from the motion vectors slightly. Previously, sometimes too few white vectors were found without a "good reason".
- Added a new "deep analysis" algorithm that makes the extraction of panning, rotation and zoom a lot more reliable in complicated clips. Unfortunately I couldn't get it as fast as I would have hoped and therefore it's not enabled in the default settings. Read more about this under the [deep analysis](#) setting above.
- Deshaker should no longer crash when running VirtualDub batch jobs (wasn't my fault...).
- You may now enter fractional numbers for the setting that discards motion of blocks that move > X pixels in

wrong direction. (I tend to use values below 1 a lot to really only follow the background, and also to make the new deep analysis algorithm find the background among moving people etc.)

- Added a new pixel aspect ratio preset for HDV 1440x1080. (HDV 1280x720 has square pixels, as does the HD resolution 1920x1080.)
- Modified the pixel aspect ratio for all NTSC variants very, very slightly. (Standard NTSC was changed from 72/79 to 4320/4739, which is the correct PAR according to [this page](#).)

### Version 1.6 (2003-11-24)

- Quick fix to make Deshaker work better with VirtualDub 1.5.5+ in interlaced mode too :)

### Version 1.5 (2003-11-22)

- Works better with VirtualDub 1.5.5 and later. (Before, log files could be emptied after pass 1.)
- Added a (pretty useless?) feature to help stabilizing on a moving object. Read about it under the '[ignore image area outside area](#)' setting.

### Version 1.4 (2003-06-13)

- Past and future frames can now be used to fill in the black borders. I recommend reading about these new settings [above](#) before using them. (Thanks to Lorin Thwaits for the idea to use future frames!)
- The settings window now fits 800x600 screen resolution (by using smaller font size when necessary).
- Now, you can enter -1 as smoothness to get "infinitite smoothness" :)
- Other minor changes.

### Version 1.3

- For some kinds of video (for example interlaced NTSC), pass 1 was unnecessarily slow. This issue was fixed.

### Version 1.2

- Added support for interlaced video.
- Added settings that limit the amount of corrections.
- Added new edge compensation option that completely eliminates the black borders
- Added an optional extra zoom factor to apply to the video.
- For some settings, Deshaker can now take quite a long time to calculate motion at the start of pass 2. So a little cancel button will now be shown during this calculation.

### Version 1.1

- Fixed a little bug when calculating rotation and zoom. You shouldn't use log files created with version 1.0. But hopefully you can use log files created with 1.1 in all future versions.

### Version 1.0

First release.

If you feel like giving me a little something for my efforts...



© 2010 Gunnar Thalin

[gunnart@guthspot.se](mailto:gunnart@guthspot.se)

[www.guthspot.se](http://www.guthspot.se)