

Frequently Asked Questions

- [How do I figure out the nature of my source video?](#)
 - [How do I figure out the field order of my video?](#)
 - [How do I install and use an external filter in VirtualDub?](#)
 - [Is there a filter to just shift an image over by a number of pixels?](#)
 - [The flaXen site is down. Can you send me the VHS filter?](#)
 - [I want to apply a filter to my interlaced video but the filter does not have an option for interlaced source video. What can I do?](#)
 - [My camcorder has a stuck pixel. Is there a filter that can remove it?](#)
 - [Is there a filter to make my clip play in reverse?](#)
-

How do I figure out the nature of my source video?

It's important to know whether your video is progressive, interlaced, telecined, field-blended, etc., when making decisions about how to process it. Here's a reliable method for analyzing your video:

The first step is to create an Avisynth script that sources your video and separates the fields. If you have MPEG video, you can use DGIndex to make a project with "Honor pulldown flags". Then make a script to step through the fields:

```
MPEG2Source("file.d2v")  
AssumeTFF() # or AssumeBFF(). See the next FAQ.  
SeparateFields()
```

If you have a different video source type, just use an appropriate source filter, such as AVISource().

Now serve the script into VirtualDub and step through the fields in a high motion section. We will label a new picture with a new letter below. If you see a repeating field sequence like this:

... a a b b c c ...

then it is progressive video. If you see this:

... a b c d e f ...

then it is pure interlaced video. If you see this:

... a a a b b c c c d d ...

Then it is 3:2 pulled-down progressive video.

Make sure that you always apply this test in a scene with continuous movement, e.g., an object moving steadily across the screen. There is no more reliable way. It also allows you to spot field blending very easily. If you see any blended pictures, then you have field blending.

How do I figure out the field order of my video?

It is essential to set the field order properly for correct rendering of interlaced material. If you are using Decomb, the field order is obtained from Avisynth and can be set using the AssumeTFF() or AssumeBFF() functions in your script before invoking Telecide(). Set AssumeTFF() for top field first; use AssumeBFF() for bottom field first. Because setting it correctly is so important, you are strongly encouraged not to make assumptions about the field order of a clip, but rather to verify the field order using the following procedure, which is a variant of the procedure described in the FAQ preceding this one.

To determine the field order, make an Avisynth script that serves the raw clip and separates the fields. If it were an AVI, then just AVISource() would be used. For our example, we'll use AVISource(). Add a script line to separate the fields using top field first, as follows:

```
AVISource("your_clip.avi")
AssumeTFF()
SeparateFields()
```

Now serve the script into VirtualDub and find an area with motion. Single step forward through the motion. Note whether the motion progresses always forward as it should, or whether it jumps back and forth as it proceeds. For example, if the field order is wrong, an object moving steadily from left to right would move right, then jump back left a little, then move right again, etc. If the field order is correct, it moves steadily to the right.

If the motion is correct with AssumeTFF().SeparateFields(), then your field order is top field first and you must use AssumeTFF(). If the motion is incorrect, then your field order is bottom field first and you must use AssumeBFF(). If you want to double check things, you can use AssumeBFF() to check correct operation for bottom field first.

Additionally, if the clip is progressive, motion will be correct for both AssumeTFF() and AssumeBFF(), i.e., field order does not matter.

How do I install and use an external filter in VirtualDub?

Following is a step-by-step procedure for using the filters.

1. First ensure that the filter .vdf file is present in a directory called "plugins" that exists in the same directory as the VirtualDub executable.
2. Start up VirtualDub and maximize its window.
3. Select File/Open video file from the pulldown menus and then open your input video file.
4. Select Video/Filters from the pulldown menus.
5. Click on the Add button in the Filters dialog box.
6. Scroll down in the filter list and highlight the desired filter by clicking on it.
7. Click OK. The configuration dialog for the filter appears.
8. Configure the filter as desired. If the filter supports preview, you can interactively explore the effect of different filter settings. When you are happy with the settings, click OK in the filter configuration dialog box.
9. Click OK in the Filters dialog box.
10. Select Video/Compression from the pulldown menus.
11. Highlight the desired compression for the output file and then click OK.
12. Select File/Save As from the pulldown menus and select the desired output file. Processing will begin and when it is completed, the output file will contain the filtered video.
(Alternatively, you can simply preview the result by using VirtualDub's timeline player.)

Is there a filter to just shift an image over by a number of pixels?

This can be accomplished using existing VirtualDub internal filters. For example, suppose you have a 720x480 clip that you want to shift down by 5 pixels. First, load a *resize* filter and set its *new width* and *new height* to the size of the input clip, i.e., 720x480. Then click the *Expand frame and letterbox image* checkbox and enter 720x490 as the frame width and frame height. Finally, select the fill color that you want and save this filter. Now we load a *null transform* filter and configure its cropping to remove 10 lines from the bottom of the frame. Note that the preview will not look right due to a quirk in VirtualDub, but ignore that and just set it to crop the bottom 10 lines. The output of this filter chain is the shifted image with the exposed area at the top filled with the selected color.

The procedure is easily generalized to any required shifts or combinations thereof.

UPDATE: Simon Walters now has a filter that makes this process easy: [ShiftWrap](#).

The flaXen site is down. Can you send me the VHS filter?

Pending further developments at the flaXen site, I have hosted two of his filters in the Hosted Filters section. You can get the VHS filter there.

I want to apply a filter to my interlaced video but the filter does not have an option for interlaced source video. What can I do?

That's an easy one. First apply VirtualDub's internal deinterlace filter set for "Unfold fields side-by-side". Then apply your filter. Then apply VirtualDub's internal deinterlace filter set for "Fold side-by-side fields together again".

My camcorder has a stuck pixel. Is there a filter that can remove it?

Here a little creative thinking is required! Think of your stuck pixel(s) as a logo that you want to remove. Sure, it is a very tiny logo, but that doesn't matter. Given this thinking, we can now select Chris Wojdon's LogoAway filter (available through the Other section of this site) and use it to replace the stuck pixel(s) with nearby pixel values.

Is there a filter to make my clip play in reverse?

It can't be done with a VirtualDub filter, but Avisynth has a built-in Reverse filter. Get Avisynth [here](#).