

# **Conversion of time-lapse sequences of still images to movie files**

**(step by step guide, freeware solutions)**

**Martin Setvák**

[setvak@chmi.cz](mailto:setvak@chmi.cz)

Version: 31 March 2008

## Preface

I was motivated to write this “guide” by two factors. First is the recent (March 2008) “Ricoh-boom” among my friends and colleagues, namely meteorologists and weather enthusiasts, so this should help them process their time-lapse sequences made by these cameras (but not only these). The second reason is that occasionally I see a time-lapse movie of rather lower technical quality (which applies also to some of my older movies), which could have been much better if the author knew how to do it and what settings to use when processing the sequence he or she has made. Some of the steps described below might appear so obvious that most of you will ask “*why does he describes this - it is so clear that he might have skipped this*” ... but I was trying to help even the beginners, so excuse the “step by step” nature of the guide.

At the beginning I wanted to prepare a simple web page, but while preparing the screenshots and realizing that this was going to be much larger than I originally thought, I changed my mind and decided to prepare a PDF version instead. The advantage of this form is that you can download or print it, and have it at your hand whenever needed.

This guide describes the software I prefer and use; all of it is freeware (for non-commercial use). Obviously, there are other alternatives - perhaps even some which can handle all three steps described below at once - without a need to jump from one software package to another to complete all the tasks. I've tried several alternatives, but the results were not as good as from the software described below.

I do not discuss here the basics and methods of time-lapse photography (alternatively called interval photography) and camera settings for this; you can find these elsewhere, e.g. on my time-lapse web page (<http://www.setvak.cz/timelapse/>). Also, I do not go into the details of post-processing the original images here, or their preparation for conversion to a time-lapse movie. There is host of software which can be used for that purpose; I personally use Adobe Photoshop CS3 for this (namely its “Actions” and batch processing).

Finally, I have to apologize to Linux and Mac OS users – most of the software below runs under Windows only and doesn't have its versions for other OS; the only exception is the *MPEG StreamClip*, available also for Mac OS. Hopefully, at least the principles and some of the comments below will be helpful to you when experimenting with other available software, under different OS.

## Contents

Introduction .....	page 03
Step 1 – from a sequence of images to an uncompressed AVI movie file .....	page 03
Step 2 – suppressing the flicker of an uncompressed AVI file .....	page 09
Step 3 – converting deflicked AVI movies to final compressed files .....	page 16
- MPEG-1 files .....	page 16
- XviD encoded AVI files .....	page 19
- H.264 encoded QuickTime MOV files .....	page 21
Playing the movie files .....	page 25

## Introduction

At the beginning you have a sequence of images captured in time-lapse (interval) mode by a still digital camera or a webcam. First, you have to prepare these images to an image format (using e.g. brightness and color adjustments – if needed, crop and resize these, etc.), which you will use next to create the movie file. When finished with this, you have to assemble this sequence of images to a single movie file, which is described in Step 1. Since almost every time-lapse sequence suffers to a certain degree from a “flicker” (more on this below, in Step 2), you have to suppress it to get a “smooth” movie file. This is described in Step 2. At the end, you want your final movie file to have the best picture quality possible, and at the same time not be too large. This is done by movie file compression, using various formats and codecs, and is described here for three options (MPEG1, XviD AVI, and H.264 MOV) in Step 3.

### Step 1 – from a sequence of images to an uncompressed AVI movie file

Before you begin, you have to post-process your captured still images into a final format and size you wish to use for the movie file. Since some of the video codecs you might use for movie compression (Step 3) require the image size (width and height) to be a multiple of 16 or 32, you should keep this in mind when downscaling and/or cropping the original images.

To convert a series of images captured in time-lapse (interval) mode to an uncompressed AVI file, I use the **RAD Video Tools** software (<http://www.radgametools.com/binkhlp2.htm>); the latest version is available from <http://www.radgametools.com/down/Bink/RADTools.exe>. The screenshots used below are from RAD Video Tools, version 1.9e.

There are other software utilities or packages which enable this step, but among the freeware which I have tried, I consider the RAD Video Tools as the most convenient, straightforward and easy to use.

-----

#### Note on the movie file size (2 GB, 4 GB, and above)

If you attempt to create a movie file from too many images and the final file size exceeds 2 GB (or 4 GB), you may encounter certain problems or error messages. The 2 GB size is a limit of the “old” AVI format, and software which uses this old AVI standard simply can’t create larger AVI files. To overpass this problem, some of the new versions of video-editing software already utilize the “OpenDML AVI” standard, which basically has no file size limit (e.g. VirtualDub). 4 GB limit is a restriction of the FAT32 file system (e.g. Win98). NTFS (e.g. Windows XP/Vista) and Mac OS have no individual file size limit.

These file size problems can be over-passed by three methods:

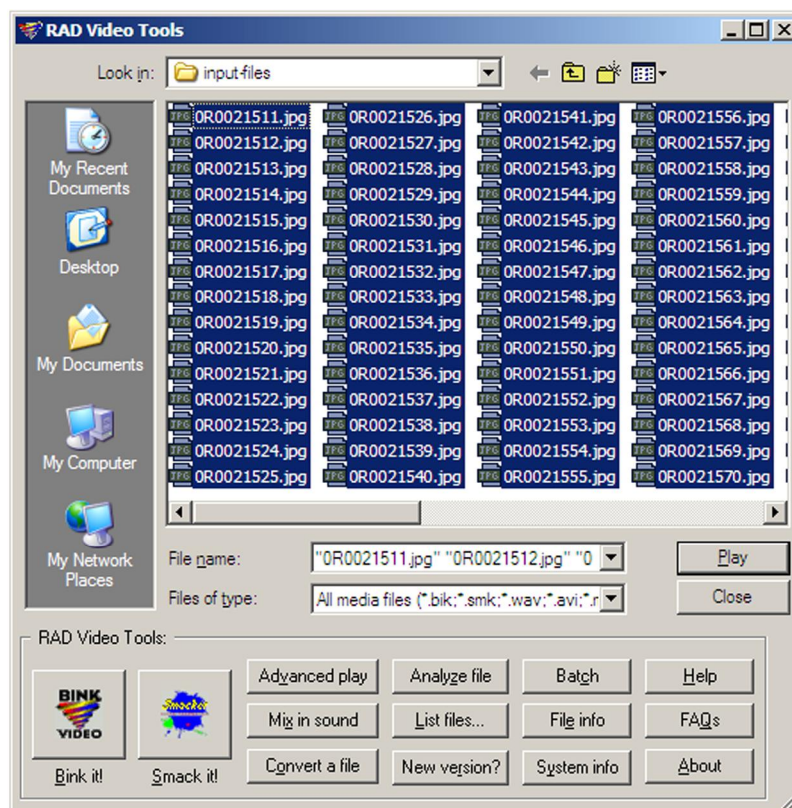
- Create two or more smaller AVI files, using only subsets of the images, each AVI file under 2 GB, and merge these in VirtualDub afterwards (Open file > Append file > Save AVI file), before proceeding with Step 2.
- Create one larger file (between 2 and 4 GB), and open it next in VirtualDub. You will get a message about errors found in the file, and VirtualDub will try to correct the problem (“reconstructing missing index block”). Save the new AVI file afterwards, the file should be OK now and you can proceed to Step 2.

- Finally, you can create a compressed (thus smaller) AVI file, and apply a deflicker filter on this one. However, any compression – decompression – compression might spoil the quality of the final movie, so I prefer to apply the deflicker filter to non-compressed AVI files.

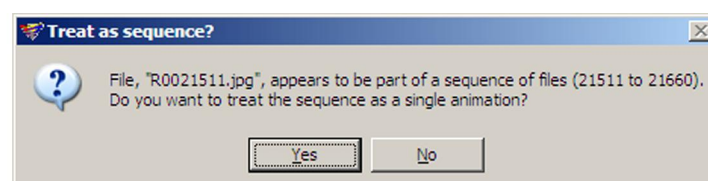
The first method produces better results than the second one, which may end up jumpy and not play smoothly. I have never tried the third method, so I can't compare or comment about it.

A recent version of the RAD Video Tools uses the old AVI standard, so you can create flawless files up to 2 GB only. Actually, the software can create AVI files up to 4 GB, but these are partially corrupt and you need to restore them using VirtualDub (see above). If the file size should exceed 4 GB, the software will restrict the size to 4 GB and the file is corrupt and unusable.

After starting the RAD Video Tools, go to a directory where you have placed your sequence of images prepared for the movie file. Select the files you want to use for the movie file (e.g. using CTRL+A for all the files in the directory), and you should see a screen like this:



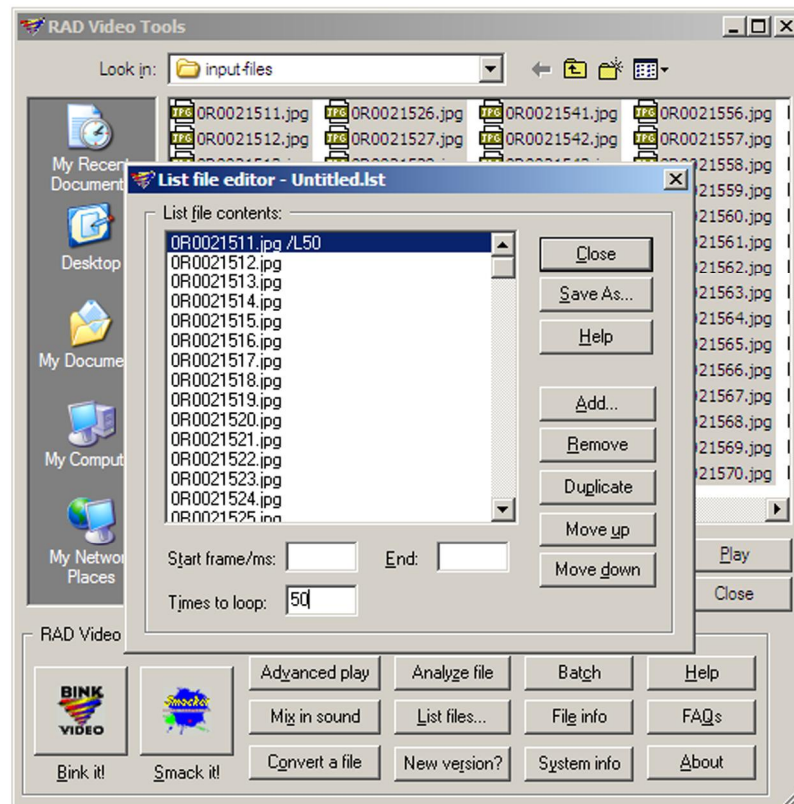
Next, click the **List files ...** button, which will take you to the “List file editor” window (next page). Occasionally, you may get the following message:



In such a case you have to confirm each of the images by clicking the **Yes** button, which is quite disgusting (the only way out is to confirm each of the files, or to kill the software from the Windows Task Manager).

To avoid this problem, rename the source image files first; in the case of images from the Ricoh cameras just add any digit in front of the “R” in their default file name (e.g. zero).

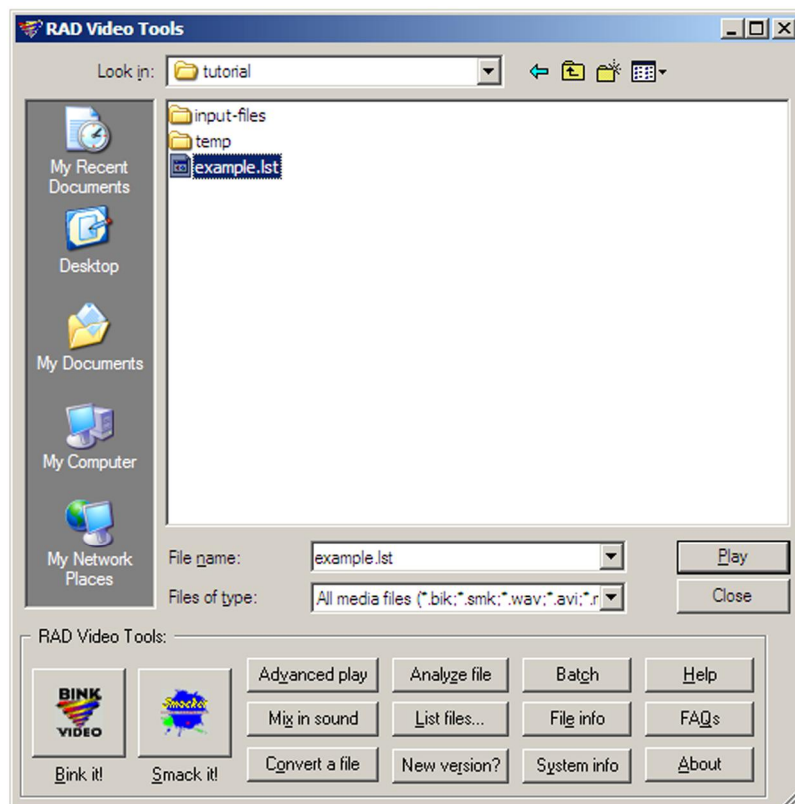
If everything goes well, after clicking the **List files ...** button you should see the following screen:



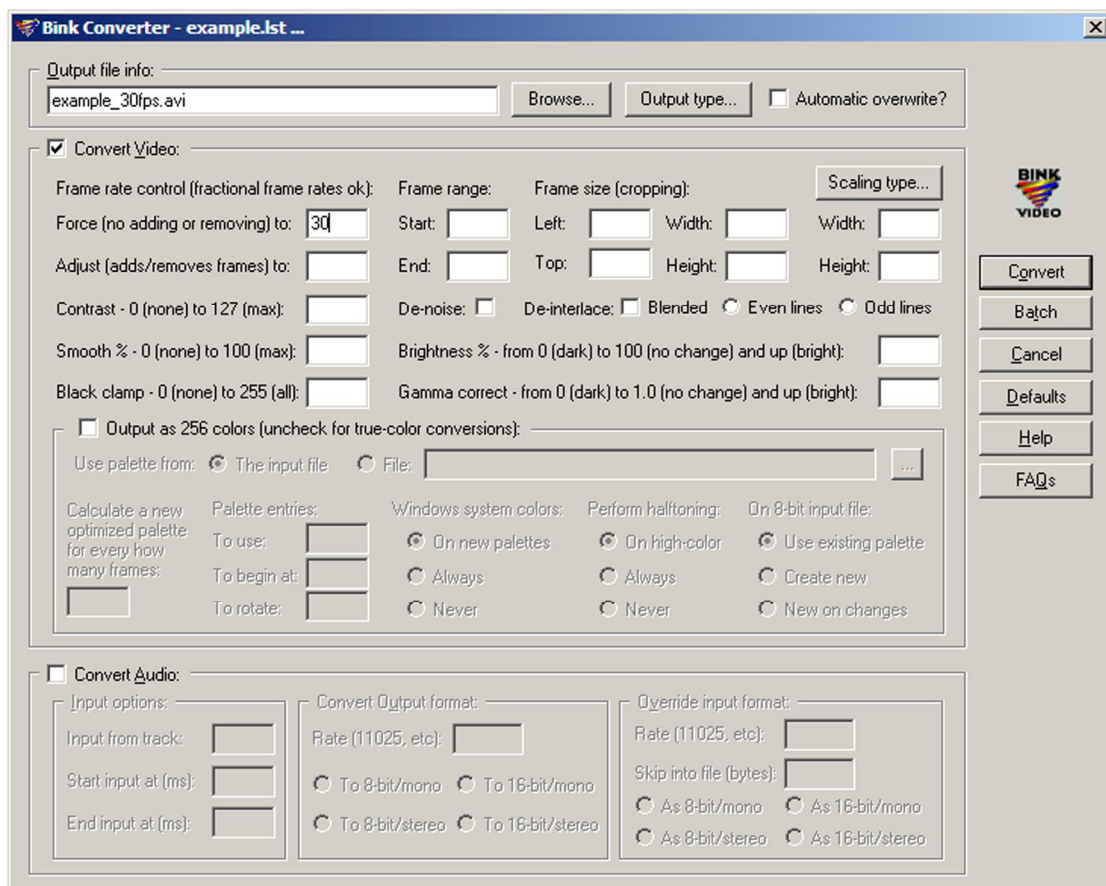
Here you can specify (among others) how long the first and last image of the movie should stay “frozen” on the screen when playing the movie. For example, if you plan to use a frame rate of 30 frames per second (fps), and want the first image (including e.g. your titles of the movie) to stay on the screen for 2 seconds before the movie starts, set the “Times to loop” for this frame to “60”. Similarly, you can set this for the last frame of your sequence (at the bottom of the list above).

When done specifying the parameters of the first and last image, save your file list by clicking the **Save As...** button. It is convenient to save the \*.lst file into a different directory than where you have your sequence of images. Btw, \*.lst stands here for any name you choose for the file list, just keep the “lst” extension.

Next you have to use the file list you created (\*.lst) to convert your sequence of images to a movie file. Locate your \*.lst file and highlight it (next screenshot, top of the following page). You can always change the saved settings (e.g. timings of the first and last frames) by simple editing the \*.lst file, or by double-clicking it here.



Highlighting the \*.lst file and clicking the **Convert a file** button opens the following screen (“Bink Converter”):

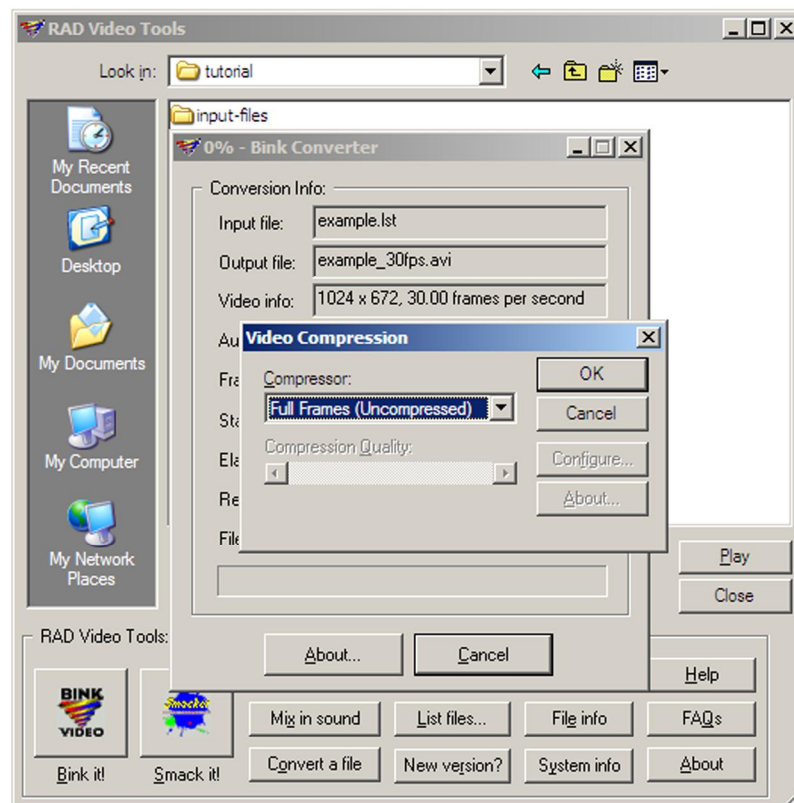




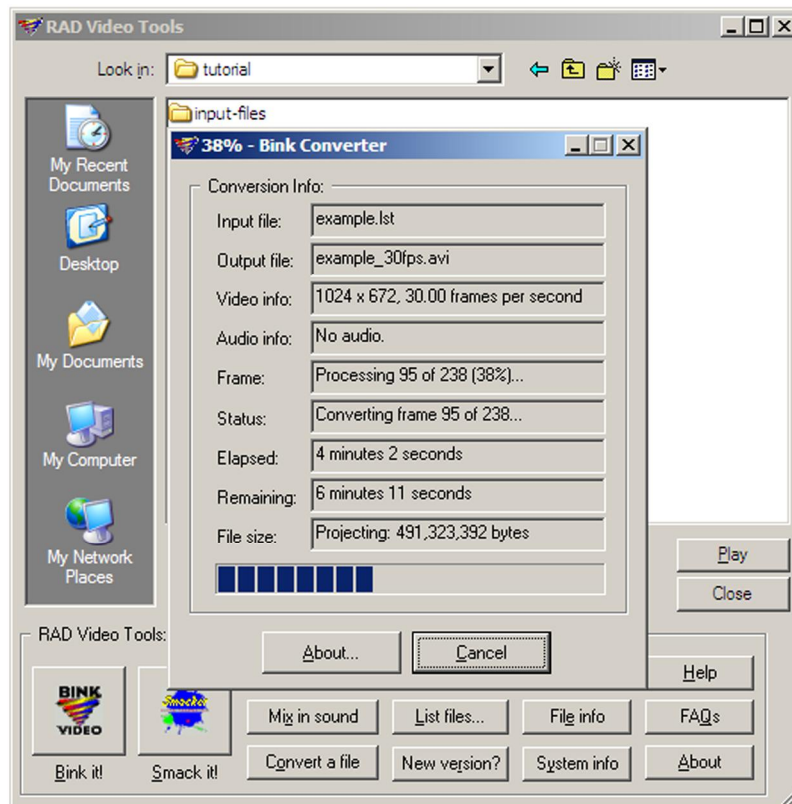
Here you can select the output movie file name (top left of the window), and most importantly the frame rate of the output movie file. The frame rate has to be specified in the “*Force (no adding or removing) to:*” window, in the example above it was set to 30 fps. This value determines how quickly your time-lapse movie will be played, what the speed (or acceleration) of the movie will be as compared to reality. Please read the note on frame rate on page 17: not all frame rates are supported by TMPGEng which you may wish to use for conversion to an MPEG-1 file.

*As an example, let’s consider a case in which you have taken a time-lapse (interval) sequence with a camera, shooting one image every 5 seconds. If you set the frame rate of your output movie file to 30 frames per second (fps), your movie will show 30 time-lapse images in one second, which is a period of  $30 \times 5 = 150$  seconds of “real time”. Since you will play these 150 seconds of “real life” in 1 second, your movie runs 150x faster than reality...*

You don’t have to fill in anything else here - just press the **Convert** button, which opens the following window (occasionally it takes some time to open the next window, so just be patient here):



Since you will most likely need to “deflicker” the movie file next, using VirtualDub’s Deflicker Filter (Step 2), leave the movie uncompressed for now, as shown on the screenshot above. However, if you feel that the movie file doesn’t need any further adjustments, you can alternatively select one of the available compression methods available here under the “Compressor” pull-down menu. After selecting the compression method or leaving the file uncompressed (**recommended!**), just click the **OK** button, which opens the following screen (next page). You can follow the progress there, until the “Done!” message appears when finished.



Congratulations! You have just assembled an uncompressed AVI movie file, ready for further processing via the next two steps.

---

**Note:**

On most of the ordinary recent computers (depending on CPU performance, memory size, and video card) you may have problems with playing the large uncompressed AVI files you have created in Steps 1 and 2 - don't worry about this, this is not an error. If you are curious how your movie does look, or just to check the movie speed (fps rate) you have chosen, you might need to compress the movie file first, using e.g. one of the possibilities described in Step 3.

Actually, I usually check the movie settings and its quality first, before proceeding to Step 2, by exporting a "preview file" from the RAD Video Tools to XviD compressed file (pages 7 and 19). Only after I'm satisfied with all the settings, I create the uncompressed AVI file and proceed to Step 2.

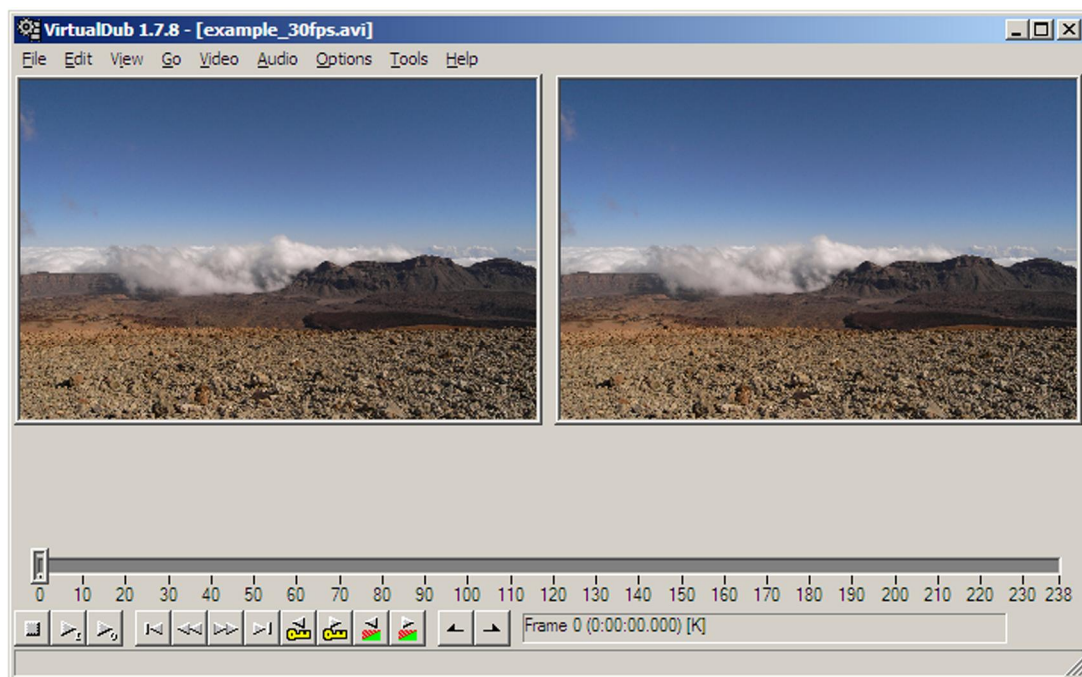


## Step 2 – suppressing the flicker of an uncompressed AVI file

Even if you take your time-lapse (interval mode) sequence with a camera set to manual exposure mode, pre-set the white balance to some defined option (e.g. “Cloudy”), and even when the lighting conditions do not vary during the sequence, some of the images will tend to “flicker” , i.e. change their overall brightness or color tone slightly. Typically, flicker is most evident on blue sky, but affects the entire image. When you display the images slowly, one by one, you may not even notice these changes, but if you view these as a fast slide show or convert these into a movie file, this flicker becomes quite annoying. It may either be a result of a certain “instability” of the camera shutter speed (minor variations of time the shutter remains open), or a consequence of conversion of the original raw file (captured by sensor of the camera) into the output JPG file, performed by the camera software. Of course, if you use a camera which doesn’t have the manual exposure mode, or if you switch to the auto-exposure mode because of some specific reason, the flicker becomes more evident.

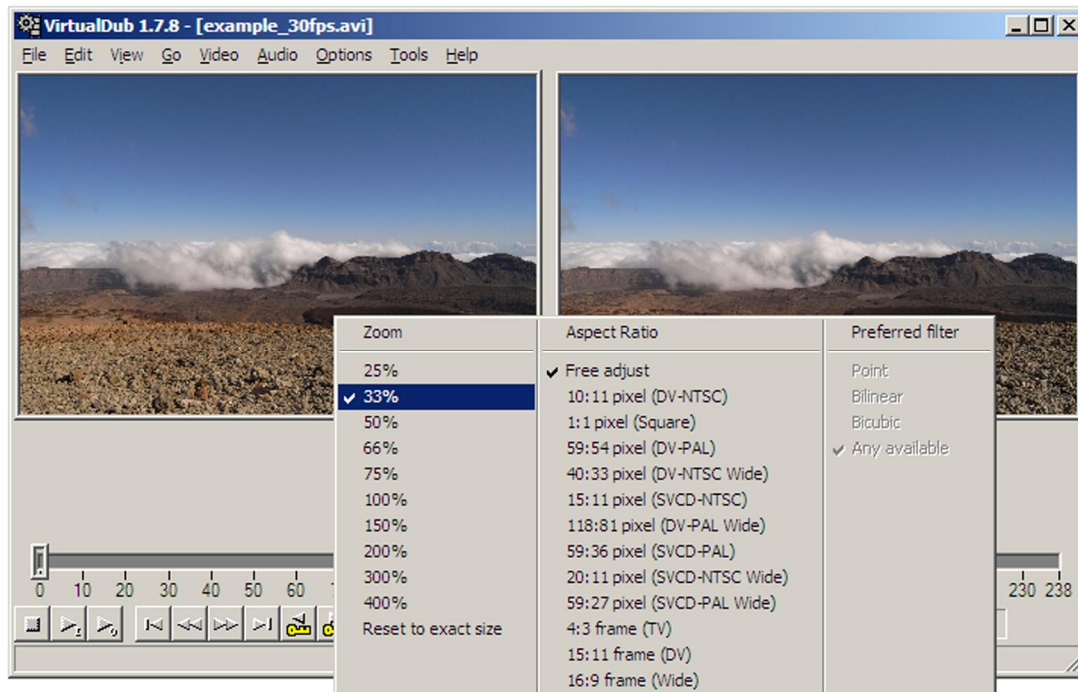
To suppress the flicker, download and install **VirtualDub** (<http://virtualdub.sourceforge.net/>) and its plug-in filter, Donald Graft’s **Deflicker Filter** (<http://neuron2.net/deflick/flick.html>), both freeware. The screenshots below are based on VirtualDub version 1.7.8, and Deflicker Filter version 1.3 beta 1. To be able to use the filter, just drop the filter file into the VirtualDub’s “plugins” folder.

After opening VirtualDub and opening your AVI movie file, you will see a screen like the one below, the layout of which strongly depends on image size of your movie file and your monitor setup.

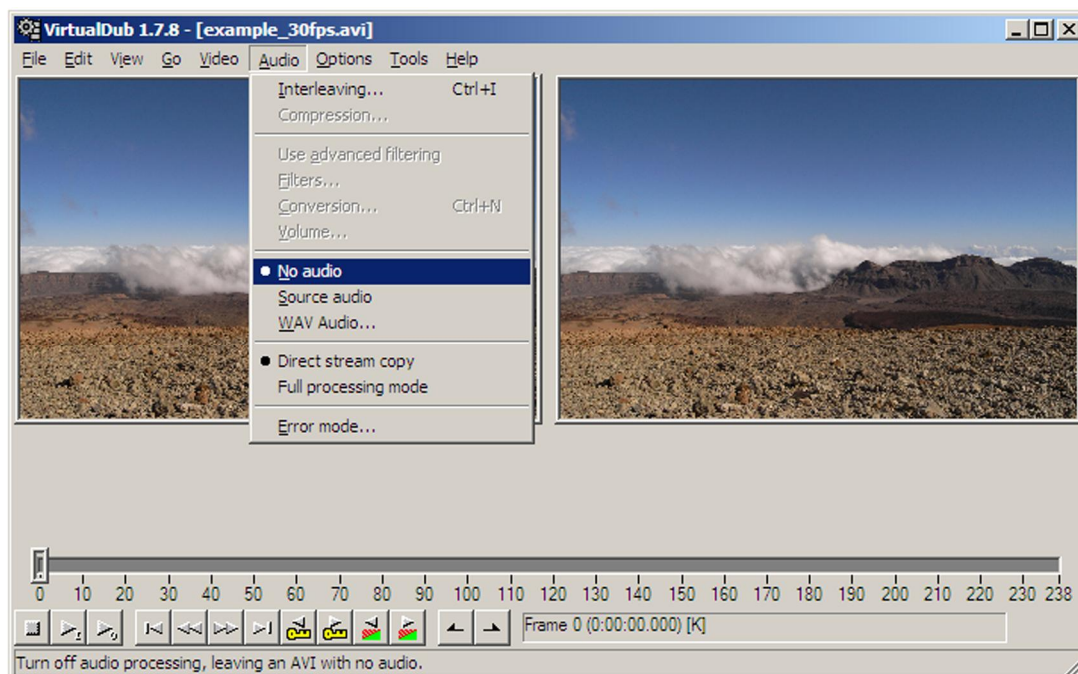


If you get a message here about a corrupted file, or if the software begins “*reconstructing missing index block*”, please read the note about file size (2 GB/4 GB issues) on page 3.

You can change the size of the two image windows (left window – source file, right window – output file) by right-clicking on those windows and setting the zoom factor for each of these. However this is not important - you can have basically anything on the screen; this will not impact your output file.

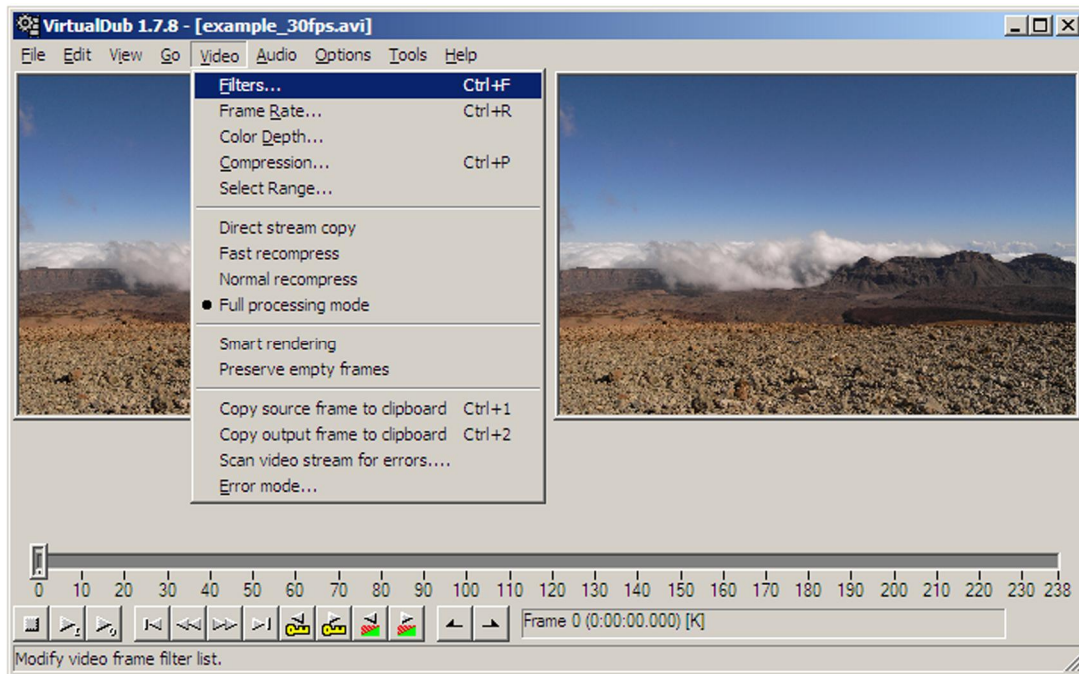


Before you proceed, you can check the audio settings; since your movie probably doesn't contain any sounds at this stage, just check the "No audio" option:

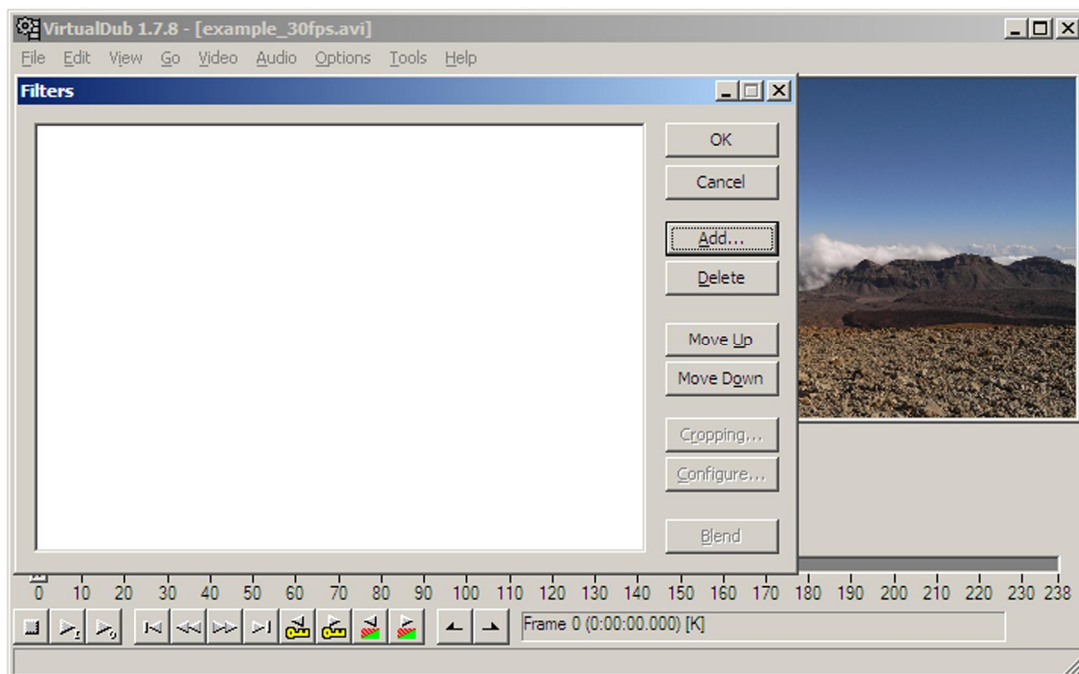


And now comes the most important part of all of this: removal (or more accurately suppression) of the movie flicker. If you are convinced that your movie doesn't flick at all or you don't care about it, you can skip this and go directly to Step 3.

To open the Deflicker Filter, go to "Video", and from there open the "Filters..." menu:

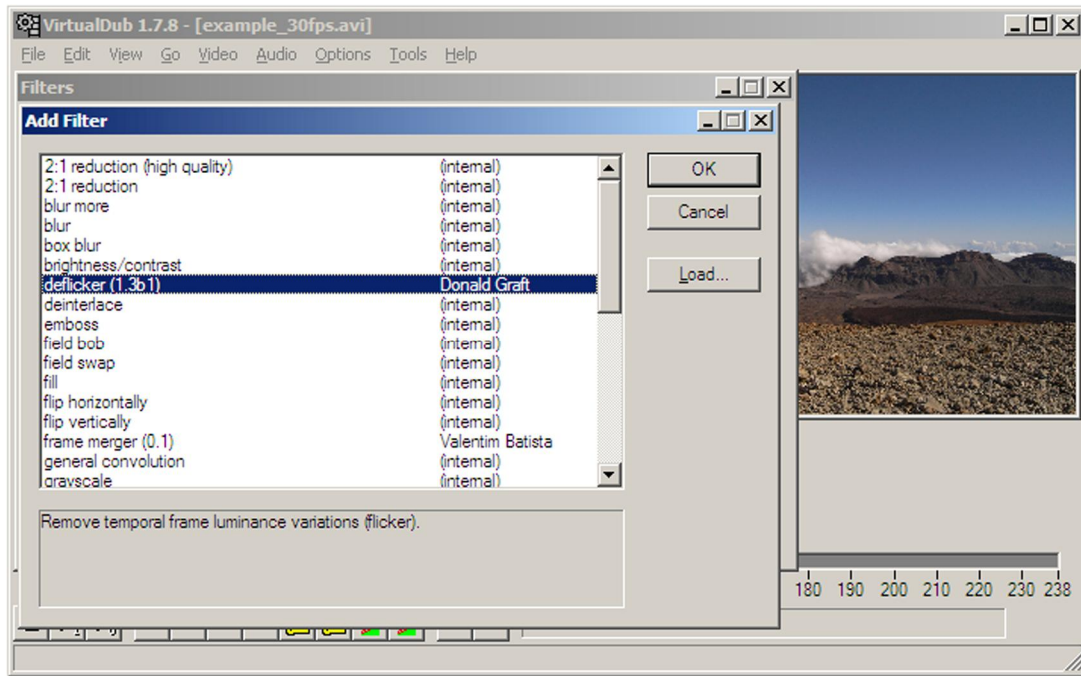


Now you should see the following empty window:

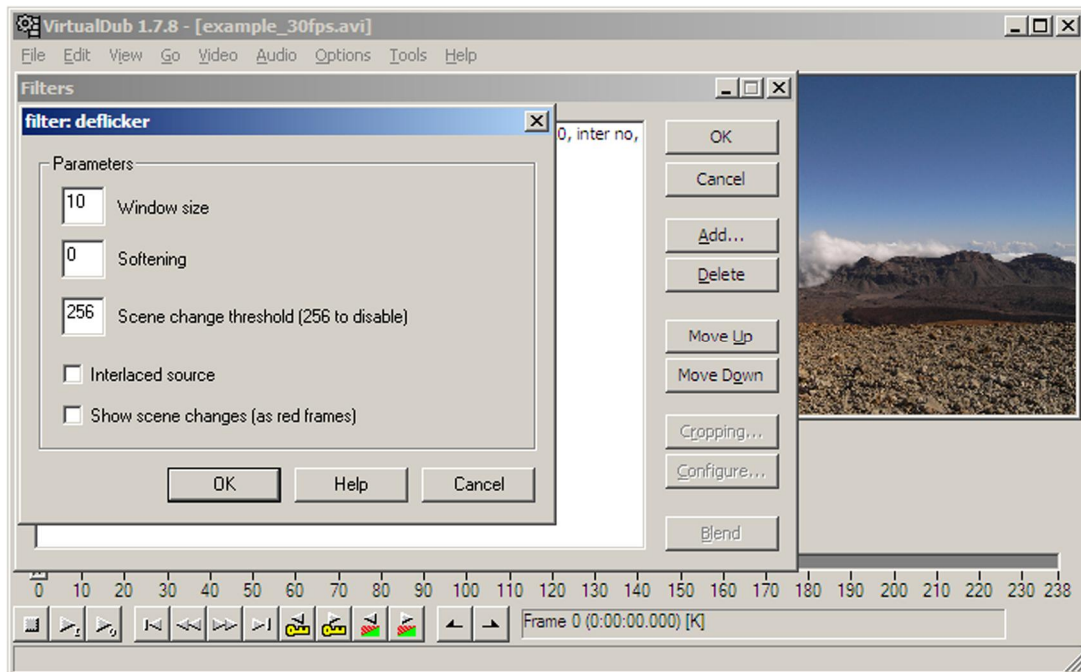


Click the **Add** button here, it opens the following window (next page).





Select the “deflicker (1.3b1)” filter from Donald Graft here, and click the **OK** button (or double-click the filter name). You should get a window similar to this one:



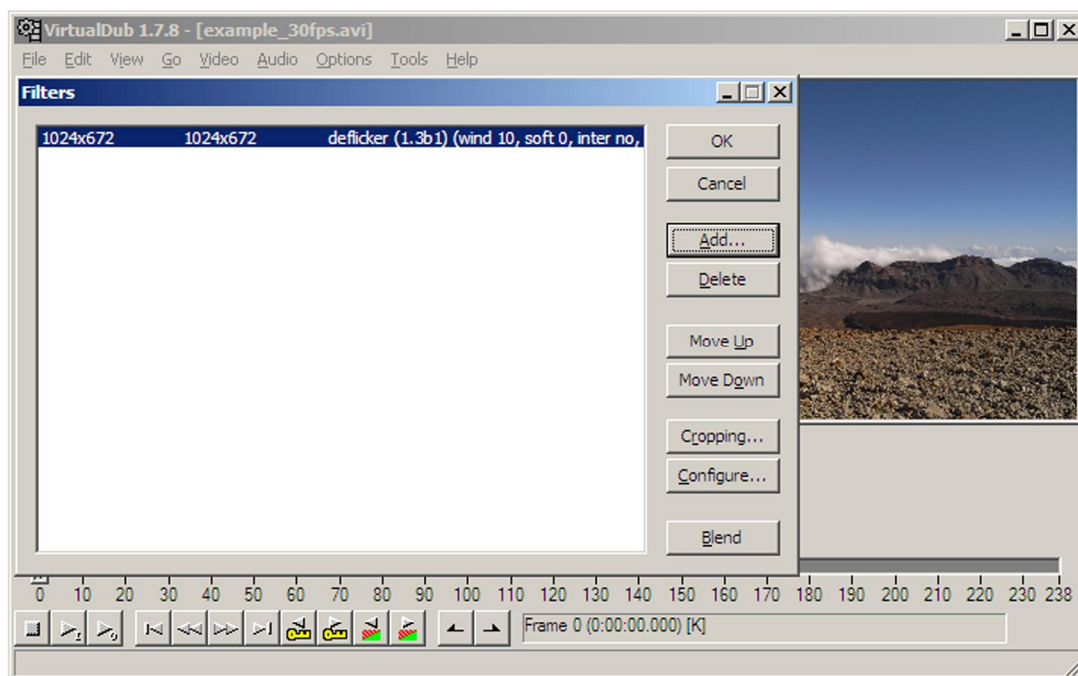
Setting the “Window size” and “Softening” parameters of this window is the trickiest part of all, with a great and crucial impact on the final movie, so let’s discuss these in more detail now.

The argument in “Window size” defines how many movie frames (your source time-lapse images) are used as a moving-window to determine the proper luminance of the actual frame. The larger you set this window size value the more previous frames are used to compute a luminance trend, which is applied next on the actual frame (provided that this frame is not a scene change frame). If you have large “flicker density” (number of bad frames per some standard number of frames), you have to choose a higher window size value. If your time-lapse sequence has only few bad frames, you can experiment with the lowest window size value possible. For both Kodak P880 and Ricoh GX100 cameras, I typically use a window size value from 5 to 10.

“Softening” offers a possibility to smooth the movie when big inter-frame scene changes are involved ... e.g. fast moving clouds shot at too long intervals (a typical problem of Kodak’s P880 basic 10 s interval in time-lapse mode). While smoothing cloud motions, the rest of the scene remains sharp. However, do not set this parameter higher than needed – values too high lead to the “motion-trails” effect. An optimum value for clouds shot at 10 s intervals with a 24 mm lens is about 5, while for a 5 s interval taken with the same lens you can leave this argument at zero.

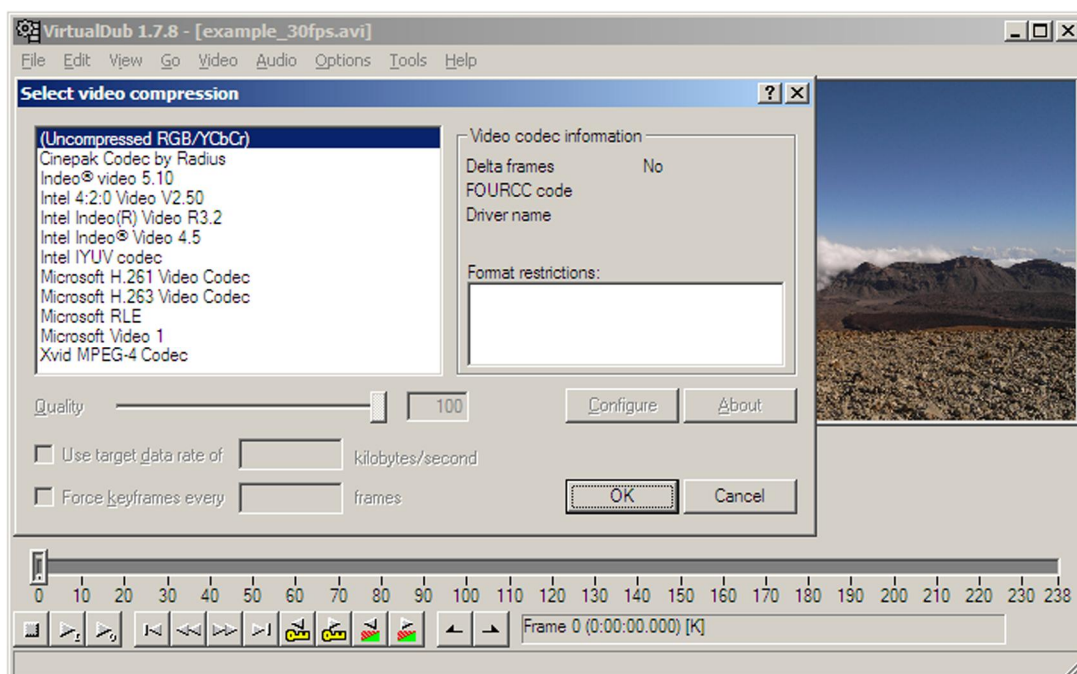
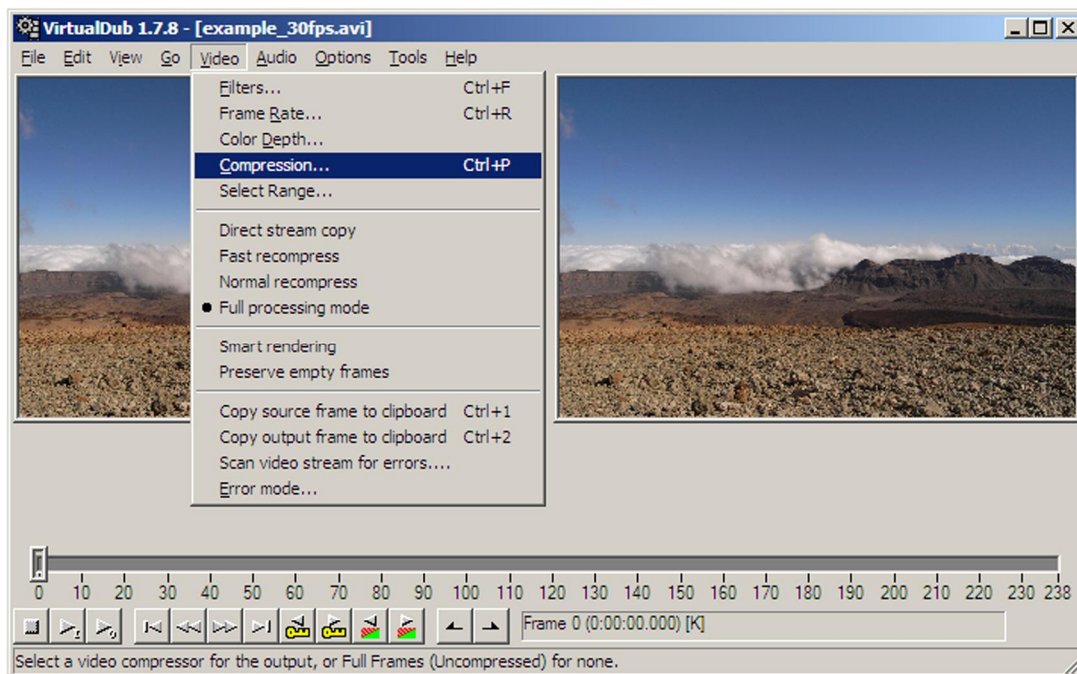
The screenshot above shows a typical window size and softening values for Ricoh cameras set at 5 s interval; you can leave the rest of the settings at their defaults.

After you are done with the settings above, confirm these by clicking the **OK** button, this will take you to the following window:



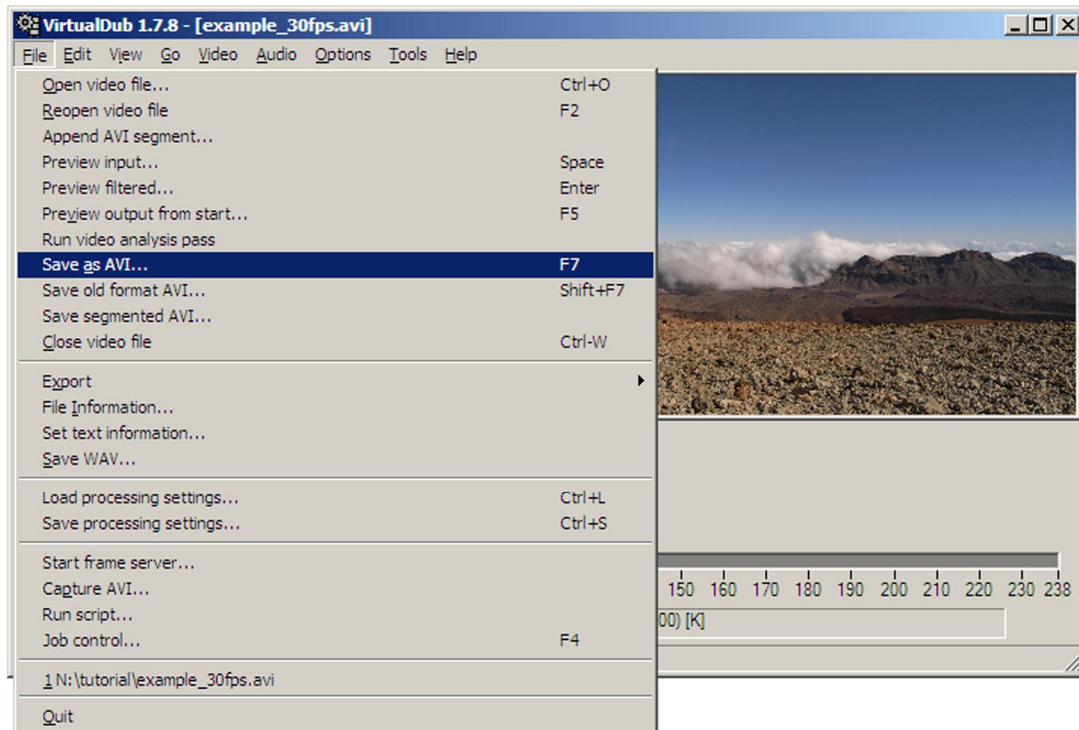
If needed, you can add more filters here. Since I prefer to fine-tune the images in advance, I usually don’t add anything more here. Clicking the **OK** button takes you back to the main VirtualDub screen.

Before saving the AVI file, you can decide now about the compression of the file. If you want to save it as a compressed file, go to Step 3 (XviD AVI file, page 19). If you prefer to keep an uncompressed file for now, save the AVI file with “Compression...” settings as shown below:

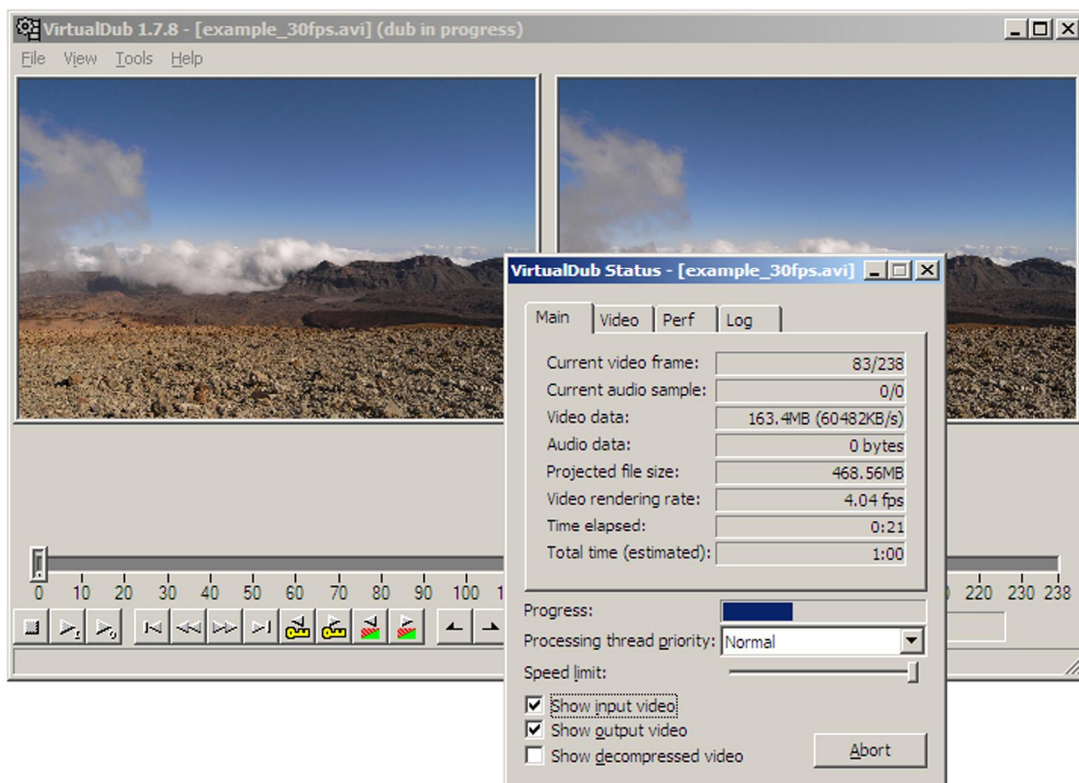


When done with the compression settings (default is “Uncompressed”, so if you have not done any changes to this before you can just skip the two screens above), proceed to “File” and “Save as AVI...” there (next screenshot):





After specifying the output file name you can follow the progress of saving the file on the “VirtualDub Status” window; ***the deflicker filter is applied now, while saving the file.***



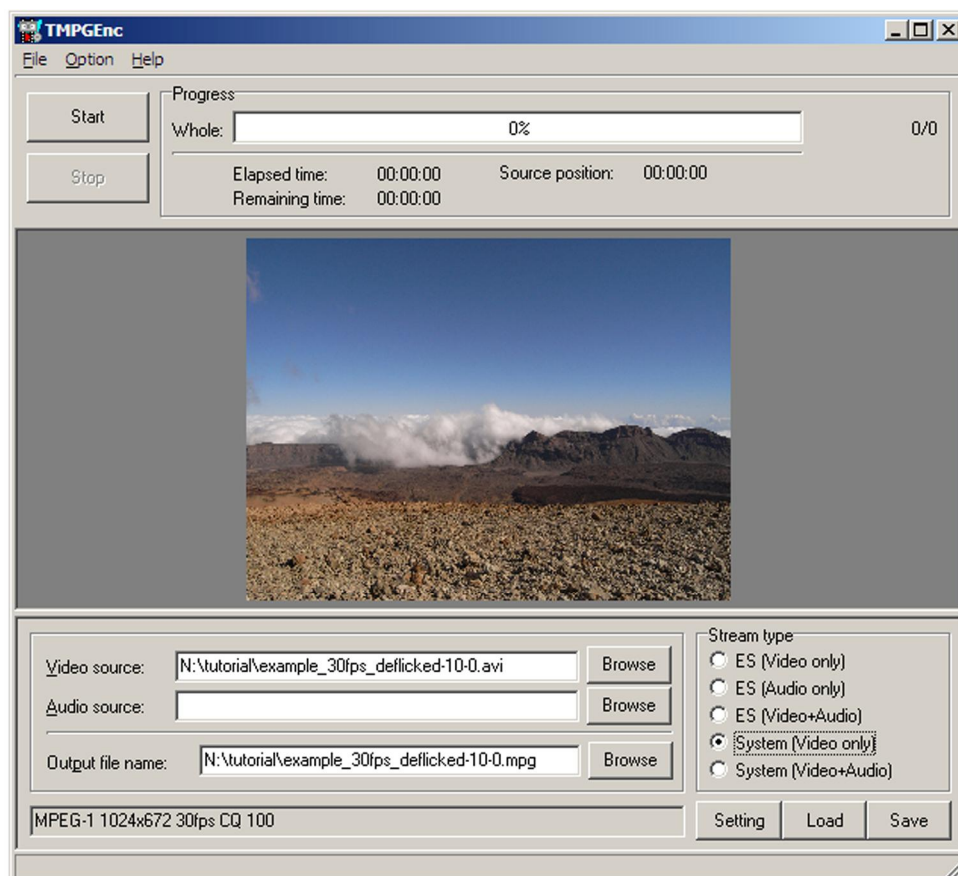
That's it, now you have saved a deflicked movie file, and can proceed to Step 3.

### Step 3 – converting deflicked AVI movies to final compressed files

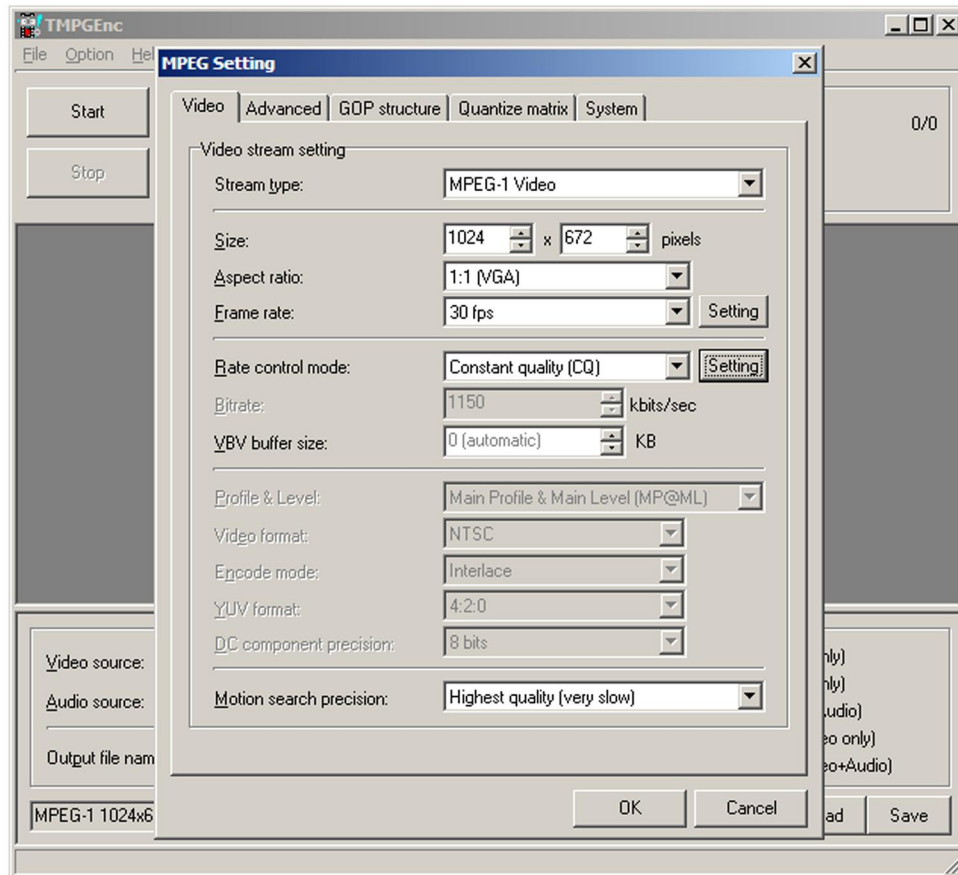
When preparing the final movie, there is always a big dilemma: size or quality? In general, the better quality you choose, the larger the file is. So the main task of this final step is to find a certain balance or compromise between the compression efficiency (to create a file as small as possible) and movie quality (to preserve as much as possible the quality of the original time-lapse images, without introducing too many compression artifacts). Moreover, some of the more advanced compression methods demand higher performance of the computer to view the movies compressed by these, so you may not be able to display the good quality movie on every computer... For these reasons I typically use three types of compressed files: MPEG-1 format which should be playable on almost any computer and any OS, but is not as good quality as the next two – which are the XviD compressed AVI file, or H.264 compressed QuickTime MOV file. Personally, I prefer the H.264 MOV files; they offer the best quality while still keeping the file at reasonable size. Not to “discriminate” the owners of older or weaker computers, I use on my time-lapse web pages (starting with 2007) both – the lower quality MPEG-1 file, and the same movie in H.264 MOV format. Those who can play the better MOV files should not bother with the MPEG-1 files. Notes concerning the movie players are at the end of this guide.

#### MPEG-1 files

To convert an uncompressed AVI movie file to the MPEG-1 format, I use the **TMPGEnc** free encoder, version 2.524 (<http://www.tmpgenc.net/en/download.html>, screenshot below):



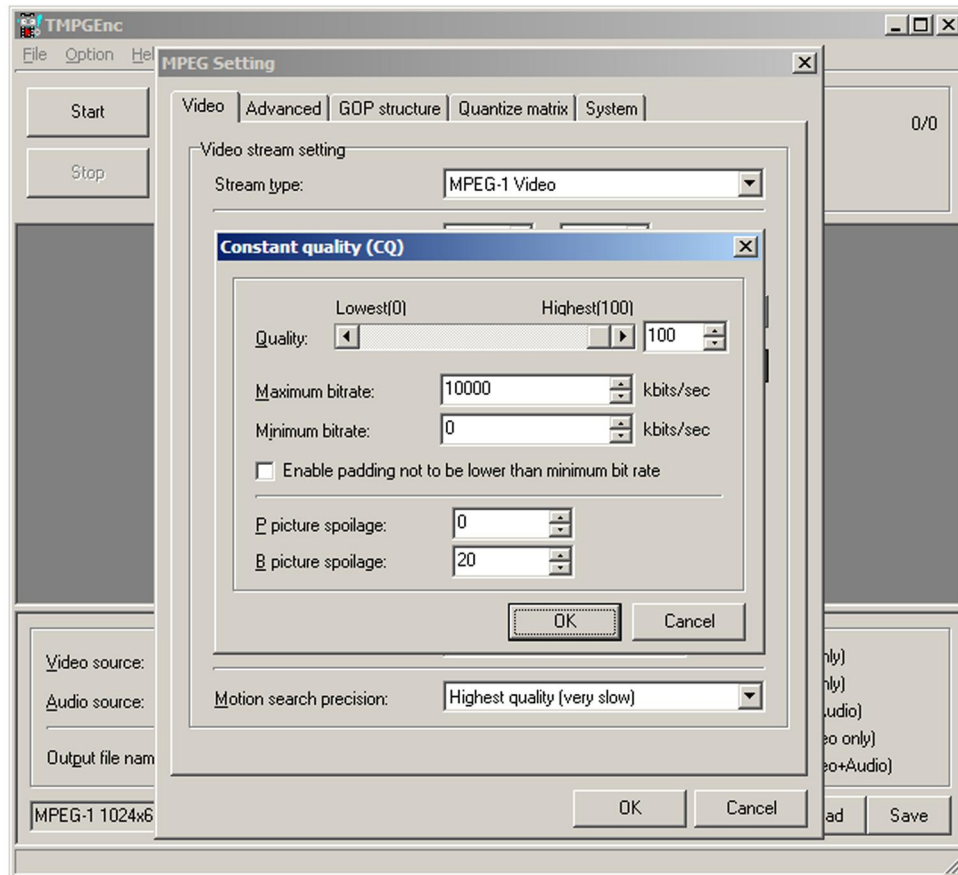
Despite the fact that this version is a bit older nowadays, it still produces much better results than many of its recent competitors. Use the **Browse** buttons (screenshot above) to locate your source movie file and to define the output file location and name, select the “System Video only”, and proceed next (clicking on the **Settings** button, above) to the “MPEG Settings” screen (below):



Here you have to specify several settings. On the first use of the encoder, change all the parameters of the “Video” tab as above. “Stream type” should be MPEG-1 (if you are using the free version, you don’t have any other option here, unless you are still in the 14-day trial period). “Size” should correspond to the size of your source image files, “Frame rate” should be the same as the one you have used when creating the uncompressed AVI file. Please note that not all frame rates are supported here, so you better check the options before assembling the uncompressed AVI file (page 7). Set the “Rate control mode” to “Constant quality (CQ)”, and “Motion search precision” to “Highest quality”. Once you set these parameters, the software should remember them until you change them again.

The last thing – but most important – which you can set here is accessible after you click the **Setting** button, to the right of the “Rate control mode”. This will take you to the screen you can see below (next page), where you can modify various parameters. To get the best result, set the “Quality” slider to “Highest (100)”, and increase the “Maximum bitrate” to some higher value, e.g. 10000. If your MPEG-1 file is too big after the conversion, you may try reducing either the “Quality” to something like 75, or the “Maximum bitrate” down to about 5000. At some point you find that reducing these parameters leads to obvious degradation of the

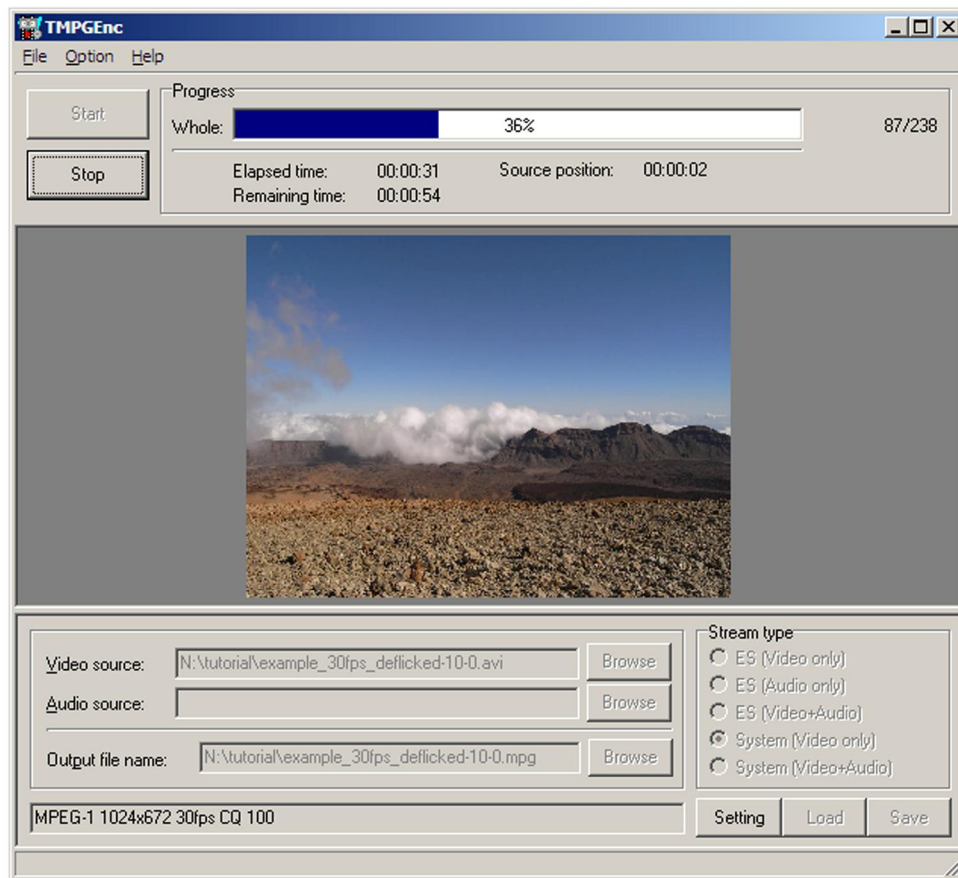
movie quality, typical “MPEG blocks” will start to appear in the movie. You have to experiment here with these settings, finding the right balance between the file size and its quality. If you intend to distribute your MPEGs on CD/DVD media only and therefore are not restricted by the movie file size, you can just keep the settings as shown below. You don’t have to change anything else here - just keep the other options at their default settings.



After setting all the options above, click twice the **OK** buttons, which will take you back to the main menu. Now click the **Start** button there (top left), which starts the file conversion. You can follow the progress as shown on the following screenshot (top of the next page).

To evaluate the MPEG file quality, I recommend using a media player which enables “pause” (most of them do nowadays), and frame-by-frame forward “stepping”. For notes on this see the last section of this guide. You will notice that some parts of your MPEG file are more vulnerable to creating MPEG blocks than the rest of the movie; so when tuning the size/quality settings try to follow these more “sensitive” movie parts.

Also, since some older computers (~ 1 GHz CPU and below) might have problems playing the files when their MPEG quality settings are set too high (e.g. as shown above), you should also check the movie files you created on such an older computer, if possible.



And that's it – you have just created your MPEG-1 movie file ...

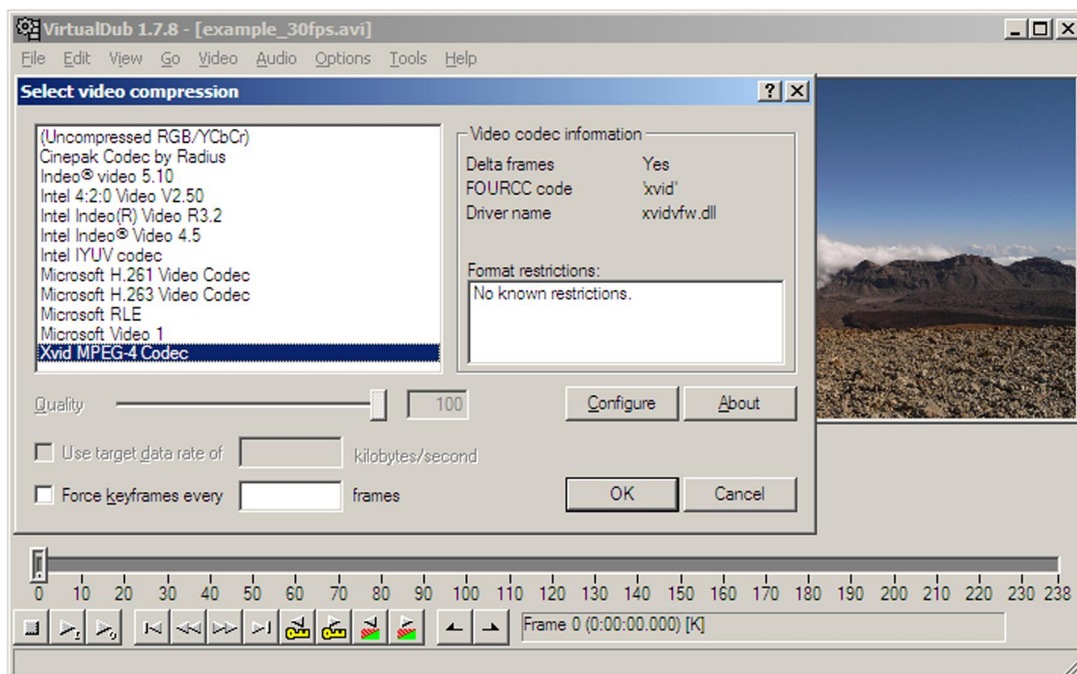
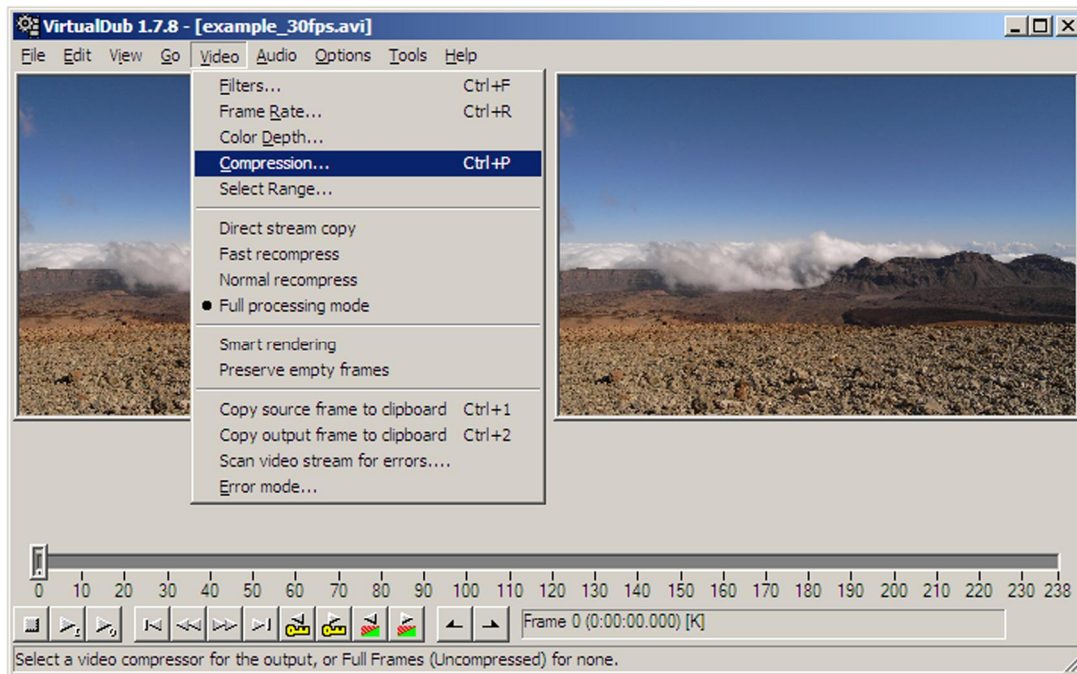
### XviD encoded AVI files

XviD codec represents a freeware alternative to the popular (but commercial) DivX codec. Both codecs should be compatible, so a movie compressed by the XviD codec should be playable on a player which supports DivX – but I have no personal experience with this, so this is without a guarantee. In any case, XviD decoders are nowadays usually included in codec bundles; however to create an XviD compressed file you will need the XviD encoder. You can get them (for all platforms: Windows XP/Vista, Linux and Mac OS) from the following two locations: <http://www.xvid.org/>, or from <http://www.xvidmovies.com/codec/>.

After downloading and installing the codec, it should become accessible from most of the video editing software. You can use it from either the RAD Video Tools or the VirtualDub, as shown next. If you don't intend to apply the deflicker filter, you can encode your file to this format right away in the RAD Video Tools, selecting the XviD compression instead of the "Uncompressed" one there. Alternatively, you can save the deflicked AVI file to the XviD encoded one from the VirtualDub, as shown next.

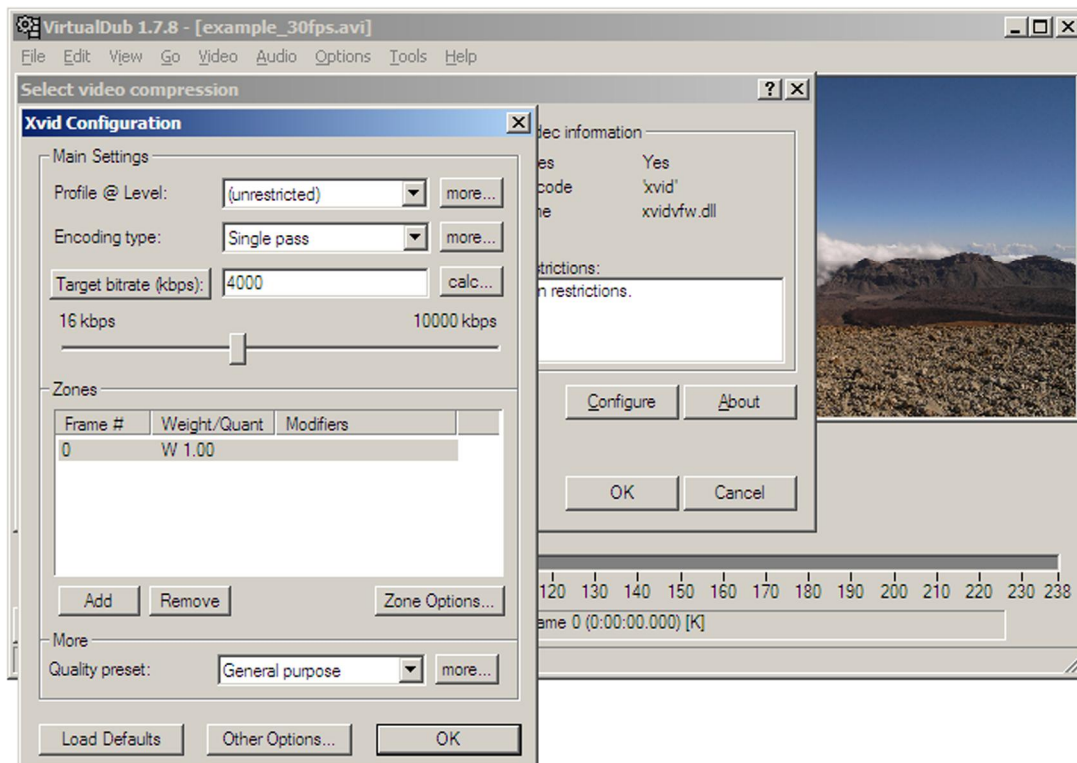


If you want to save a file using the XviD compression from the VirtualDub, open the file, go to “Video”, select “Compression...”, which opens the “Select video compression” window (the two screenshots below):



Select the “Xvid MPEG-4 Codec” here, and click the **Configure** button. This opens the “Xvid Configuration” window (next page):





Click here first the **Target quantizer:** button, it will change to **Target bitrate (kbps):** as shown above. Set the "Target bitrate" to a value somewhere between 3000 and 4000, this seems to be fine for most cases. Again: the higher the bitrate, the better the quality, but also the larger the file size. Confirm, and save the file... You can apply the XviD compression right away when deflicking the file.

## H.264 encoded QuickTime MOV files

From the options described in this guide, the H.264 codec, together with the QuickTime MOV file format provide probably the best quality at a very reasonable file size. To save a MOV file you need either a full version of QuickTime Pro (<http://www.apple.com/quicktime/>, spending about \$30 on this), or its freeware alternative, the Squared 5 MPEG Streamclip, available at <http://www.squared5.com/svideo/mpeg-streamclip-win.html>, combined with KL QuickTime Alternative ([http://www.filehippo.com/download\\_quicktime\\_alternative/?2615](http://www.filehippo.com/download_quicktime_alternative/?2615), version 1.81; install this version, not a higher one!), or with QuickTime Player, both freeware.

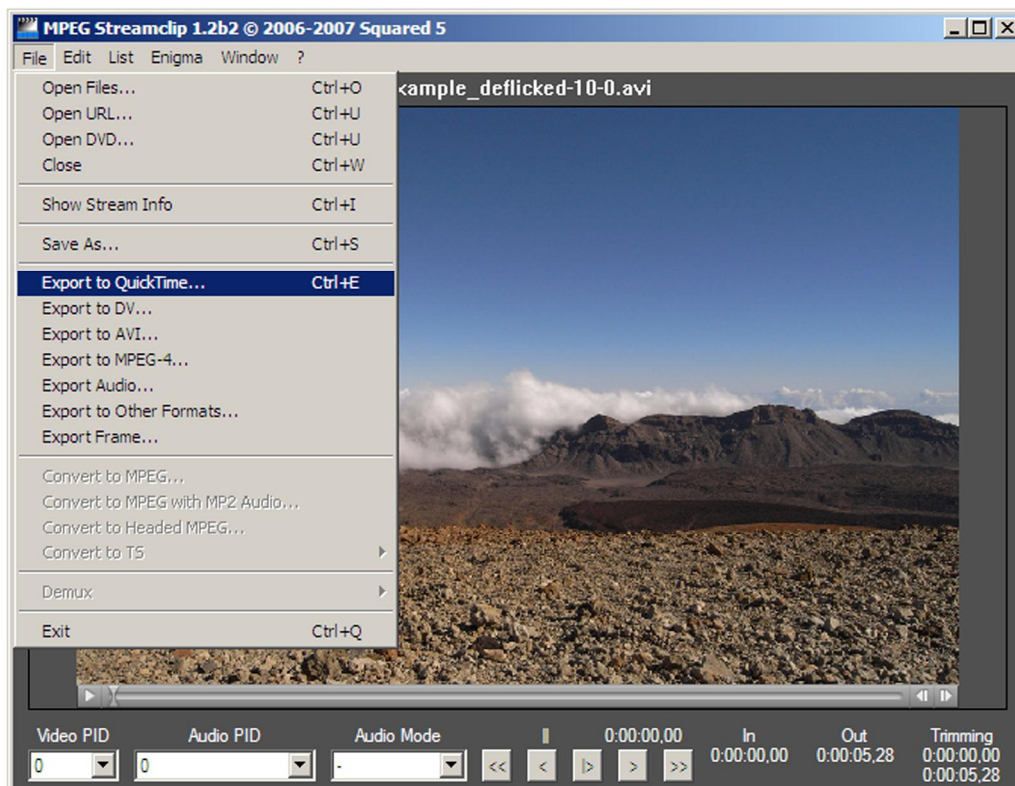
The screenshots below are from the MPEG Streamclip version 1.2b2 beta (combined with QuickTime Alternative 1.81).

MPEG Streamclip is available for Windows XP/Vista, as well as for Mac OS, and should work also in Linux with WINE.

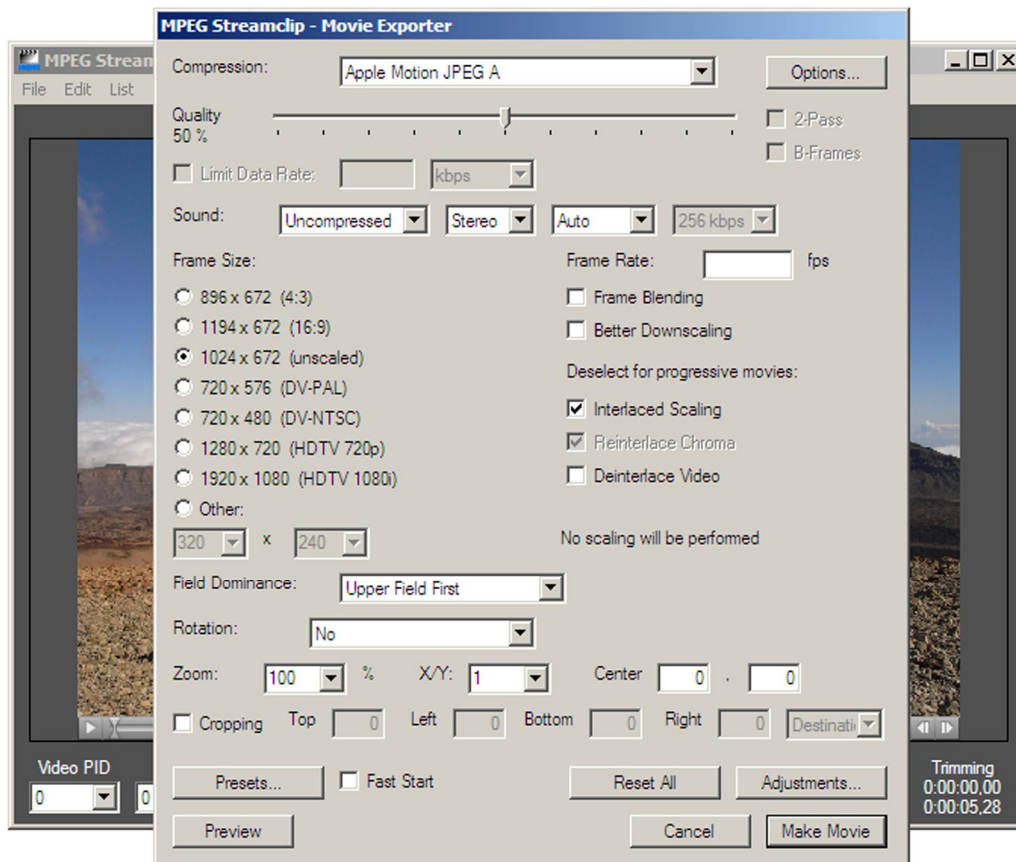
After launching the MPEG Streamclip, open your file you wish to convert to the MOV file, and you should get a screen like this:



Go to "File", select "Export to QuickTime..." (screen below):



... and the following screen (“MPEG Streamclip – Movie Exporter”) appears:



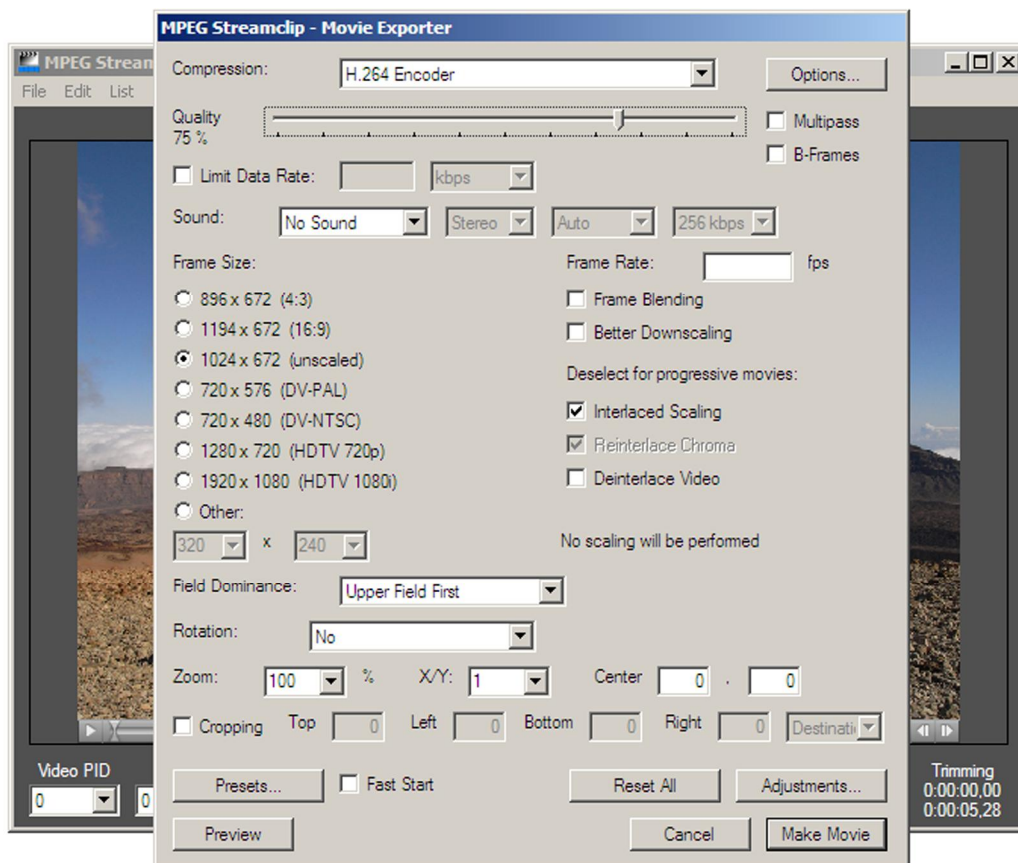
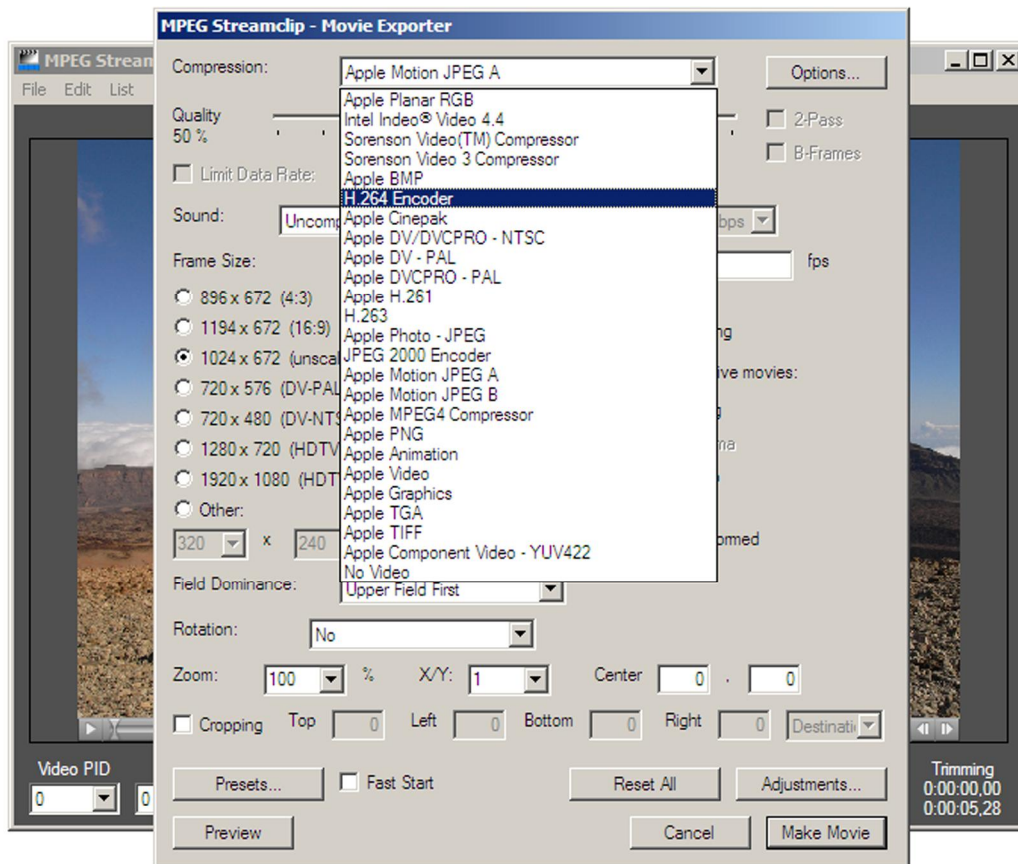
Go to “Compression” (the uppermost field there), and change the default “Apple Motion JPEG A” to “H.264 Encoder”, see the screenshot at top of the next page.

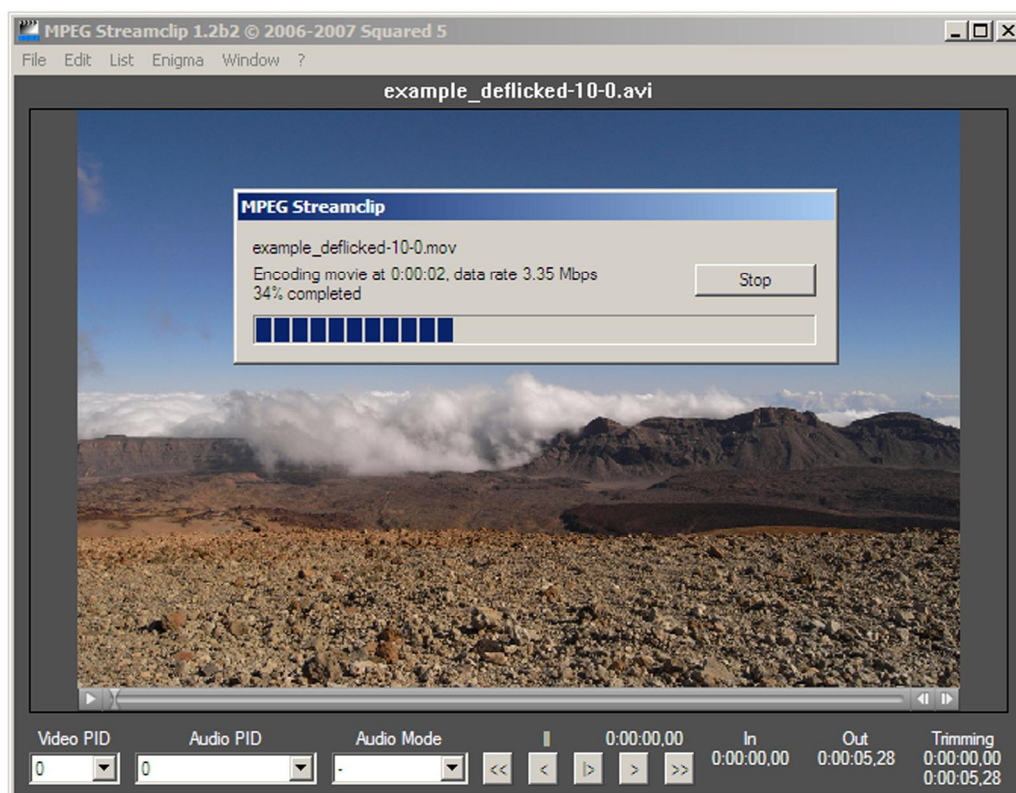
Now, in the “Movie Exporter” window (next page, bottom) you can change the “Quality”; reasonable values are somewhere between 60 and 80. Lower values result in lower quality of the movie, while higher ones lead to a larger file size. You can find the optimum value by starting at higher values, saving the file, and decreasing the quality factor next, till you start observing a degradation of the movie quality.

Also, if your file doesn’t contain any sounds, choose “No Sound”. You don’t have to modify any other settings; you better leave the “Multipass” and “B-Frames” unchecked (may result in certain incompatibility). When done, press the **Make Movie** button.

After selecting and confirming the output file name, you will be able to follow the progress of the compression of the file (last screenshot, page 25). And that’s it.







## Playing the movie files

For playing the movie files (of all types) I prefer a simple, but fully sufficient and user-friendly **Media Player Classic** ([http://www.free-codecs.com/download/Media\\_Player\\_Classic.htm](http://www.free-codecs.com/download/Media_Player_Classic.htm)), recent version 6.4.9.1. With the QuickTime Alternative (1.81, which you might have already installed because of the MPEG Streamclip) and some additional codec packs available from the same page, you will be able to play basically all movie file types. A nice feature of this viewer is that that you don't install it, you just copy it (a single file) to your system. So no boring add-ons like from most of the other freeware movie players...

When playing a movie in this player, pressing the space bar of your keyboard stops the movie playback, and you can go frame by frame forward or backward, using the cursor keys (␣, and ␣). This might be very useful, e.g. when checking the details of your movies...