

VirtualDub help - Processing: Smart rendering

What smart rendering does

Smart rendering attempts to identify which portions of a video stream must be re-encoded and which can be copied verbatim from the source file. This can save a lot of time, as typically only the regions around an edit must be re-encoded. It also preserves the quality of most of the non-edited video.

As an example, take this edit:

```
table.timeline { margin-left: 32px; } table.timeline th { background: #8ae; } table.timeline td  
{ background: #aea; padding-right: 16px } table.timeline td.cut { background: #a88; } table.timeline  
td.broken { background: #f00; }
```

K K K K

The red frames indicate frames which are meant to be kept, but cannot be preserved in their original form because their dependencies have been removed. Without the prior frames, it is not possible to decode those dangling frames before the next key frame. Ordinarily when [direct stream copy mode](#) is used, VirtualDub would add the previous two frames back into the output file to satisfy video compression restrictions.

Smart rendering re-encodes the affected frames instead. The result is that, in exchange for a little degradation in those three frames, the edit occurs in the exact desired location and the rest of the frames are left intact.

Requirements for smart rendering

To use smart rendering, the following are required:

1. The video must be compressed.
2. A video codec must be installed which can produce the same format as the source. Note that some video codecs can decode more formats than they can encode, which means you may be able to open a video file using a codec, and encode with that codec, but not encode to the same format as the source.
3. Inverse telecine must be disabled.

In addition, the video codec must be configured to match the source. There are often parameters in the video codec's configuration dialog that VirtualDub cannot programmatically access, and which may affect the encoding in incompatible ways. When in doubt, try to match the source; this includes parameters such as B-frame encoding, packed bitstream, and color space. For instance, with Huffiyuv, the prediction mode must match.

If the video codec in use has an option for multi-pass rendering, it must be disabled for smart rendering.

Filtering video while smart rendering

In order to use video filters while smart rendering is active, all video filters must be appropriately scoped using opacity curves. Any frames for which any filters have an opacity above zero will be forced to re-render.

Any video filter instance which does not have an opacity curve attached will force all frames to render, since it must process all frames. Since filters which change the frame size cannot have opacity curves attached, any size-changing in the filter list will defeat smart rendering.

Caveats of smart rendering

Not all video codecs will work with smart rendering, even if they normally work for regular encoding. If you can encode two videos with the same video codec, join them together, and play back the result,

then smart rendering is more likely to work. This is usually not a problem with video compression algorithms that do not use delta frames; Motion JPEG, for instance, should work without problems. Quality at the join point may be an issue. The video codec is not aware that VirtualDub is using it to smart-render transitions, so it may not be able to match quality very well with the frames that are copied. This is especially an issue with short runs, since the codec may require a few frames to "ramp up" in bitrate. If this occurs, consider increasing bitrate when using smart rendering.

Technical details

When smart rendering is active, VirtualDub tries to copy all the frames verbatim; whenever it detects a violation in frame dependencies, it drops to recompressing frames, and continues to do so until the next key frame, at which direct copying resumes. The recompression of the necessary segments is done by "hot-starting" the video codec for each segment; from the video codec's standpoint, each range to be recompressed is a separate video. The copied and re-encoded sections are then seamlessly joined during the output process.