


Smart Processor Picks for Consumer Media Applications

Optimized DSP Software • Independent DSP Analysis




Introduction to Video Compression

(Workshop 210)

Berkeley Design Technology, Inc.

info@BDTI.com
<http://www.BDTI.com>

© 2004 Berkeley Design Technology, Inc.



Outline

- **Motivation and scope**
- Still-image compression techniques
- Motion estimation and compensation
- Reducing artifacts
- Color conversion
- Conclusions

© 2004 Berkeley Design Technology, Inc. 2

© 2004 Berkeley Design Technology, Inc.



Motivation and Scope

- Consumer video products increasingly rely on video compression
 - DVDs, digital TV, personal video recorders, Internet video, multimedia jukeboxes, video-capable cell phones and PDAs, camcorders...
- Video product developers need to understand the operation of video "codecs"
 - To select codecs, processors, software modules
 - To optimize software
- This presentation covers:
 - Operation of video codecs and post-processing
 - Computational and memory demands of key codec and post-processing components



Outline

- Motivation and scope
- **Still-image compression techniques**
- Motion estimation and compensation
- Reducing artifacts
- Color conversion
- Conclusions



Still-Image Compression

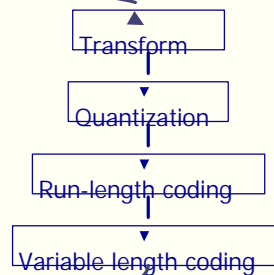
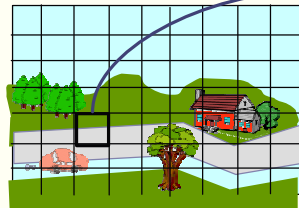
- Still-image compression
 - Still-image techniques provide a basis for video compression
 - Video can be compressed using still-image compression individually on each frame
 - E.g., "Motion JPEG" or MJPEG
- But modern video codecs go well beyond this
 - Start with still-image compression techniques
 - Add motion estimation/compensation
 - Takes advantage of similarities between frames in a video sequence

© 2004 Berkeley Design Technology, Inc.

5



Still-Image Compression



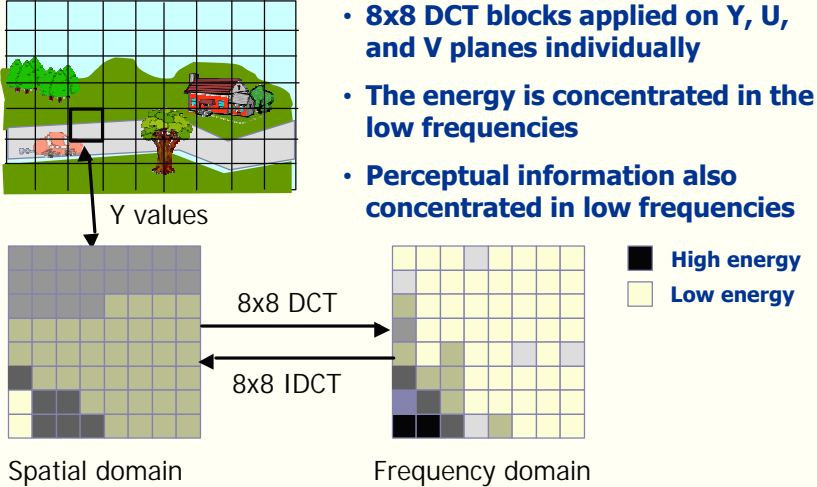
Typical Still-Image Compression Data Flow

© 2004 Berkeley Design Technology, Inc.

6

BDTi

Block Transform: 8x8 DCT



- 8x8 DCT blocks applied on Y, U, and V planes individually
- The energy is concentrated in the low frequencies
- Perceptual information also concentrated in low frequencies

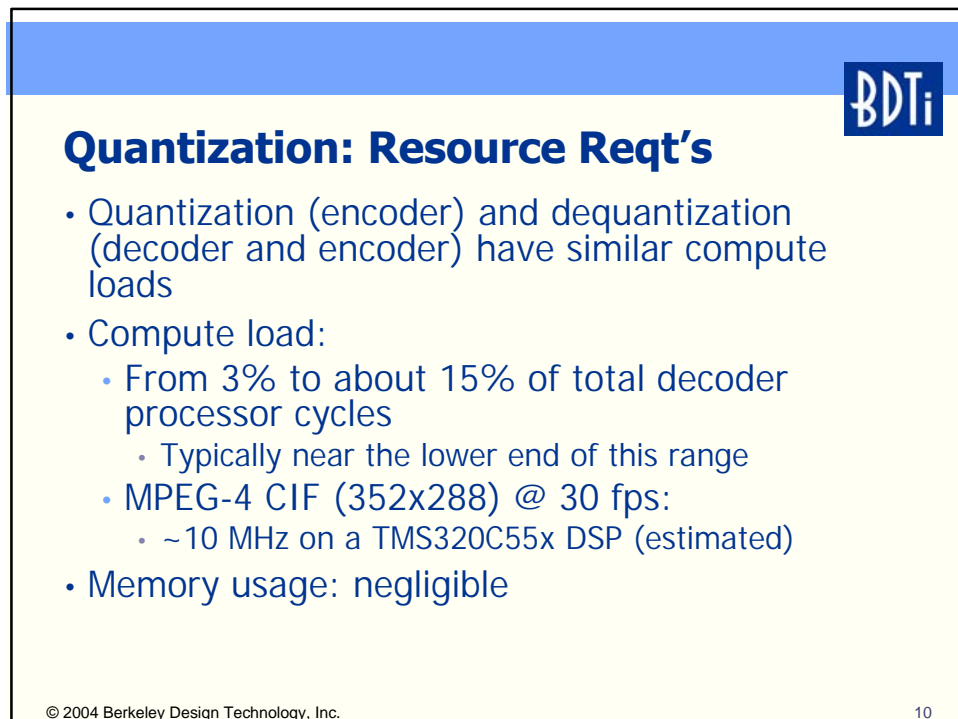
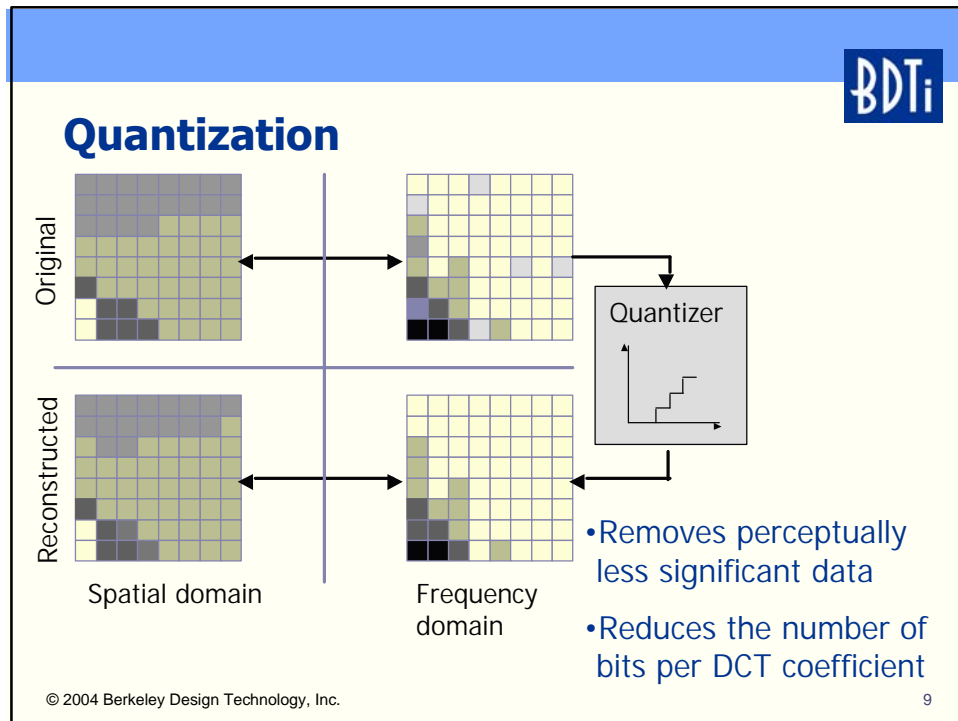
© 2004 Berkeley Design Technology, Inc. 7

BDTi

Block Transform: Resource Req't's

- Compute load:
 - Up to 30% of total video decoder processor cycles
 - MPEG-4 CIF (352x288) @ 30 fps:
 - 71,280 DCTs/s
 - ~40 MHz on a TMS320C55x DSP
 - ~10 MHz if using TMS320C55x DCT accelerator
 - Many implementation and optimization options
 - Can make compute requirements hard to predict
- Memory usage: negligible

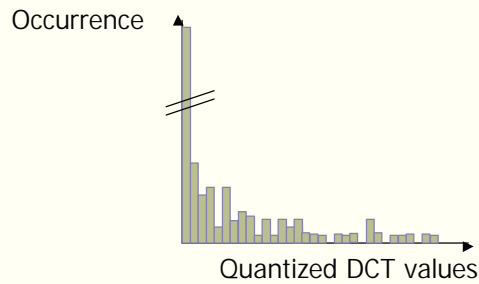
© 2004 Berkeley Design Technology, Inc. 8





Coding Quantized DCT Coefficients

Goal: Reduce the number of bits required to transmit the quantized coefficients



Observation: Unequal distribution of quantized DCT coefficient values



Variable Length Coding (VLC/VLD)

- Allocates fewer bits to the most frequent symbols (e.g., using Huffman)
- Integer number of bits per symbol
 - Not the most efficient coding method
 - Arithmetic coding more efficient, but expensive
 - Run-length coding improves efficiency of VLC/VLD for image and video coding

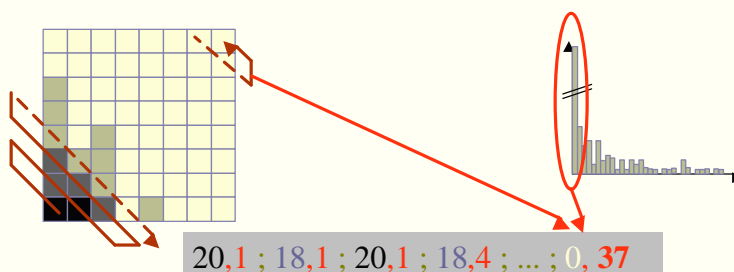
<u>Symbol</u>	<u>Frequency</u>	<u>Code</u>
A	22	1
B	16	011
C	9	0101
D	7	0100
E	4	0011
F	2	0010
...



Run-Length Coding

Encodes value and number of successive occurrences

Takes advantage of the high number of recurring zeros



© 2004 Berkeley Design Technology, Inc.

13



Variable Length Coding: Processing Reqt's

- VL decoding much more computationally demanding than VL encoding
- VLD compute load:
 - Up to 25% of total video decoder processor cycles
 - MPEG-4 CIF (352x288) @ 30 fps, 700 kbps:
 - ~15-25 MHz on a TMS320C55x DSP (estimated)
 - About 11 operations per bit on average
- Memory usage
 - A few KB of memory for lookup tables
 - More for speed optimizations

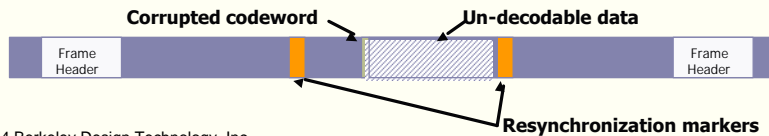
© 2004 Berkeley Design Technology, Inc.

14



Resynchronization Markers

- Without markers, a single bit error in the coded bitstream prevents decoding of the rest of the frame
 - Size of a corrupted variable-length code word is unknown
 - Therefore, the start of the next code word (and all following code words) is unknown
- Resynchronization markers help the decoder recover from bitstream errors
 - Provide a known bit pattern interspersed throughout the bitstream
 - In case of an error, decoder searches for next marker, then continues decoding



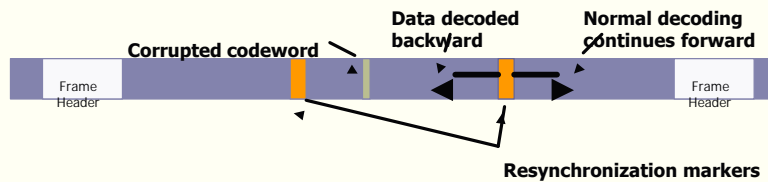
© 2004 Berkeley Design Technology, Inc.

15



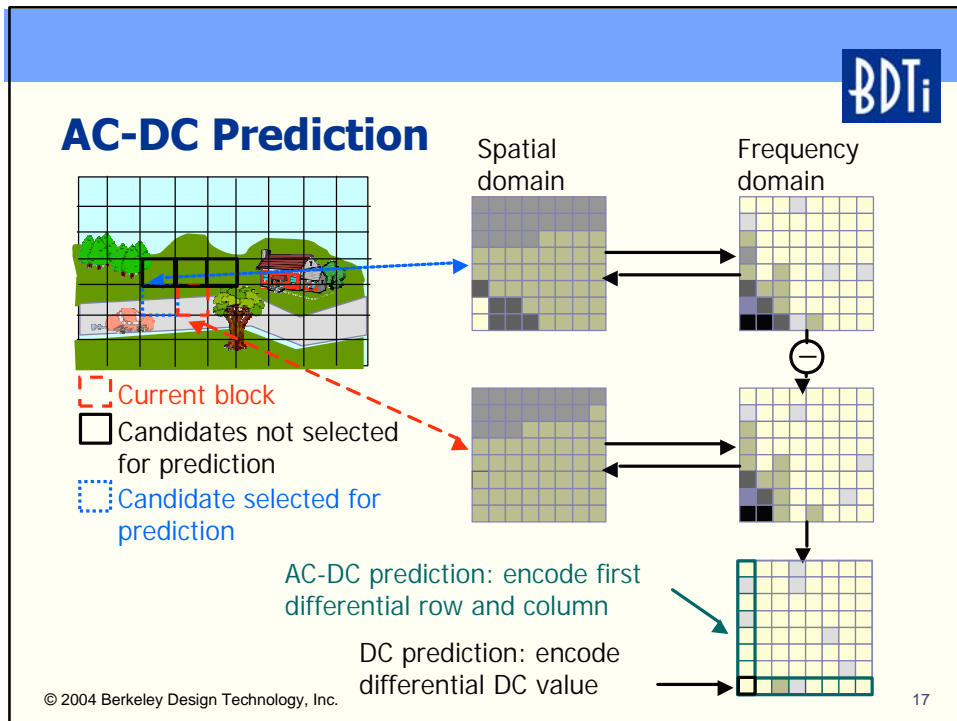
Reversible VLC

- Reversible VLC, in conjunction with resynchronization markers, further assists error recovery
 - Code words can be decoded forward and backward
 - In case of error, data can be decoded backward from the next resynchronization marker
 - More data is recovered compared to resynchronization markers alone



© 2004 Berkeley Design Technology, Inc.

16



-
- The slide lists the characteristics of AC-DC prediction. It states that AC-DC prediction cannot be used in conjunction with motion compensation, is used mostly for compressing a single image, and that DC prediction is used in JPEG. It also notes that AC-DC prediction often uses simple filters to predict each coefficient value from one or more adjacent blocks. The BDTi logo is in the top right corner.
- AC-DC Prediction**
- AC-DC prediction cannot be used in conjunction with motion compensation
 - ⇒ Used mostly for compressing a single image
 - DC prediction used in JPEG
 - AC-DC prediction often uses simple filters to predict each coefficient value from one or more adjacent blocks
- © 2004 Berkeley Design Technology, Inc. 18



AC-DC Prediction: Processing Reqt's

- Compute load:
 - DC prediction has negligible load
 - AC-DC prediction used in about 8% of frames in typical video
 - Negligible average load (~2% of processor cycles in decoder)
 - Substantial per-frame load (~20-30% of cycles required to decode a frame that uses AC-DC prediction)
- Memory usage:
 - Under 2 KB for CIF (352x288) resolution
 - But more memory (up to 10 KB) can result in faster code
 - May be overlapped with other memory structures not in use during prediction

© 2004 Berkeley Design Technology, Inc.

19



Outline

- Motivation and scope
- Still-image compression techniques
- **Motion estimation and compensation**
- Reducing artifacts
- Color conversion
- Conclusions

© 2004 Berkeley Design Technology, Inc.

20



Motion Estimation and Compensation

- Still image compression ignores the correlation between frames of video
 - JPEG achieves ~10:1 compression ratio
 - Wavelet transform-based image coding reaches compression ratios up to ~30:1
- Adding motion estimation and compensation results in much higher compression ratios
 - Good video quality at compression ratios as high as ~200:1



Motion Estimation and Compensation

- Requires at least one "reference frame"
 - Reference frame must be encoded before the current frame
 - But, reference frame can be a future frame in the display sequence
 - Three kinds of frames: I, P, and B
- I frame is encoded as a still image and doesn't depend on any reference frame




- P frame depends on previously displayed reference frame



- B frame depends on previous and future reference frames

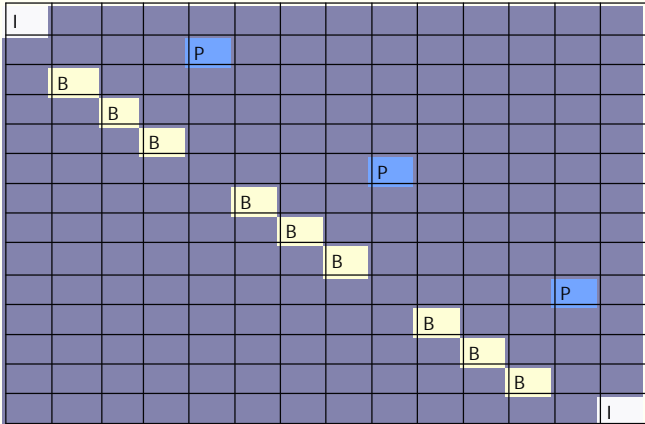





Typical Sequence of I, P, B frames

Display order (left to right)

Encoding order (top to bottom)

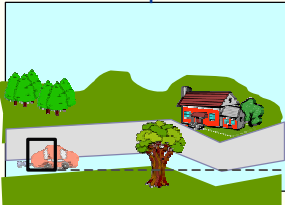


© 2004 Berkeley Design Technology, Inc. 23

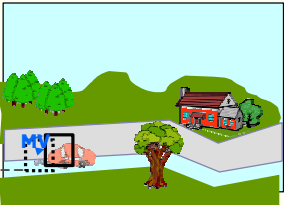


Motion Estimation

- Predict the contents of each macroblock based on motion relative to reference frame
 - Search reference frame for a 16x16 region that matches the macroblock
 - Encode motion vectors
 - Encode difference between predicted and actual macroblock pixels



Reference frame

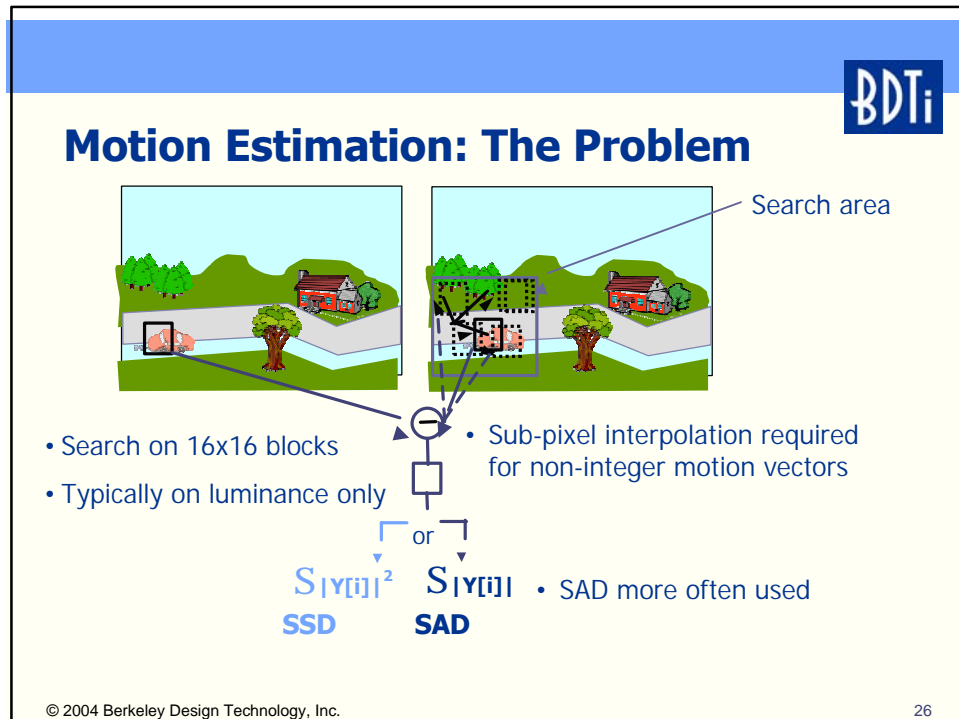
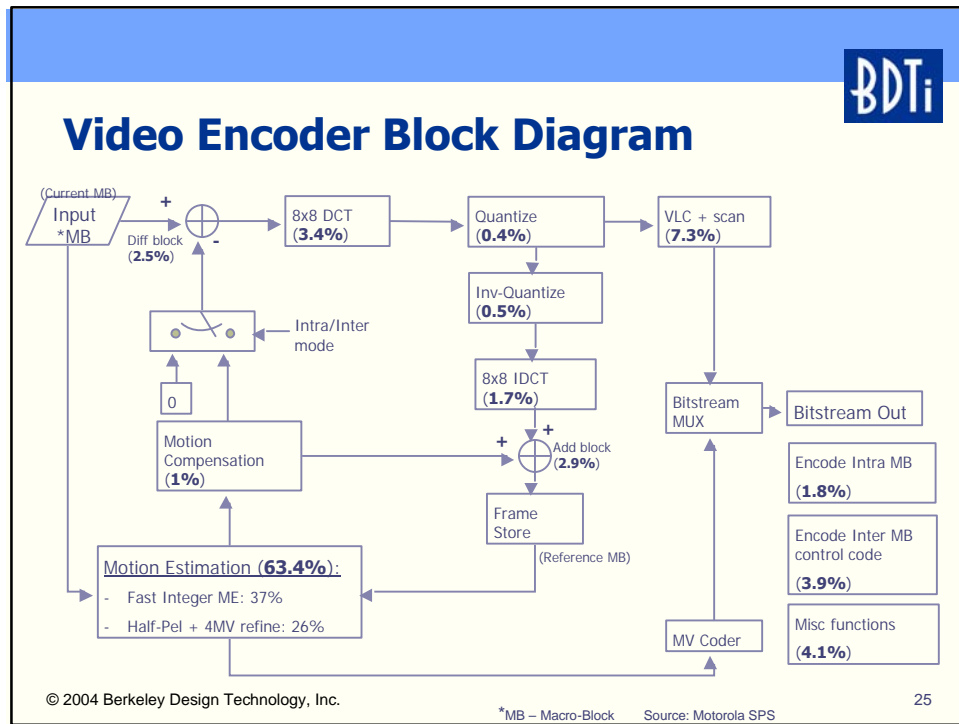


Current frame


**MV =
Motion
Vector**

© 2004 Berkeley Design Technology, Inc. 24

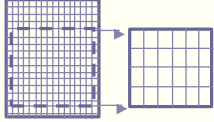
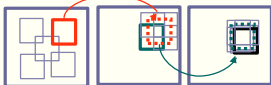

Smart Processor Picks for Consumer Media Applications



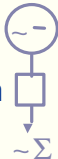
© 2004 Berkeley Design Technology, Inc.



Motion Estimation: Efficient Motion Vector Search






- Exhaustive search is impractical
- Evaluate only promising candidate motion vectors
 - Don't have to find absolute best match
 - Trade video quality for computational load
- Many methods in use
 - Often proprietary
 - Refine candidate vector selection in stages
 - Predict candidate vectors from surrounding macroblocks and/or previous frames



Motion vector search approach is a key differentiator between video encoder implementations


© 2004 Berkeley Design Technology, Inc. 27



Motion Estimation: Processing Req't's

- Compute load
 - Most demanding task in video compression
 - Up to 80% of total encoder processor cycles
 - Many search methods exist; requirements vary by method
 - May vary with video program content
 - Makes encoder computational demand several times greater than that of the decoder
 - Dominated by SAD computation
- Memory usage
 - Motion estimation requires reference frame buffers
 - Frame buffers dominate the memory requirements of the encoder
 - E.g., 152,064 bytes per frame @ CIF (352x288) resolution
 - High memory bandwidth required


© 2004 Berkeley Design Technology, Inc. 28



Motion Compensation: Processing Reqt's

- Motion compensation copies pixels from reference frame to predict current macroblock
 - Requires interpolation for non-integer motion vector values
- Compute load
 - Varies with video program content
 - Can require from 5% to 40% of total decoder processor cycles
 - MPEG-4 CIF (352x288) @ 30 fps:
 - Roughly 15-25 MHz on a TMS320C55x DSP (estimated)
- Memory usage
 - Requires reference frame buffers
 - Frame buffers dominate decoder memory requirements
 - Good memory bandwidth is desirable, but less critical compared to motion estimation


© 2004 Berkeley Design Technology, Inc. 29



Outline

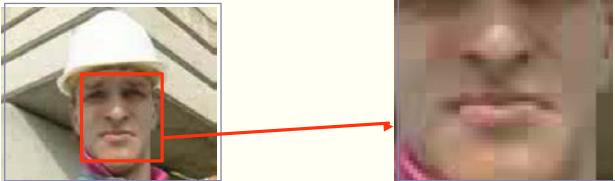
- Motivation and scope
- Still-image compression techniques
- Motion estimation and compensation
- **Reducing artifacts**
- Color conversion
- Conclusions

© 2004 Berkeley Design Technology, Inc. 30





Artifacts: Blocking and Ringing


- **Blocking:** Borders of 8x8 blocks become visible in reconstructed frame



- **Ringing:** Distortions near edges of image features

Original image   Reconstructed image (with ringing Artifacts)

© 2004 Berkeley Design Technology, Inc. 31



Deblocking and Deringing Filters

Low-pass filters are used to smooth the image where artifacts occur

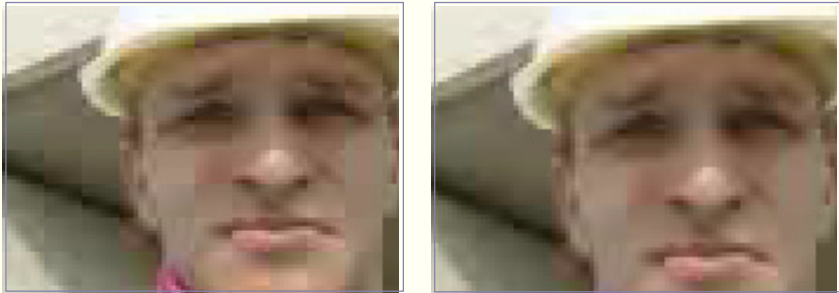
- **Deblocking:**
 - Low-pass filter the pixels at borders of 8x8 blocks
 - One-dimensional filter applied perpendicular to 8x8 block borders
- **Deringing:**
 - Detect edges of image features
 - Adaptively apply 2D filter to smooth out areas near edges
 - Little or no filtering applied to edge pixels in order to avoid blurring

© 2004 Berkeley Design Technology, Inc. 32

BDTi

Artifact Reduction

Example: Deblocking



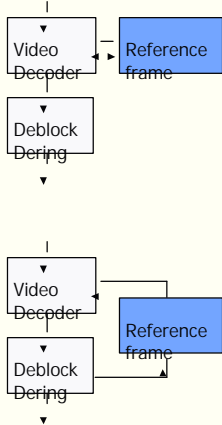
Original MPEG Still Frame Horizontally & Vertically Deblocked Still Frame

© 2004 Berkeley Design Technology, Inc. 33

BDTi

Artifact Reduction: Post-processing vs. In-loop

- Deblocking/deringing often applied after the decoder (post-processing)
 - Reference frames are not filtered
 - Developers free to select best filters for the application or not filter at all
- Deblocking/deringing can be incorporated in the compression algorithm (in-loop filtering)
 - Reference frames are filtered
 - Same filters must be applied in encoder and decoder
 - Better image quality at very low bit-rates



© 2004 Berkeley Design Technology, Inc. 34



Artifact Reduction: Processing Req't's

- Deblocking and deringing filters can require more processor cycles than the video decoder
 - Example: MPEG-4 Simple Profile, Level 1 (176x144, 15 fps) decoding requires 14 MIPS on ARM's ARM9E for a relatively complex video sequence
 - With deblocking and deringing added, load increases to 39 MIPS
 - Nearly 3x increase compared to MPEG-4 decoding alone!
- Post-processing may require an additional frame buffer

© 2004 Berkeley Design Technology, Inc.

35



Outline

- Motivation and scope
- Still-image compression techniques
- Motion estimation and compensation
- Reducing artifacts
- **Color conversion**
- Conclusions

© 2004 Berkeley Design Technology, Inc.

36



Color Space Conversion

- Need for color conversion
 - Capture and display video equipment: RGB...
 - ...while codecs use YUV
- Computational demand
 - 12 operations per pixel \Rightarrow 36 million operations/second for CIF (352x288) @ 30 fps
 - About 36 MHz on a TMS320C55x DSP
 - Not including interpolation of chrominance planes
 - Roughly 1/3 to 2/3 as many processor cycles as the video decoder

© 2004 Berkeley Design Technology, Inc.

37



Outline

- Motivation and scope
- Still-image compression techniques
- Motion estimation and compensation
- Reducing artifacts
- Color conversion
- **Conclusions**

© 2004 Berkeley Design Technology, Inc.

38



Conclusions

Understanding the computational and memory requirements of video compression is critical but challenging

- Application design choices are driven by computational and memory requirements
 - Algorithm selection, processor selection, software optimization
 - Video processing often stresses processing resources
- But video applications combine many different signal-processing techniques
 - Transforms, prediction, quantization, entropy coding, image filtering, etc.
- And there is large variation in computational and memory requirements among different applications
 - E.g., digital camcorder has vastly different requirements from a video-enabled cell phone, even when using the same compression standard



Conclusions, cont.

- Understanding computational load
 - Computational load of encoder is several times greater than that of decoder due to motion estimation
 - Computational load proportional to frame size and rate for most functions
 - Note: VLD computational load is proportional to bit rate
 - Post-processing steps—deblocking, deringing, color space conversion—add considerably to the computational load
- Computational load can be difficult to predict
 - Many different approaches to motion estimation
 - Computational load of some tasks can fluctuate wildly depending on video program content
 - E.g., motion compensation



Conclusions, cont.

- Understanding memory requirements:
 - Memory requirements dominated by frame buffers
 - A decoder that supports only I and P frames requires two frame buffers (current and reference)
 - A decoder that supports I, P, and B frames requires three buffers (current and two reference)
 - Deblocking/deringing/color conversion may require an additional buffer
 - Program memory, tables, other data comprise a non-negligible portion of memory use
 - But this portion is still several times smaller than frame buffers
- High memory use often necessitates off-chip memory
 - Off-chip memory bandwidth can be a performance bottleneck

© 2004 Berkeley Design Technology, Inc.

41



Conclusions, cont.

- Video compression used in many products
 - DVDs, digital TV, personal video recorders, Internet video, multimedia jukeboxes, video-capable cell phones and PDAs, camcorders...
- Different products have different needs
 - Wide range of frame sizes and rates, video quality, bit rates, post-processing options, etc.
 - Result in wide range of computational and memory requirements
- Need to understand the operation of video codecs
 - To understand computational and memory requirements
 - To select codecs, processors, software modules
 - To optimize software

© 2004 Berkeley Design Technology, Inc.

42

Smart Processor Picks for Consumer Media Applications

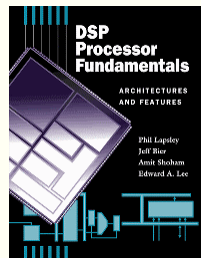


For More Information...

www.BDTI.com

Free Information

- White papers/presentation slides on
 - DSP software optimization
 - Streaming media implementation
 - Processor architectures and performance
 - Digital audio compression
- Article reprints on DSP-oriented processors and applications
 - EE Times
 - IEEE Spectrum
 - IEEE Computer and others
- comp.dsp FAQ



2004 Edition

43

© 2004 Berkeley Design Technology, Inc.