

Creating video for Series 40 devices

This article gives tips and tricks for creating video for Series 40 devices.



31 Jul
2011

Overview

For developers who want to optimise or enable videos for Series 40 devices, this document is a good place to start. In addition to covering some basics, it also makes suggestions for trade-offs regarding perceived quality, file size, and compatibility. This is not a detailed guide to any particular codec, file format tool, or device, although each of these aspects will be discussed. In many cases, links are provided to internet resources that offer more detailed information about a particular topic.

Recommendation: Always trust your own eyes.

These recommendations are good choices in almost all cases. The text that follows each recommendation gives readers some background and also suggests when you should consider alternatives. If you are a novice at video creation for mobile devices, this document will help you get started. If you are already somewhat experienced, you will hopefully find useful suggestions for optimisations and compatibility considerations. Experts will probably not find much new information here, but your comments and suggestions are always welcome.

Glossary

The following definitions will help readers who have limited experience with video technology.

Codec	Decoder and encoder for a certain video or audio format.
Decoder	Decompresses video/audio so it can be played back.
Encoder	Compresses video/audio so that the number of bytes needed to store it becomes smaller.
Frame	A single image in a video sequence.
I-Frame	Intraframe; a self-containing frame that can be decoded independently of other frame(s).
Level	An upper limit for selected encoding parameters; typically, image size, bit rate, frame rate.
P-Frame	Predicted frame; a frame in a video sequence that is predicted from other frame(s).
Profile	Many codecs have profiles to allow trade-offs between complexity and amount of compression.
Transcoding	Convert a video sequence to a different image size, frame rate, bit rate, etc.

Compatibility

Recommendation: Make at least two version of your video when targeting all Series 40 models. Your first consideration when creating video for Series 40 devices is the range of devices you want to support. Since capabilities vary widely between models, you should provide more than one version of your video. By making the right choices, you should be able to keep the number of versions low (typically two or three) and still be able to provide an optimised experience across the full range of Series 40 devices.

The compatibility of your content is defined by your choice of file type, audio and video codecs, and the parameters you choose for the codecs. The following subsection covers each of these in more detail.

File types

Recommendation: Use .3gp for mobile optimised video. Use .mp4 if your tools do not support .3gp.

File format is simply a container for audio and video tracks. In Series 40, you can use exactly the same audio and video tracks in both .3gp and .mp4 containers, and the file format processing will be for most parts identical. The main reason to recommend .3gp is to make it obvious which versions of your videos are optimised for mobile use.

The secondary reasons are the additional features and constraints specified by the 3rd Generation Partnership Project (3GPP) [2]. Even if you do not currently need (or your tools do not support) all features specified for 3GPP media files, your video library will be more consistent should your needs change in the future.

.3gp,	For most purposes, these file types are interchangeable from the Series 40 playback point of view.
.mp4,	

.m4v	
.asf, .wmv	These file types are interchangeable from the Series 40 playback point of view. If you produce Windows Media video, you should choose one of these.
.avi	Audio Video Interleave (AVI) is supported in some Series 40 devices. If you already have AVI files, there may be no need to transcode them. However, it is not recommend to transcode to AVI because of possible compatibility issues with some Series 40 devices and because the random access capability to .3gp/.mp4 files in Series 40 is typically much better than to .avi files.

Video coding

Decoding complexity is a result of the following factors:

1. Codec
2. Frame rate
3. Image size
4. Bit rate
5. Bit-stream-specific details

Example: An H.264 stream encoded with the in-loop deblocking filter enabled is more complex to decode than one without.

When you create videos for low-end mobile devices you have to make compromises. If you optimise for video quality only, you will quickly end up with unnecessarily large video files that do not play at all, or that play badly in some or all of your target devices.

On the other hand, with the right choices you can create great looking video that plays beautifully in most, if not all of your target devices. The following subsections provide separate optimisation suggestions for each of the above areas.

Select the right codec

Recommendation: Use MPEG-4 SP when you cannot meet your frame rate and image size targets with H.264. If the decoding throughput becomes a bottleneck, switch to MPEG-4 SP (see Figure 1). This will allow you to use larger image sizes and/or frame rates than with H.264. Just be aware that you will need to use a higher bit rate to preserve quality.

Table 2 gives general guidance on the bit rates and image sizes for each codec available on Series 40 devices. At the lower end of the suggested bit rate range you should expect significant quality degradation, whereas the quality will be very good at the higher end.

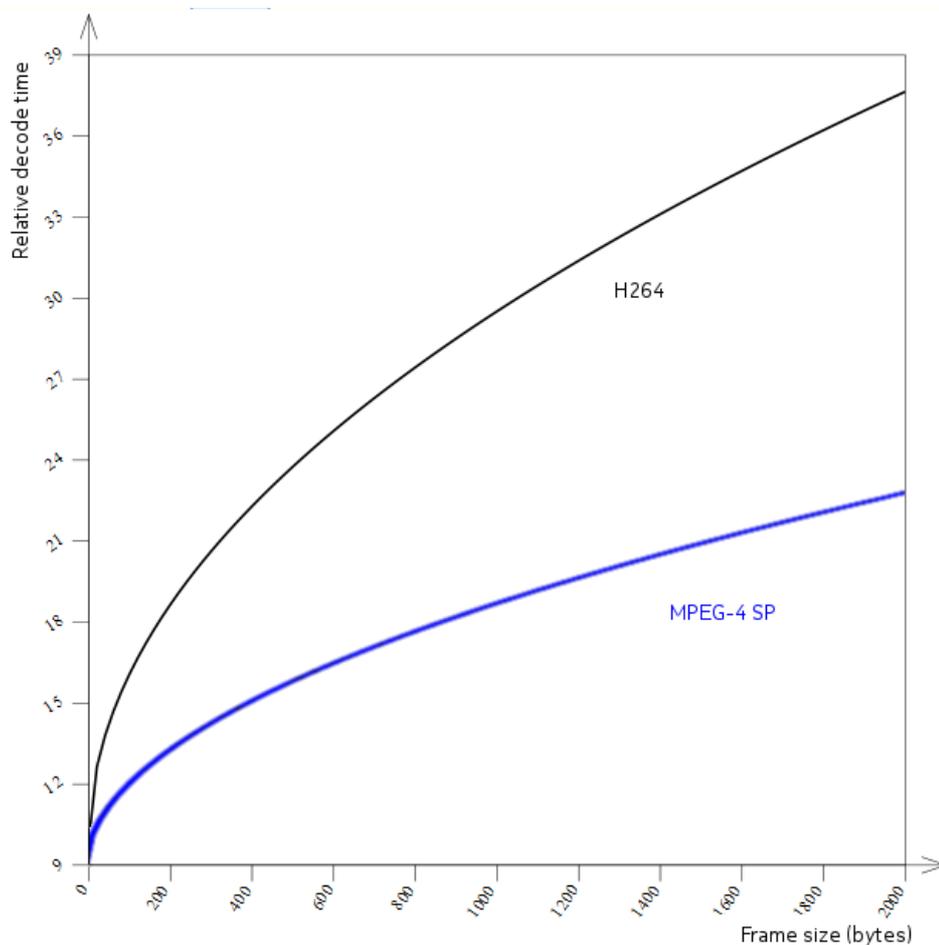


Figure 1: Relative decoding time for MPEG-4 SP and H.264 frames (QVGA) as a function of bit rate. This graph is created using a mathematical model derived from real measurements. Notice that only the time spent in video decoding is shown. Other sources of CPU load (audio decoding, video rendering, UI, etc.) are not included.

Table 2: Overview of video decoders available on Series 40 devices

Codec	Complexity	Compatibility	Bit-rate recommendation	Notes
MPEG-4 AVC/H.264, baseline profile	High	Good	40-128 kbit/s @ QCIF, 64-368 kbit/s @ QVGA	Best file size/video quality ratio. Down side is the lack of decoding power in low-end Series 40 devices.
MPEG-4 ASP (Advanced Simple Profile)	High	Medium	N/A	Only supported on a limited number of Series 40 devices. Transcoding to MPEG-4 ASP is not recommended. If you already have content in this format you can try and see whether it works in your device.
MPEG-4 SP (Simple Profile)	Medium	Good	64-128 kbit/s @ QCIF, 128-512 kbit/s @ QVGA	With this codec you will get the best image quality/decoding complexity trade-off on most Series 40 devices. Use it if you target low-end Series 40 devices or higher frame rates unless the smallest possible file size is your main target.
H263	Medium	Good	64-128 kbit/s @ QCIF, 256-512 kbit/s @ QVGA	A subset of MPEG-4 SP. Transcoding to this format is not recommended.
VC-1 / WMV9, simple and main profiles	High	Good	64-128 kbit/s @ QCIF, 256-512 kbit/s @ QVGA	WMV9 is Microsoft's implementation of the VC-1 standard. This codec is a good choice if you work with Microsoft tools.

Frame rate

Recommendation: Cut the frame rate in half if you need to reduce it. It is recommended to use the same frame rate as in the source video. If that frame rate is too high for your target device, cut it in half (even if your target devices support a slightly higher frame rate). For example, consider a 25 fps source you want to play on a device that supports video up to 15 fps. Figure 2 illustrates how maximising the frame rate can lead to distortion of the frame durations (sequences 2 and 3). This distortion can make the video look jerkier than it would at a slightly lower frame rate.

An advanced frame rate conversion algorithm can maintain correct durations for the frames by generating intermediate frames using interpolation techniques [3]. Use of such algorithms may be the right choice if you want to have the best possible quality (for example, for a 25 fps video source for devices that cannot go beyond 15 fps). A possible alternative to frame-rate conversion is to use a smaller image size so that your target device can play the video using the original frame rate. You will end up having less detail in the final video, but the motion will be smooth. Whether or not this is a good trade-off depends on the content you use.

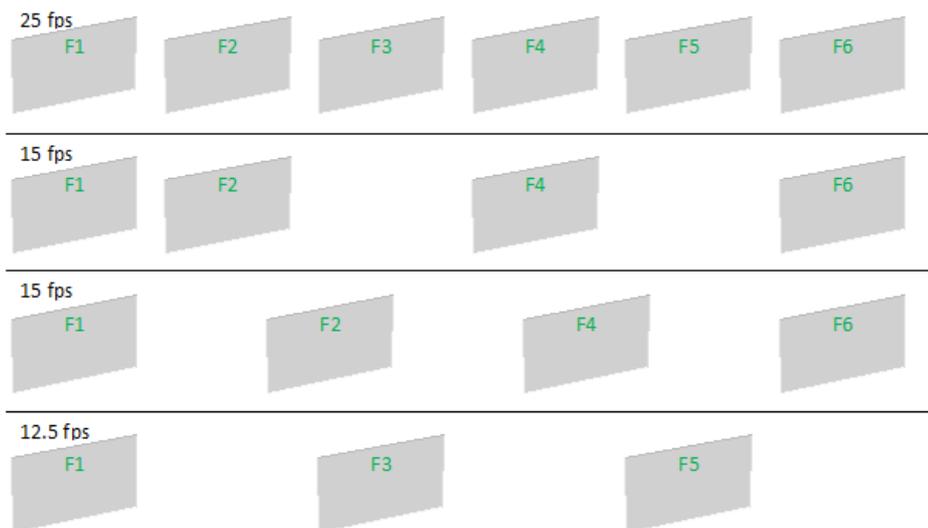


Figure 2: Illustration of frame sequences after downsampling from 25 fps source (top) to 15 fps (middle), and then to 12.5 fps (bottom). Notice how the relative frame duration is distorted for frame 1 in sequences 2 and 3.

Image size

Recommendation: Use the display resolution of your target devices as an upper limit. The choice of image size plays a key role in the viewing experience you provide to users. The following are general recommendations for image size when transcoding video for Series 40 devices:

1. Use the display size of your target device(s) as an upper limit.
2. Consider matching the aspect ratio with the display size of your target device(s):

Series 40 playback will respect the aspect ratio you have defined. Letterboxing [4] is often a poor alternative given the small display size.

1. Use multiples of 16, 8, 4, 2 for width and height (in this order). This recommendation is due to the overheads that follow when the width and the height are not divisible by 16
2. Use custom resolutions to:
 - Optimise aspect ratio mapping between your source material and device display.
 - Optimise the balance between bit rate, frame rate, and image quality. [5][4].
1. Do not upscale unless you know what you are doing and why.

Scaling algorithms

Scaling algorithms are not all equal! Area-based algorithms are good for downscaling. Bicubic and LancOS are good choices for upscaling. When in doubt, experiment with the available algorithms and trust your own eyes.

Other preprocessing considerations

With the right preprocessing you can improve the quality of the resulting video tremendously. The following subsections touch on a few areas that everyone producing video should be aware of. However, these descriptions only scratch the surface – there is much more to cover on these and other preprocessing techniques.

Deinterlace

Recommendation: Discard every second field. If you have an interlaced video source you should always use a good method to remove interlacing. This can be tricky when you target big screens. However, in the case of Series 40 devices, the vertical resolution of your interlaced source is probably at least 2x the vertical resolution of the display of your target devices. Consequently, the best approach to interlace removal is almost always to discard every other field. [6]

Remove unwanted camera movement

Recommendation: Do deshaking for videos shot by hand. If your video is shot by hand, it is highly recommend that you do deshaking [7]. It will improve your video by making it more pleasant to view, and it will improve the quality you get with a given bit rate due to the simplified motion compensation.

Remove noise

Recommendation: Invest some time experimenting with noise-reduction techniques. Noise filtering can improve quality substantially. As with deshaking, the benefits come from both more pleasant images and from the benefits for the encoding process when the bits can be used for actual content and not for the noise.

Audio coding

Recommendation: Use 48 kHz AAC-LC at 64 kbit/s. AAC audio comes in a number of different profiles [8]. Each profile can provide transparent audio quality. However, your choice of the profile will make a big difference when you search for the right balance between the quality, decoding time, and bit rate. At 64 kbit/s, AAC-LC will sound reasonably good [9] and will be relatively fast to decode. The linked study includes AAC and AAC+. AAC-LC will require a higher bit rate to achieve similar quality. If you want to have better audio quality, either increase the bit rate or switch to AAC, or both. However, before committing to any of these solutions, do a listening test to verify whether the increased quality is perceptible in your target devices in a real-use environment. If your primary goal is to minimise file size, use eAAC+ (or HE AAC v2, as it is also called). With this codec you can go down to 24 kbit/s and still get reasonable quality. Since eAAC+ is slow to decode, use it only with low bit rates and to reduce sampling frequency of 24 kHz.

AMR-NB	Lowest	Good	Okay for speech	12.2 kbit/s	Use with narrowband sources when you need to have really small files or your source contains only speech.
AMR-WB	Low	Medium	Good	16 kbit/s	Do not transcode to this format. Reduced compatibility trumps benefits.
AAC	Medium	Good	Scalable	32-128 kbit/s	Use if your tool does not support AAC-LC.
AAC-LC	Low	Good	Scalable	32-128 kbit/s	Recommended choice.
eAAC+ / HE AAC v2	High	Good (see notes)	Scalable	24-32 kbit/s	Good quality audio even at low bit rates.
WMA	Medium	Good	Scalable	32-128 kbit/s	Should be your default choice when you create Windows media files.
WMA 10 Pro	High	Medium	Scalable	24-32 kbit/s	Computationally complex. Only use with very low bit rates.

Sample rate

48 kHz is the frequency used by the digital analog converter (DAC) in Series 40 devices. If you use another sample rate it needs to be upsampled to this frequency. In most cases, it is better to do this conversion once when you create the video and not every time you play it back in a mobile device. There are two exceptions to this general rule:

1. If your source material is narrowband audio (like AMR–NB) you should keep it that way. Converting to a wideband format will only increase the file size and decoding time.
2. If your target devices do not have enough processing power you should use 24 kHz for audio. This will practically halve the decoding time (compared to 48 kHz).

References:

1. <http://www.3gpp.org/ftp/specs/html-info/26234.htm>
2. http://compression.ru/video/frame_rate_conversion/index_en_frcn.html
3. <http://en.wikipedia.org/wiki/Letterbox>
4. <http://en.wikipedia.org/wiki/Macroblock>
5. <http://www.100fps.com/>
6. http://wn.com/Deshake_test
7. http://en.wikipedia.org/wiki/High-Efficiency_Advanced_Audio_Coding
8. <http://tech.ebu.ch/docs/tech/tech3296.pdf>