White Paper

High-Definition Video Deinterlacing Using FPGAs

This white paper explains different deinterlacing techniques and shows how they can be implemented using Altera's Video and Image Processing Suite of IP. This video design methodology lets designers explore hardware trade-offs when implementing different deinterlacing algorithms.

Introduction

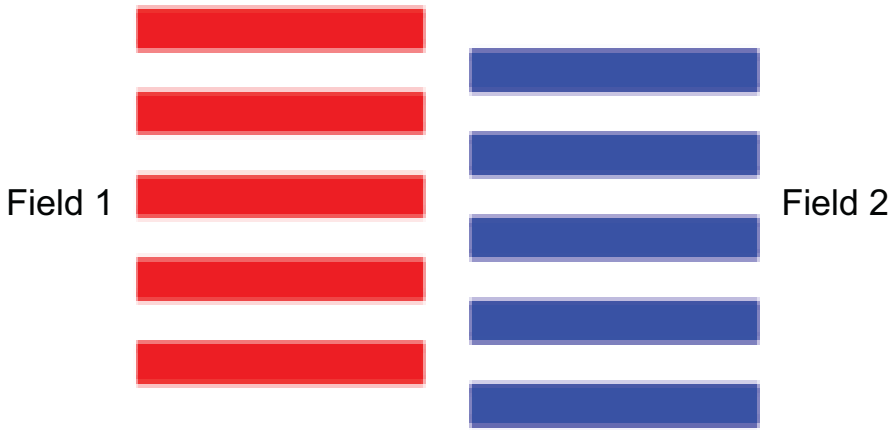
Deinterlacing was developed to address a legacy problem: the interlaced video that was required by old analog televisions must be converted to be shown on today's digital televisions. An interlaced video is a succession of 50/60 fields per second, where each field carries only half of the rows that are displayed in each frame of video. In some ways, interlaced video was an elementary compression technique when older display technologies were based on cathode ray tubes (CRTs).

Today, deinterlacing is an important video processing function and is required in many systems. Much of video content is available in the interlaced format and almost all of the newer displays—whether LCD or plasma—require progressive video input. However, deinterlacing is by nature complex and no algorithm produces a perfect progressive image.

Background

In interlaced video, one frame of video is broken into two fields, one of which contains the even lines and the other the odd lines. However, to display any interlaced video on a LCD or plasma display, the display must be deinterlaced. All newer displays are progressive in that each frame is comprised of a set of pixels (i.e., 1920 × 1080). Figure 1 shows how these two fields contain the pixels in one frame. Also, note that each field records pixel values that are separated in time.

Figure 1. Two Interlaced Fields Contain One Frame of Video



Assuming there are 30 frames per second (fps) or 60 fields per second, then field 0 is at time "t" and field 1 is at time "t + 1/60." Since the fields are recorded at slightly different time intervals, the two fields cannot be combined to create a progressive frame for any video that has motion. The complexity associated with deinterlacing is due to the need to estimate and compensate for the potential motion in that one-sixtieth of a second.

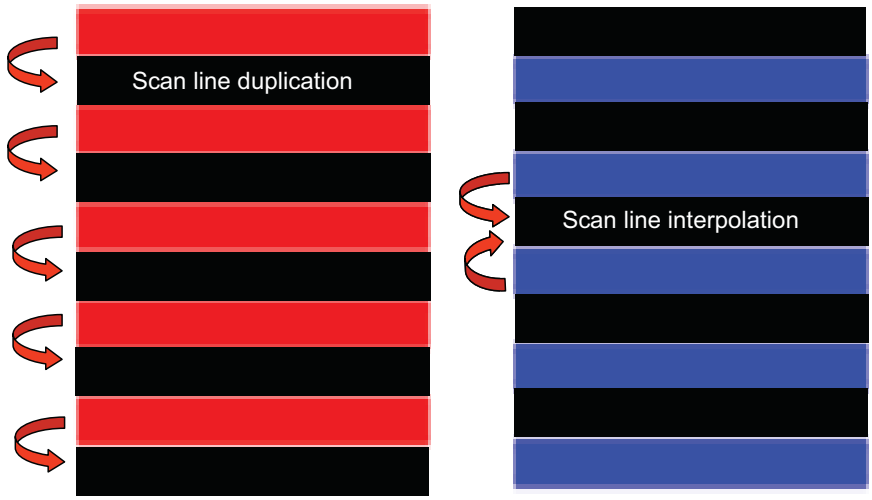
Basic Deinterlacing Techniques

Fundamentally, deinterlacing is the process of taking a stream of interlaced frames and converting it to a stream of progressive frames. The two basic deinterlacing methods are commonly referred to as "bob" and "weave."

In the bob deinterlacing technique, each field becomes its own frame of video, so an interlaced NTSC clip at 29.97-fps stream becomes a 59.94-fps progressive. Since each field has only half of the scan lines of a full frame, interpolation must be used to form the missing scan lines.

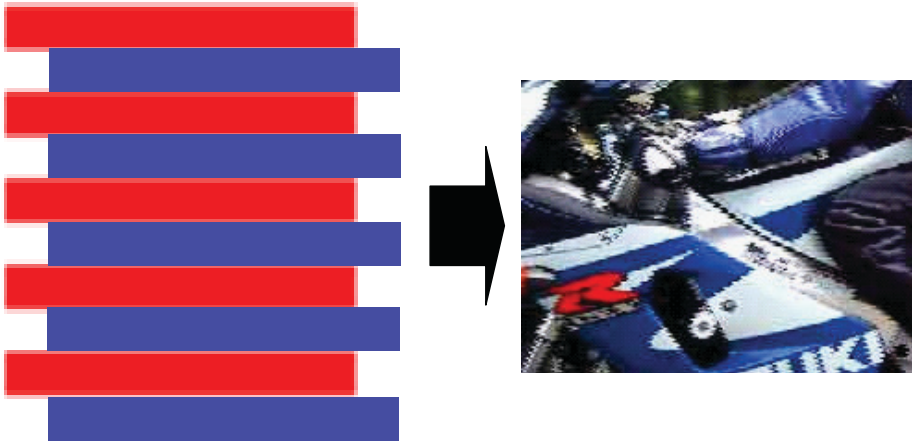
Bob deinterlacing can also be described as spatial line doubling, in which the lines in each field are doubled. The new line generated can either be just a copy of the previous line (scan-line duplication) or computed as an average of the lines above and below (scan-line interpolation), as shown in Figure 2. Bob deinterlacing provides a good result when the image intensity varies smoothly, but it can soften the image because it also reduces the vertical resolution.

Figure 2. Two Methods for Bob Deinterlacing



The weave technique for deinterlacing involves weaving two fields, which are separated in time, into one full frame, as illustrated inFigure 3. This technique provides good results if there is no motion in the one-sixtieth of a second that separates the two fields (for NTSC video). Sometimes, when the pairs of interlaced fields have been created from original progressive frames, the results of the weave algorithm are perfect. If there is motion, however, artifacts such as “mouse teeth” appear.

Figure 3. Using Weave Deinterlacing to Combine Two Fields Can Cause Mouse Teeth Artifacts



Both bob and weave deinterlacing can affect the image quality, especially when there is motion. The bob method can soften the image and the weave method can create jagged images or mouse teeth artifacts. Figure 4 contrasts an image generated with the bob deinterlacing technique with an image generated with the weave deinterlacing technique.

Figure 4. Differences Between Bob (left) and Weave (right) Deinterlacing

Advanced Deinterlacing Techniques

An obvious way to get better quality deinterlacing would be to mix up both the techniques described in the preceding section, first computing whether there is motion between successive frames of video. This technique, which advocates the weave technique for static regions and the bob technique for regions that exhibit motion, is referred to as "motion-adaptive deinterlacing."

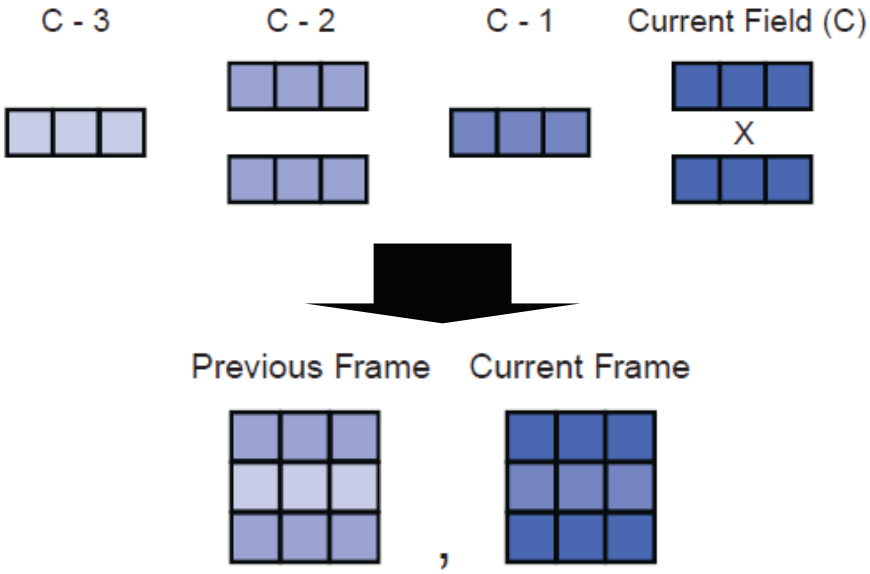
The key to motion-adaptive deinterlacing is obtaining accurate motion detection, usually by comparing an $m \times n$ matrix of pixels from one frame to the next. This simple motion-adaptive deinterlacing algorithm is implemented by the deinterlacer provided with Altera's Video and Image Processing (VIP) Suite of intellectual property (IP):

1. Collect the pixels from the current field and the three pixels preceding it, as shown in Figure 5. These pixels create two frames, a current and a previous as shown.

2. Assemble these pixels into two groups of 3 × 3 pixels, one for the current frame and one for the previous frame.

3. Calculate the difference in the pixel values (also called motion values) between the two frames.

Figure 5. Collecting Pixels for Calculating Motion



The motion value calculated can be used as is or compared to the previous motion value generated. If the previous motion value is higher, then the current motion value is adjusted so that it is between the calculated amount and the previous amount. This additional computation is also called “motion bleed,” because the motion values from more than one frame in the past are carried over. It is an exponential decay; after a motion, it may take 3 to 10 frames before the weave is again stabilized.

In either case, based on the motion value, either the weave algorithm is selected or a new pixel is calculated by spatial interpolation of the upper and lower pixels. A simple equation computes a weighted mean of the new pixel generated by interpolation or the weave algorithm (the output pixel in the previous field):

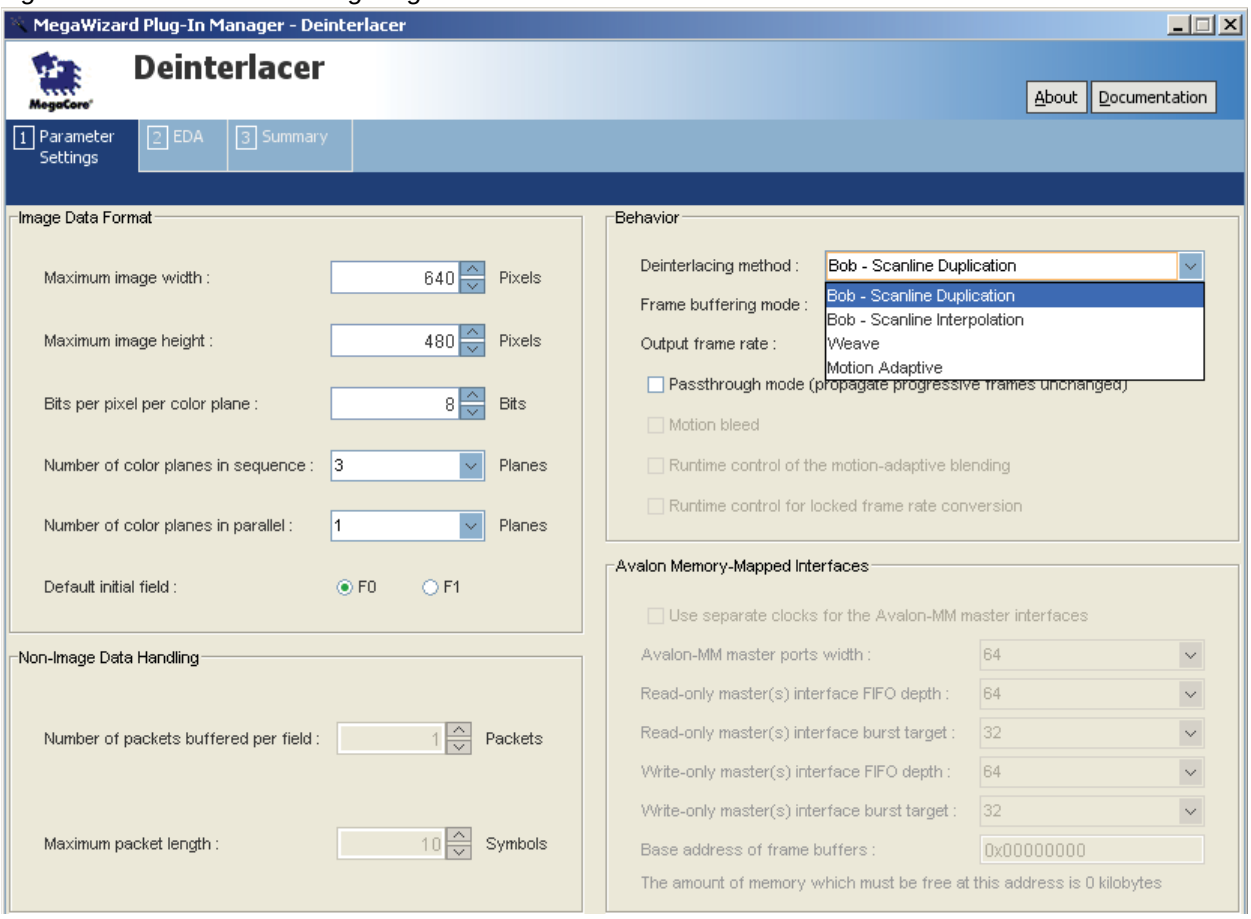
$$\text{Output Pixel} = M \cdot \frac{\text{Upper Pixel} + \text{Lower Pixel}}{2} + (1 - M) \cdot \text{Still Pixel}$$

Altera’s Deinterlacer

Altera’s VIP Suite provides a library of video and image processing IP cores designed for easy plug-and-play type interface. This suite includes IP that ranges in complexity from a color space converter to a polyphase scaler and motion-adaptive deinterlacer. As shown in Figure 6, the Deinterlacer MegaCore® function provided in the VIP Suite supports four deinterlacing methods:

- Bob with scan-line duplication
- Bob with scan-line interpolation
- Weave
- Motion adaptive

Figure 6. Altera’s Deinterlacing MegaCore Function



The bob deinterlacing method implemented by the Deinterlacer MegaCore function with the scan-line duplication algorithm is the simplest and cheapest in terms of logic. Output frames are produced by simply repeating every line in the current field twice. If the output frame rate used is the same as the input frame rate, then half of the input fields are discarded because only the current field is used.

The bob deinterlacing method implemented by the Deinterlacer MegaCore function with scan-line interpolation algorithm has a slightly higher logic cost than the bob deinterlacing method with scan-line duplication, but offers significantly better quality. Output frames are produced by filling in the missing lines from the current field with the linear interpolation of the lines above and below. At the top of an F1 field or the bottom of an F0 field, where only one line is available, that line is just duplicated. Again, if the output frame rate used is the same as the input frame rate, then half of the input fields are discarded because only the current field is used.

The weave deinterlacing method creates an output frame by filling all of the missing lines in the current field with lines from the previous field. This option gives good results for the still parts of an image but unpleasant artifacts in the moving parts. However, the weave algorithm requires external memory, so either double or triple buffering must be selected. This makes it significantly more expensive in logic elements and external RAM bandwidth than either of the bob algorithms. The results of the weave algorithm can sometimes be perfect, such as in instances where pairs of interlaced fields have been created from the original progressive frames. The weave deinterlacing method simply stitches the frames back together and the results are the same as the original.

The Deinterlacer MegaCore function provides a simple motion-adaptive algorithm. This is the most sophisticated of the algorithms provided but also the most expensive, both in terms of logic area and external memory bandwidth requirement. This algorithm avoids the weaknesses of bob and weave algorithms by using a form of bob deinterlacing for moving areas of the image and weave style deinterlacing for still areas. This algorithm is explained in the previous section

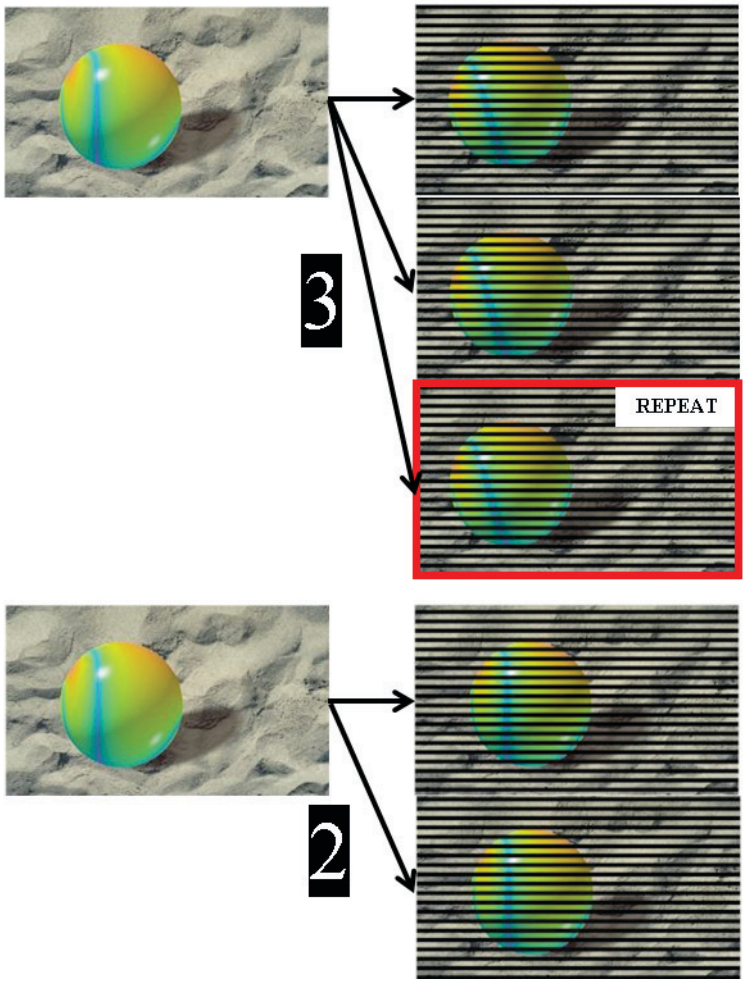
In addition, the Deinterlacer MegaCore function provides double or triple buffering in external RAM. Buffering is required by the motion-adaptive and weave deinterlacing methods and is optional for the bob method. The deinterlacer can be configured to produce one output frame for each input field or to produce one output frame for each input frame (a pair of two fields).

Cadence Detection

Interlaced video can be even more complex than just the transmission of odd and even fields. Motion picture photography is progressive and is based on 24 fps, whereas the NTSC format is 60 fields per second. This means that the conversion of motion picture photography into interlaced video creates even more complex cadences.

To convert motion picture photography to interlaced video, each progressive frame is converted into two fields, so 24 fps converts to 48 fields per second. To increase the 48 fields to the required 60, a 3:2 pull-down technique—or “cadence”—is used to generate three fields from one film frame and two fields from the other film frame, as shown in Figure 7.

Figure 7. 3:2 Cadence Generates 5 Fields from 2 Frames or 60 Fields from 24 Frames of Video



In addition, sometimes every twelfth field is dropped to accelerate the film and fit it within a given time slot. This loss is barely noticed by an average viewer, but results in a 3:2:3:2:2 cadence:

Frame 1: 3 fields
Frame 2: 2 fields
Frame 3: 3 fields
Frame 4: 2 fields
Frame 5: 2 fields
Repeat

Although 24 fps film and its associated 3:2 video cadence is the most common format, professional camcorders and various types of video processing use different types of cadences. This is a challenge for deinterlacers since a comparison must be made between the incoming fields and detecting the cadence. Most deinterlacers can detect commonly used cadences such as 3:2 and implement the right deinterlacing technique. However, if the esoteric cadences are not detected, video data may be discarded unnecessarily.

It is also conceivable that one part of the frame might have 3:2 cadence, while the other part may be straight interlaced (e.g., a film is inserted in an interlaced video). To detect and correctly deinterlace such a source would require deinterlacers to implement a per-pixel cadence detection. All this adds to the complexity of the system, for both logic and external memory accesses.

Motion-Compensated Deinterlacing

Motion-compensated deinterlacing is the most advanced deinterlacing technique by far. While this technique is not described in detail here, it uses the motion-compensation techniques generally used for video compression. The technique looks at more than one field to determine motion for a block of pixels, then shifts the pixels to compensate for the motion. This type of deinterlacing is the most computationally demanding technique, but has the best output quality.

Hardware Considerations

Ideally, deinterlacers are implemented in hardware, and FPGAs are used to implement sophisticated high-definition (HD) deinterlacers. Memory is the most important hardware resource required to build a highly efficient deinterlacer. This applies to both on-chip memory to store the $m \times n$ block of pixels across the different fields (the calculated and previous motion value matrices), as well as the external (generally DDR) memory to store multiple-input video fields and the calculated frames.

Table 1 shows the resources used to implement a motion-adaptive deinterlacing algorithm on a PAL video source in Altera® Cyclone® III and Stratix® III FPGAs. The table also contrasts the resources used for a motion-adaptive deinterlacing technique with the resources used for a simple weave technique. Notice the drop-off in the amount of memory used even when the weave technique is applied to a higher resolution image.

Table 1. Silicon Resources Required to Implement Deinterlacing in an FPGA Fabric

Device Family	Combinational LUTs/ALUTs	Logic Registers	Memory			DSP Blocks		f_{MAX} (MHz)
			ALUTs	Bits	M9K	(9x9)	(18x18)	
Deinterlacing PAL (720x576) with 8-bit Y'CbCr 4:4:4 color using the motion-adaptive algorithm.								
Cyclone III	5723	5678	—	81514	39	1	—	121
Stratix III	4803	5772	5	73292	41	—	1	243
Deinterlacing HDTV 1080i resolution with 23-bit Y'CbCr 4:4:4 color using the weave algorithm.								
Cyclone III	3231	2546	—	3078	15	—	—	170
Stratix III	3539	2540	—	3078	19	—	—	280

Additionally a deinterlacer requires frequent external memory access, thus an important consideration in the design of a deinterlacer is the expected memory bandwidth. To buffer one field of a 480i video source requires 165.7 Mbps:

$$(720 \times 240 \text{ pixels/field}) \times (16 \text{ bits/pixel}) \times (59.94 \text{ fields/sec}) = 165.7 \text{ Mbps}$$

The bandwidth doubles for a progressive frame and increases even more for HD video. To calculate the bandwidth, calculate the number of frame accesses that a deinterlacer has to do and then add the total bandwidth. Compare this to the expected bandwidth of the DDR memory interface, which depends on the throughput and the width of the memory interface.

Conclusion

For many infrastructure, military, and industrial systems, FPGAs are the ideal platform for implementing high-quality video deinterlacing. While the complexity of the deinterlacing algorithm implemented is driven by system needs, sophisticated deinterlacing solutions available from Altera's Cyclone III and Stratix III FPGAs and VIP Suite of IP can assist in starting designs.

Before starting a video system design, check out the largest, most comprehensive collection of video solutions in the FPGA industry—the Altera VIP Suite. This combination of IP cores, interface standards, and system-level design tools was developed to enable a plug-and-play video system design flow. Altera's comprehensive suite of video-function IP blocks can be connected together to design and build video systems. In addition, designs built using the VIP Suite are open, allowing a designer to easily replace Altera's IP with a custom function block.

Further Information

- Altera Video Solutions:
  www.altera.com/technology/dsp/video/dsp-video-solutions.html
- Altera's Video and Image Processing Suite MegaCore Functions:
  www.altera.com/products/ip/dsp/image_video_processing/m-alt-vipsuite.html
- *Video and Image Processing Suite User Guide*:
  www.altera.com/literature/ug/ug_vip.pdf

Acknowledgements

- Suhel Dhanani, Sr. Manager DSP Marketing, Altera Corporation

101 Innovation Drive
San Jose, CA 95134
www.altera.com