

Multimedia

Randy Bryant

CS 740

October 27, 1997

Topics

- **Overview**
- **Encoding & Compression**
 - Audio, still images, video
 - JPEG, MPEG
- **Storage & Transmission**
 - Rates & Capacities
- **Processing**
 - Architectural Extensions
 - Intel MMX

What is (Digital) Multimedia?

- *Integration of two or more media using computer technology*
 - E.g., audio, video

Reproductions

- Still or video images, recorded music or speech

Synthesized

- Animations, MIDI recordings, virtual reality

Major Driver of Computer Technology

- High market demand
- High computation, communication, and storage requirements
 - Real time processing required
 - Requirements scale rapidly with increased quality
 - » E.g., quadratically with image resolution

Glossary

Storage Sizes

- KB = 1024 bytes MB = 1,048,576 bytes GB = 1,073,741,824 bytes

Acronym Interpretation

- NTSC U.S. / Japan television standard
- JPEG Still image compression & encoding format
- MPEG Moving image compressing & encoding format
- CD / CDROM Based on compact disk technology
- DVD Digital Video Disk

Complexity Example

NTSC” Quality Computer Display

- 640 X 480 pixel image
- 3 bytes per pixel (Red, Green, Blue)
- 30 Frames per Second

Bandwidth

- 26.4 MB/second
- Corresponds to 180X CDROM
- Exceeds bandwidth of almost all disk drives

Storage

- CDROM would hold 25 seconds worth
- 30 minutes would require 46.3 GB

Observation

- Some form of compression required

Lossless Data Compression



Principle

- Reconstructed = Original
- Exploit property that some bit sequences more common than others
- Time (compress/decompress) – Space (data size) tradeoff

Huffman-Ziv encoding

- Build up table of frequently occurring sequences
- E.g., Unix compress (.Z), DOS Gzip (.zip)
- Achieve compression ratio ~ 2:1 (text, code, executables)

Huffman Encoding

- Assign shorter codes to most frequent symbols

Lossy Compression



- Reconstructed data approximates original
- Want compression ratios $\geq 10:1$

Tradeoffs

- Time vs. space
- Space vs. quality
 - Greater compression possible if looser approximation allowed

Application Dependent

- Exploit characteristics of human perception
- Examples
 - Eye's sensitivity to color variations less than to brightness

Digitizing Analog Waveforms



Sampling

- Measure signal value at regular time intervals
- $dt = 1/\text{Sampling frequency}$
- Nyquist Theorem
 - Must sample at $\geq 2 \times$ highest frequency signal component
 - Otherwise get “aliasing” between low and high frequency values



Analog to Digital Conversion

- Convert each sample into k-bit value
- Limits dynamic range
- Low resolution gives “quantization error”

Audio Encoding

Frequency Response

- Humans can hear sounds ranging from around 5 Hz to 15 KHz
- Piano notes range from 15 Hz to 15KHz
- Telephone limits frequency range to between 300Hz & 3 KHz

Dynamic Range

- Humans can perceive sounds over 10 order of magnitude dynamic range
- 100 Decibels
 - Every 3 dB corresponds to doubling sound intensity

Compact Disk Recording

Parameters

- **44,100 samples per second**
 - Sufficient for frequency response of 22KHz
- **Each sample 16 bits**
 - 48 dB range
- **Two independent channels**
 - Stereo sound
 - Dolby surround-sound uses tricks to pack 5 sound channels + subwoofer effects

Bit Rate

- **44,100 samples/second X 2 channels X 2 bytes = 172 KB / second**

Capacity

- **74 Minutes maximum playing time**
- **747 MB total**

CD ROM Technology

Basis

- Use technology developed for audio CDs
- Add extra 288 bytes of error correction for every 2048 bytes of data
 - Cannot tolerate any errors in digital data, whereas OK for audio

Bit Rate

- $172 * 2048 / (288 + 2048) = 150 \text{ KB / second}$
 - For 1X CDROM
 - $N \times$ CDROM gives bit rate of $N * 150$
 - E.g., 12X CDROM gives 1.76 MB / second

Capacity

- $74 \text{ Minutes} * 150 \text{ KB / second} * 60 \text{ seconds / minute} = 650 \text{ MB}$
- Typically around 527 MB to reduce manufacturing cost

Image Encoding & Compression

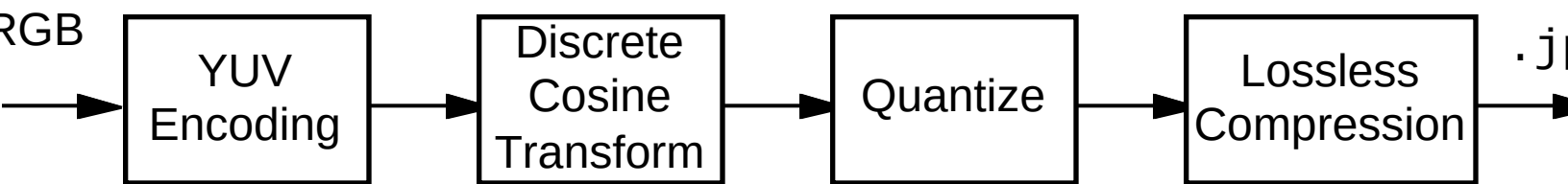
Computer-Generated Images

- Limited number of colors (e.g., ≤ 256)
- Large monochrome regions
- Sharp boundaries between regions
- GIF encoding works well
 - Lossless compression of 8-bit color images

Natural Scenes

- Large number of colors (want 24-bit quality)
- Widely varying intensities and colors
- Boundaries not sharp
- Can eliminate details for which human perception is weak
- JPEG encoding works well
 - Lossy compression based on spectral transforms
 - Can achieve from 10:1 to 20:1 compression with little loss in quality

JPEG Encoding Steps



Encoding

- Convert to different color representation
- Typically get 2:1 compression

Discrete Cosine Transform (DCT)

- Transform 8 X 8 pixel blocks

Quantize

- Reduce precision of DCT coefficients
- Lossy step

Lossless Compression

- Express image information in highly compressed form

YUV Encoding

Computation

- RGB numbers between 0 and 255
- *Luminance* Y encodes grayscale intensity between 0 and 255
- *Chrominance* U, V encode color information between -128 and +127
 - Similar to Color (Hue) and Tint (Saturation) controls on color TV

Conversion

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Values saturate at ends of ranges

Color Subsampling

- Average U,V values over 2 X 2 blocks of pixels
- Human eye less sensitive to variations in color than in brightness

Image Examples



- Scanned from CMU catalog
- 248 X 324 X 3B
- Moiré interference pattern caused by scanning digitized image

Color Subsampling



- **2:1 Compression**
- **No visible effect**

Discrete Cosine Transform

Image Partitioning

- Divide into 8 X 8 pixel blocks
- Express each block as weighted sum of cosines
 - Similar to Fourier Transform

Transform Computation

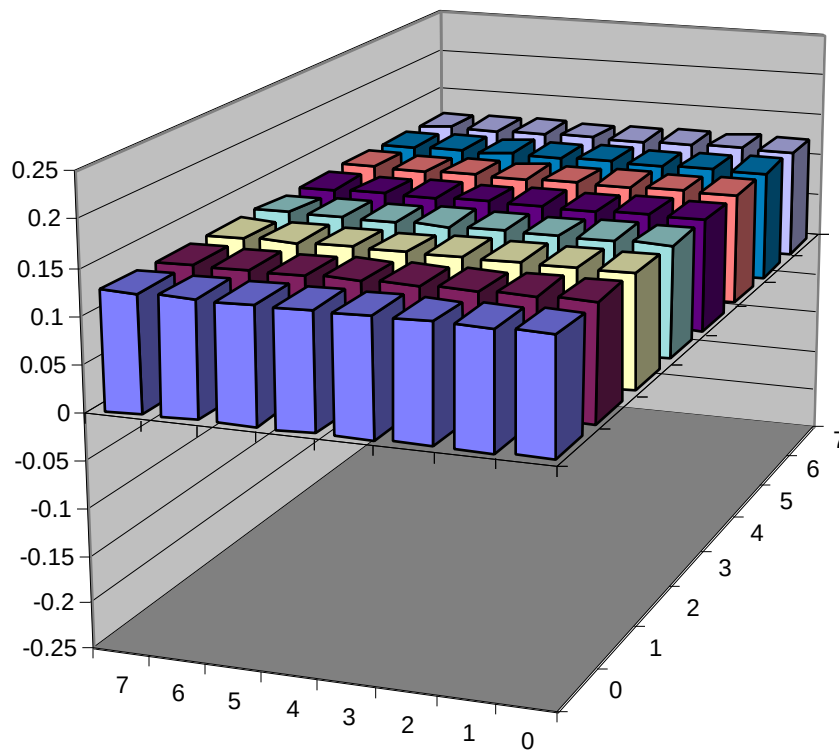
$$F(u,v) = K(u,v) * \text{Sum}(i = 0,7) \text{Sum}(j = 0,7) \\ f(i,j) * \cos([2i+1] * u * p/16) * \cos([2j+1] * v * p/16)$$

Inverse Transform

$$f(i,j) = \text{Sum}(u = 0,7) \text{Sum}(v = 0,7) \\ k(u,v) * F(u,v) * \cos([2i+1] * u * p/16) * \cos([2j+1] * v * p/16)$$

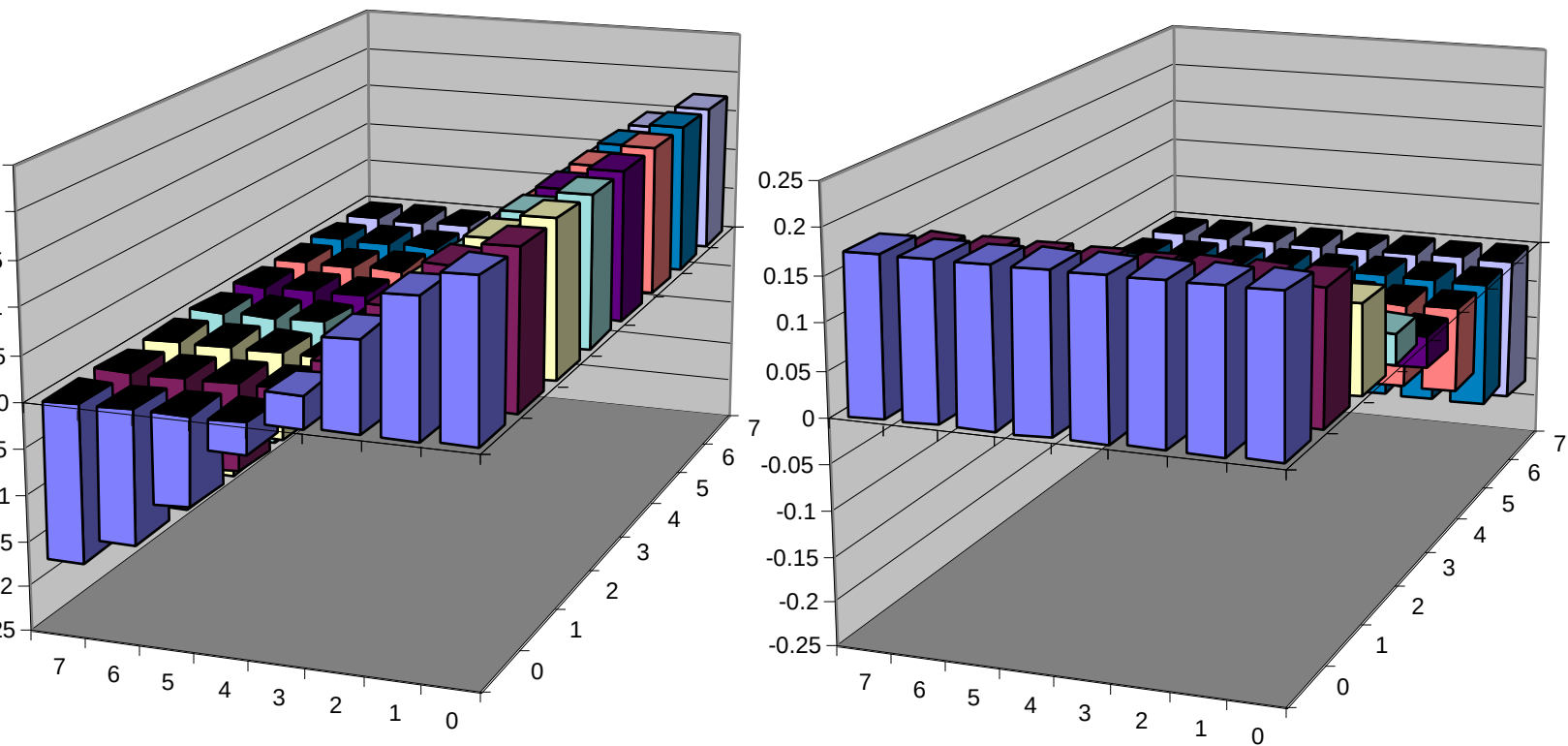
- Expresses image as sum of discretized cosine waveforms
 - Would be exact, except for roundoff and saturating values
- Each waveform weighted by coefficient $F(u,v)$
- Low values of u, v correspond to slowly varying waveforms

Inverse DCT of Selected Coefficients



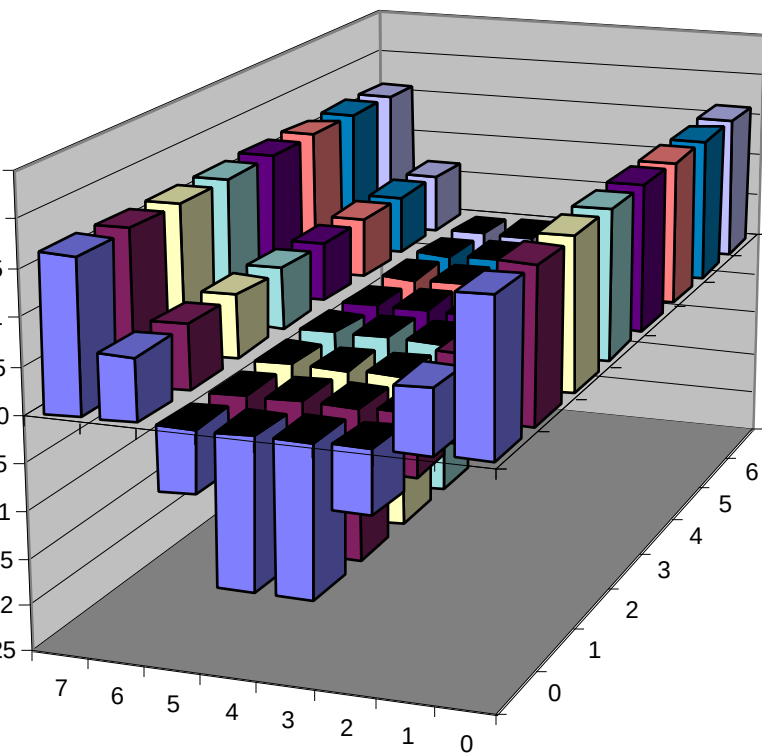
- **Coefficient (0, 0)**
 - i.e., $F(0, 0) = 1$, all others = 0
- **Characterizes overall average**

Inverse DCT of Selected Coefficients

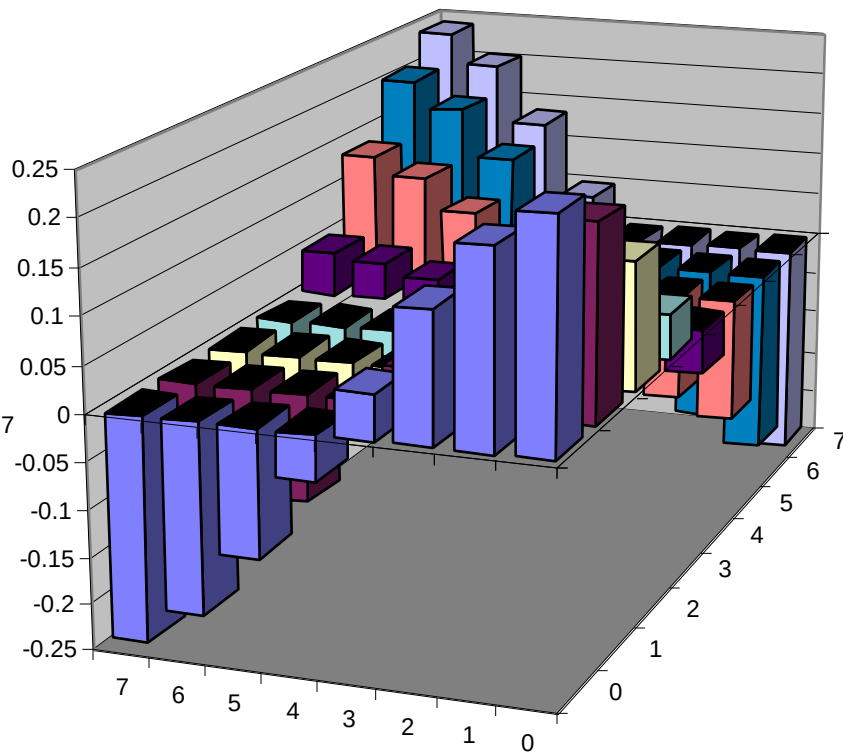


- Coefficients (1,0) and (0,1)
- Capture horizontal or vertical gradient

Inverse DCT of Selected Coefficients

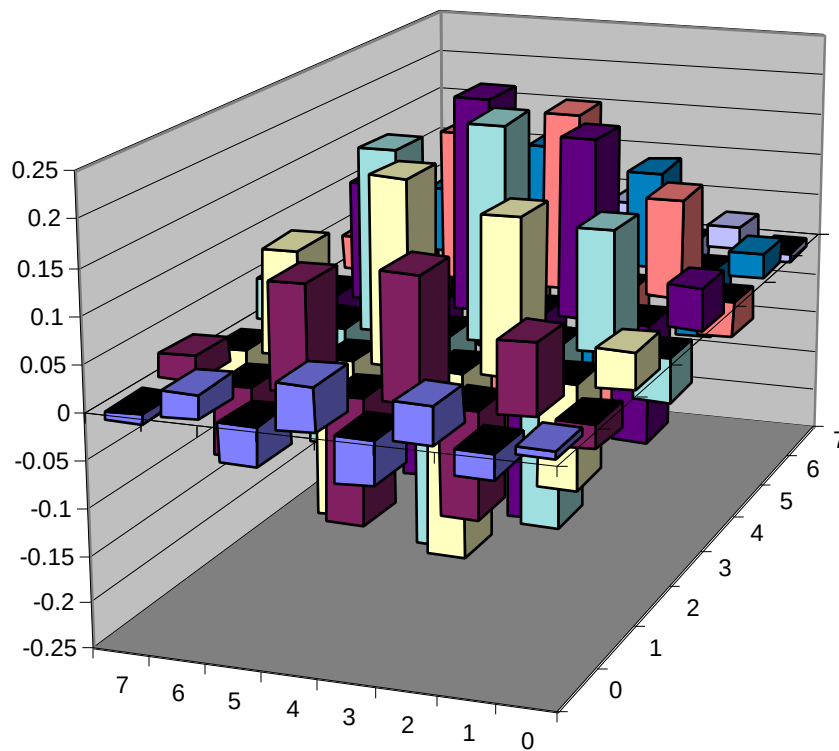


- Coefficient (2,0)
- Captures vertical banding



- Coefficient (1,1)
- Captures diagonal variation

Inverse DCT of Selected Coefficients



- **Coefficient (7,7)**
- **Captures high spatial variations**

Quantization

Quantization Coefficient $Q(u,v)$ for each waveform (u,v)

- Approximate $F(u,v)$ as $Q(u,v) * \text{Round}[F(u,v) / Q(u,v)]$
 - E.g., if Q is 2^k , just set low order k bits to 0
- High value of Q gives coarser approximation

Selecting Q 's

- Increase to get greater compression
 - Major source of loss in JPEG
- Generally use Q 's that increase with u and v
 - High spatial frequencies not as important
 - Hope that many coefficients will be set to 0

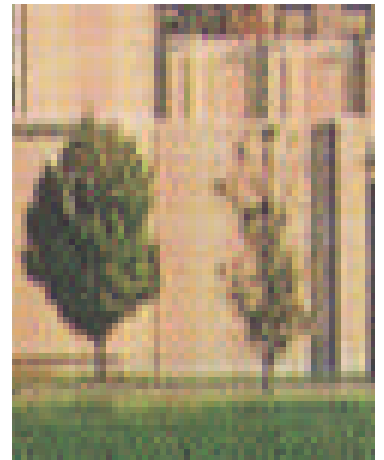
JPEG Compression Examples

28:1 Compression

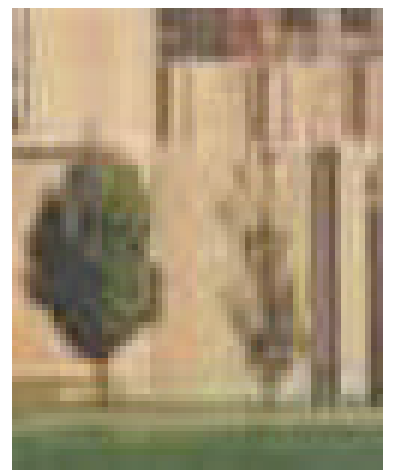


- **Quality still reasonable**
- **Some loss of fine detail**

Original



Compressed



JPEG Compression Examples



66:1

- **Quality drops dramatically as increase compression ratio**
- **Throws out color information first**

JPEG Compression Examples



86:1

101:1



DCT Quantization Effects

Blow Up Sections of Low Quality Images

- See 8 X 8 blocks
- Only low frequency coefficients for Y
- Extreme subsampling for U, V
 - Single value for 16 X 8 pixels



MPEG Video Encoding

MPEG-1

- **Targeted to “VHS” quality video on CDROM**
 - 352 X 240 pixels
 - Display on computer screens
- **Two forms of lossy compression**
 - Spatial compression similar to JPEG
 - Temporal compression exploiting similarity between successive frames
 - » E.g., stationary background, panning

MPEG-2

- **Broader range of applications**
 - Including Digital Video Disk (DVD) players
- **Support for fancier features**
- **Aim for baseline of 720 X 480 pixels**

Baseline MPEG-1

Video Channel

- 352 X 240 pixels, with subsampling of chrominance to 176 X 120
- 30 frames per second
- 26:1 compression drops bit rate to 143 KB / second

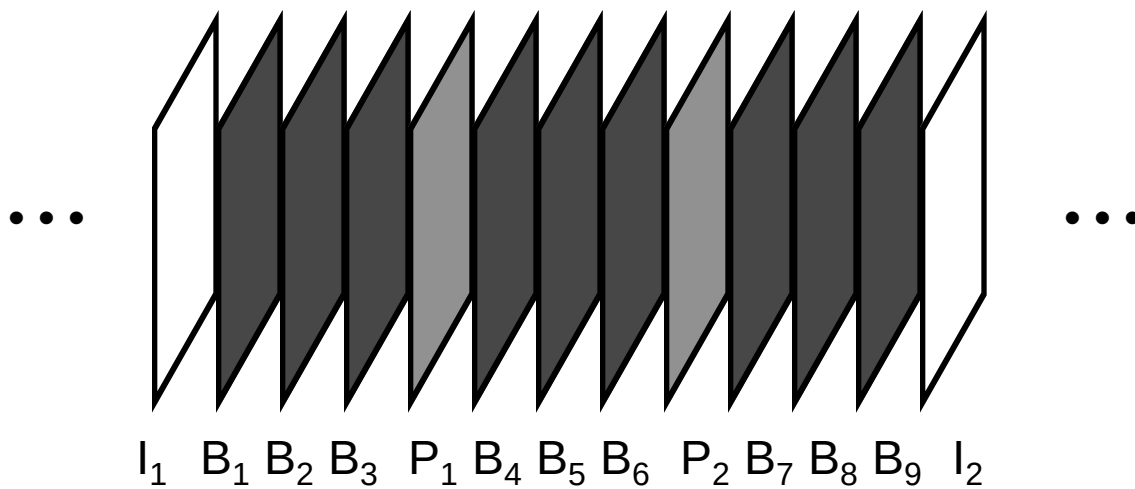
Audio

- Compress CD sound by 6:1
- Bit rate = 32 KB / second

Performance

- 175 KB / second matches performance of early CD ROM drives
- Store > 1 hour on CD
- Software-only decoders running on PC-class machines cannot decode fast enough
 - Typically limited to 8–10 FPS

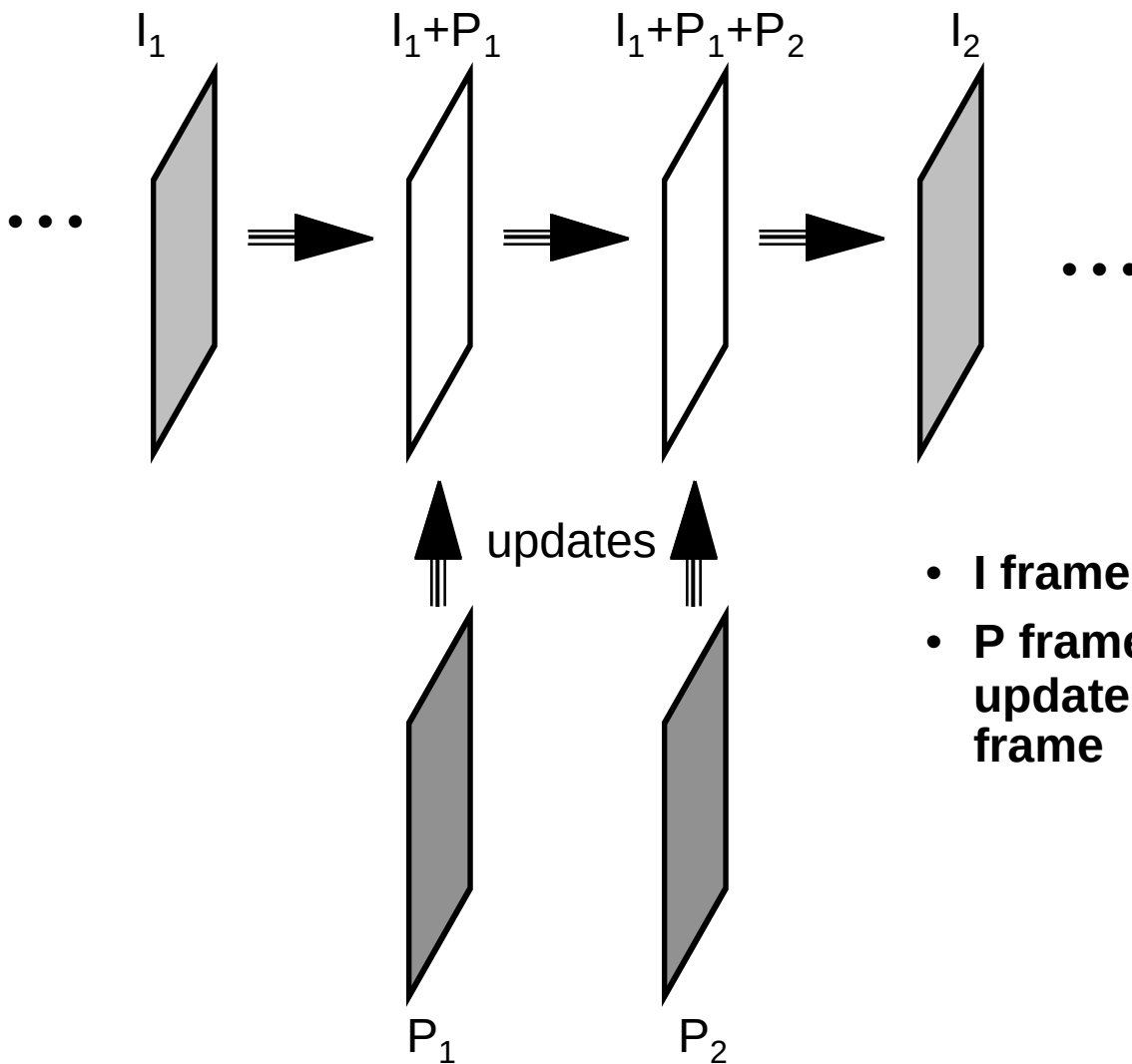
MPEG Encoding



Frame Types

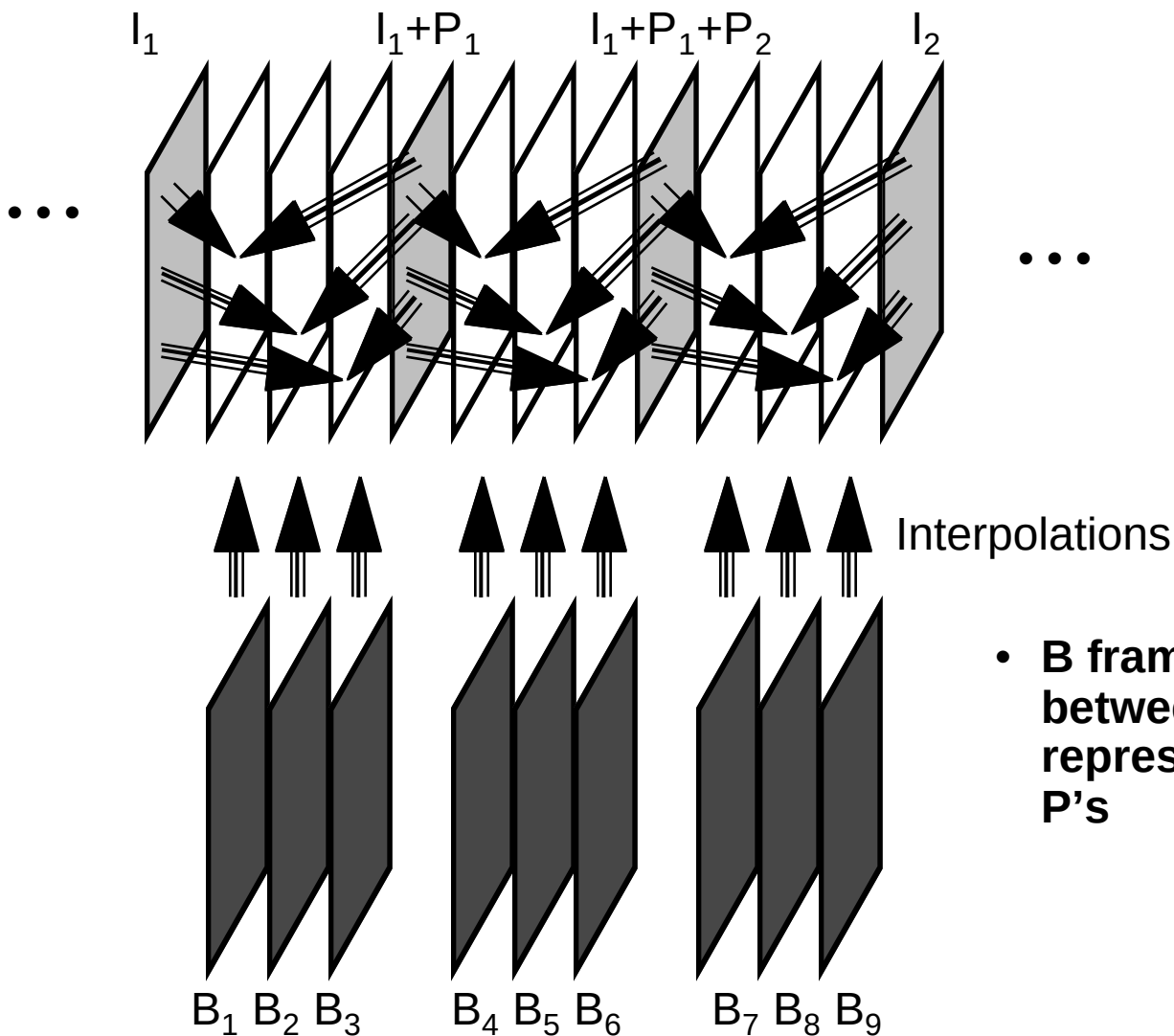
I	Intra	Encode complete image, similar to JPEG
P	Forward Predicted	Motion relative to previous I and P's
B	Backward Predicted	Motion relative to previous & future I's & P's

Frame Reconstruction



- I frame complete image
- P frames provide series of updates to most recent I frame

Frame Reconstruction (cont).



- B frames interpolate between frames represented by I's & P's

Updates / Interpolations

Describe how to construct 16 X 16 block in new frame

- Block from earlier frame
- Block from future frame (B frame only)
- Additive correction

Encoding

- All blocks coded using format similar to JPEG

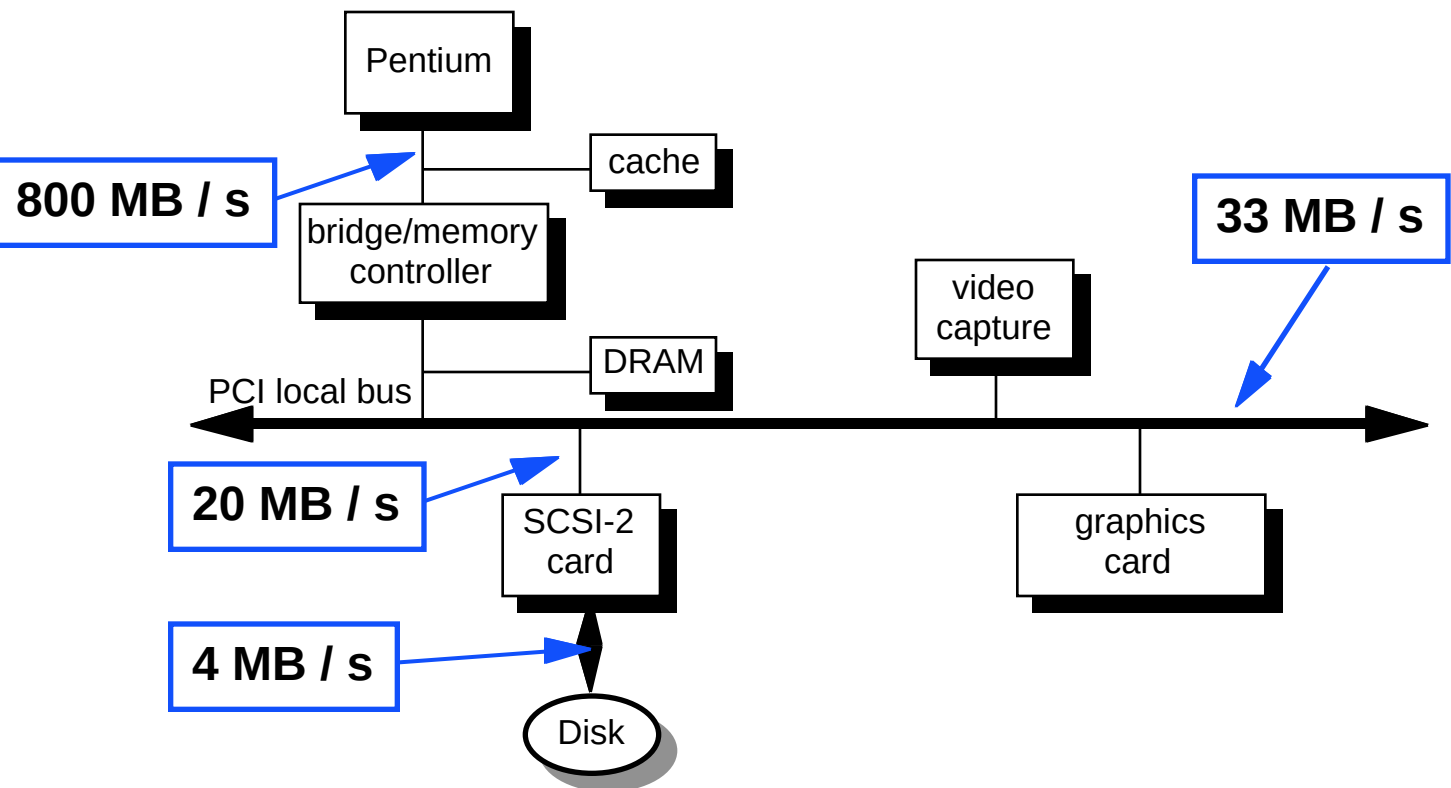
Typical Numbers

- 320 X 240 X 148 frames
- 10 I frames 18.9 KB average 6:1 compression
- 40 P frames 10.6 KB average 11:1 compression
- 98 B frames 0.8 KB average 141:1 compression
- 148 total 4.7 KB average 24:1 compression

Bandwidth Requirements

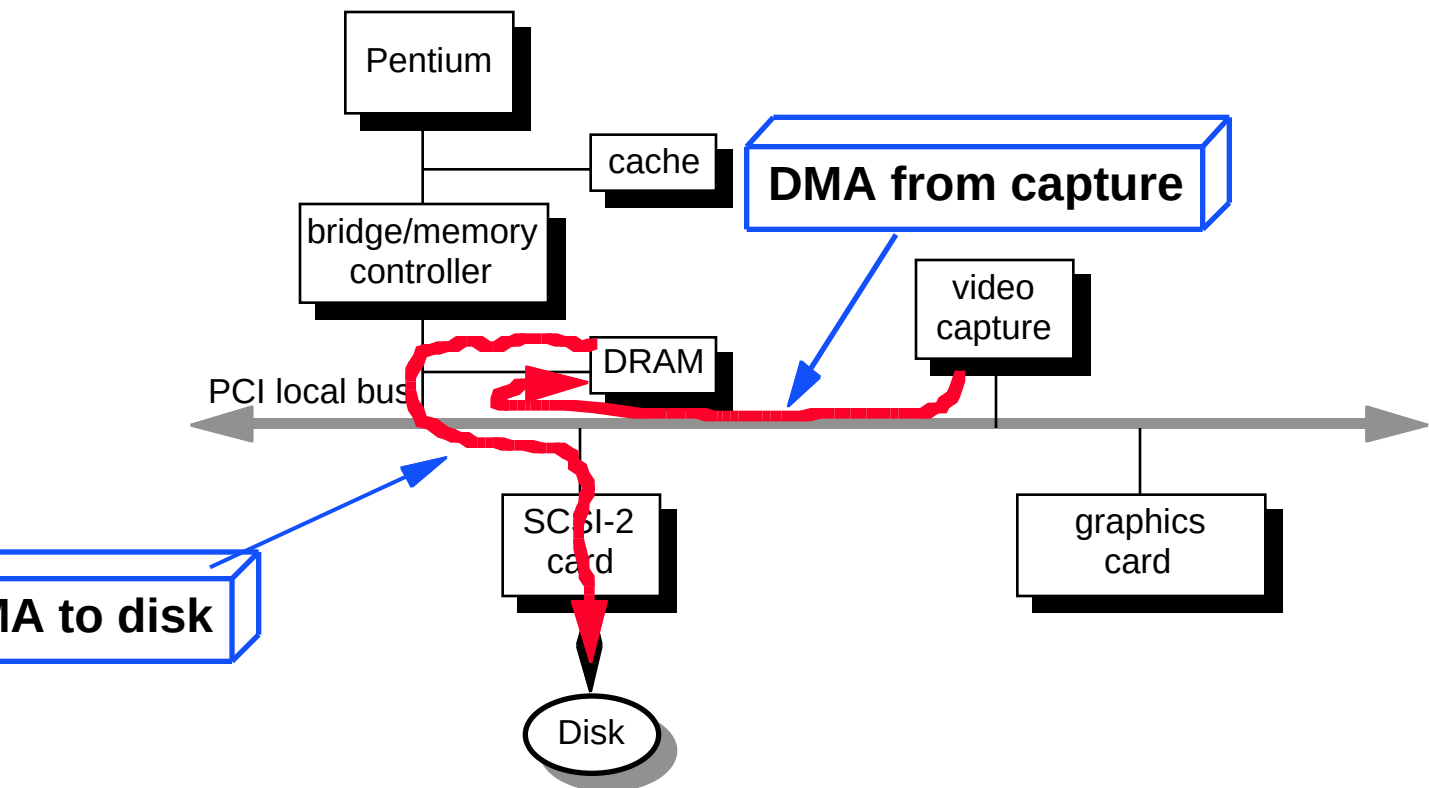
Consider MPEG-1 (352 X 240) vs. MPEG-2 (720 X 480)

- Uncompressed 3.6 MB / s 14.8 MB / s
- Compressed 0.14 MB / s 0.43 MB / s



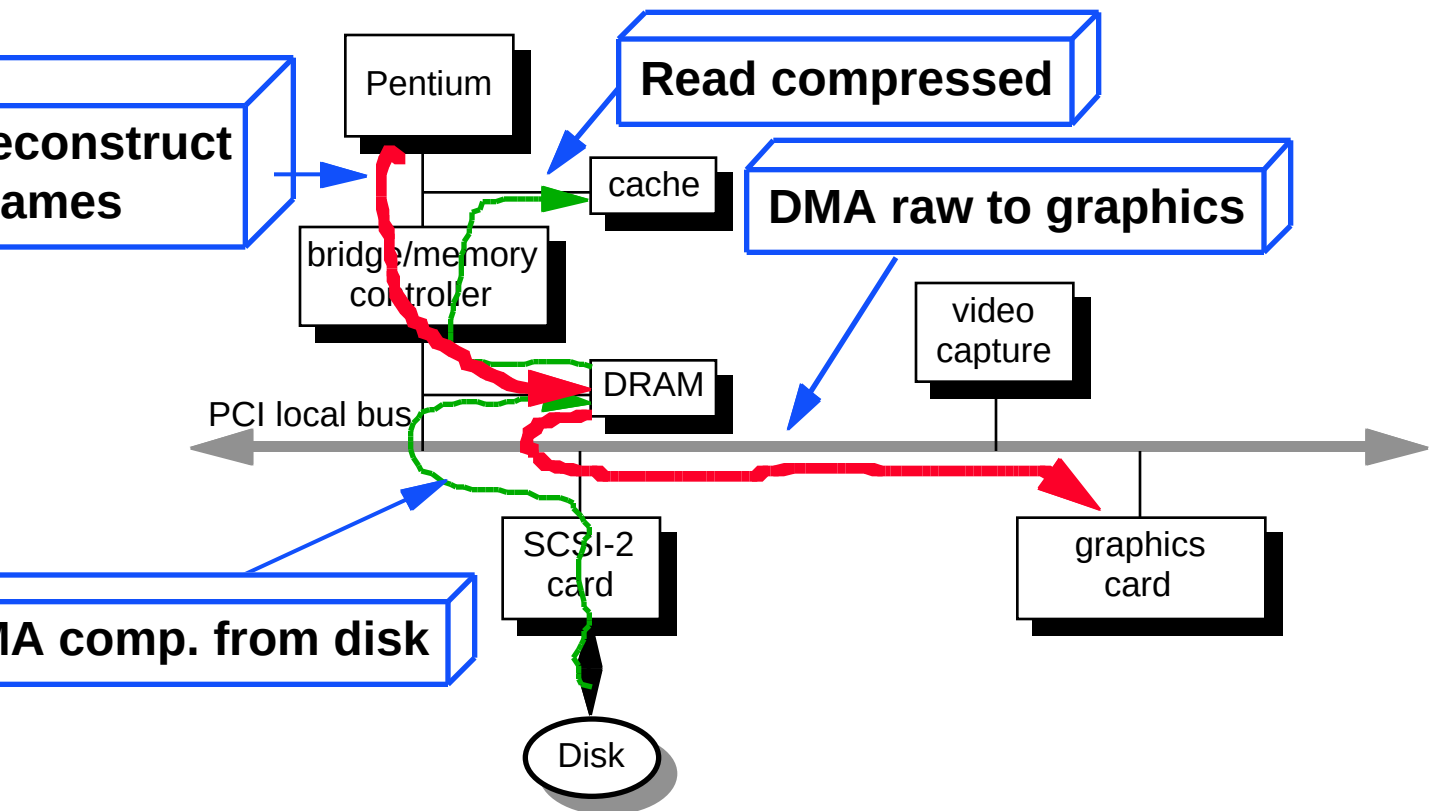
Capture Raw Video to Disk

	MPEG-1	MPEG-2	
PCI	7.2 MB / s	29.6 MB / s	(shaky)
SCSI-2	3.6 MB / s	14.8 MB / s	(OK)
Disk	3.6 MB / s	14.8 MB / s	(No Way!)



Decompress & Playback

	MPEG-1	MPEG-2	
PCI	7.5 MB / s	30.5 MB / s	(shaky)
SCSI-2	0.14 MB / s	0.43 MB / s	(OK)
Disk	0.14 MB / s	0.43 MB / s	(OK)



ISA Extensions to Support Multimedia

	MIPS V/ MDMX	Intel MMX	Sun VIS	HP MAX2	Alpha MVI
Number of Registers	32	8	32	31	31
Register Type	MM/FP†	MM/FP†	MM/FP†	Integer	Integer
Parallel Arithmetic	8 × 8 bits 4 × 16 bits	8 × 8 bits 4 × 16 bits	4 × 16 bits 2 × 32 bits	4 × 16 bits	Min/max only
Saturating?	No	Yes	Yes	Yes	n/a
Non-saturating?	Yes	Yes	No	Yes	n/a
Free Operands?	Yes	No (2)	Yes	Yes	Yes
Parallel Multiplies	4 or 8	4	4	None	None
Multiply/Add	8 × 8 → 24 16 × 16 → 48	16 × 16 → 32	8 × 16 → 16	Shift-and-add	None
Vector-to-Scalar?	Yes	No	No	No	No
Parallel Shifts?	Yes	Yes	No	Yes	No
Parallel Average?	No	No	No	Yes	No
Parallel Compare?	Yes	Yes	Yes	No	No
Shuffle/Unpack?	Yes	Yes	Yes	Yes	No
Interleave?	Yes	Yes	Yes	Yes	No
Commute?	No	No	No	Yes	No
Round Error?*	No	No	Yes	No	Yes
Block Load/Store?	No	No	Yes	No	No
Parallel FP?	Yes	No	No	No	No

Microproces
Report
11/18/96

- CPU manufacturers extending architectures to support multimedia
- Low precision (1 & 2 byte), saturating arithmetic
- Vector operations

MIPS MMDX

erload Use of FP egisters

64 bits total

View as vector of small
integers

- 8 single byte words
- 4 two-byte words

Operations

- Addition, multiplication
 - » Different vector modes
 - » Saturating
- Rearrange
 - » Shuffle, pack, unpack

Microprocessor Report 11/18/

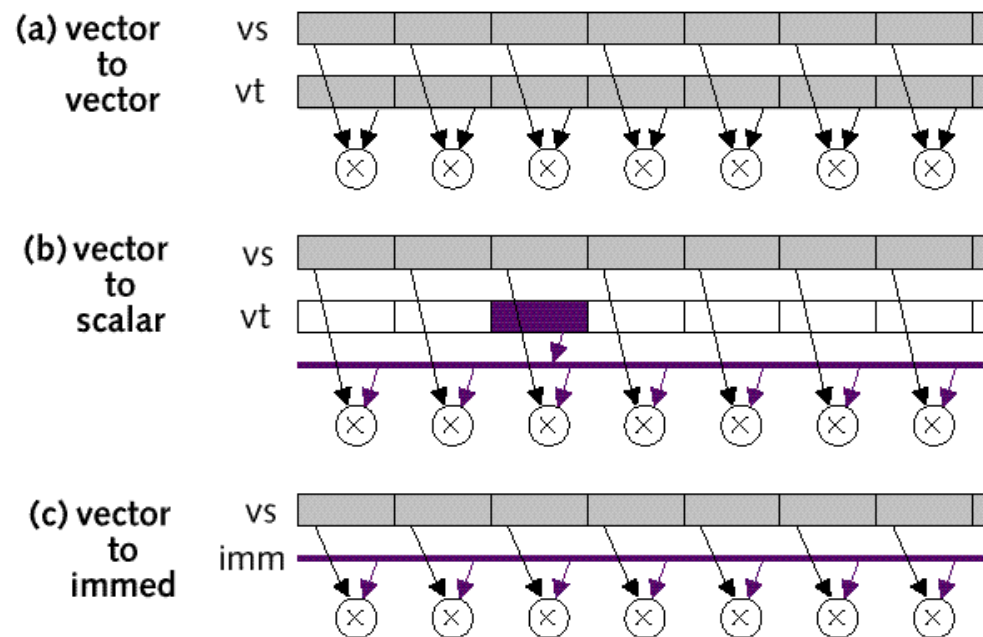


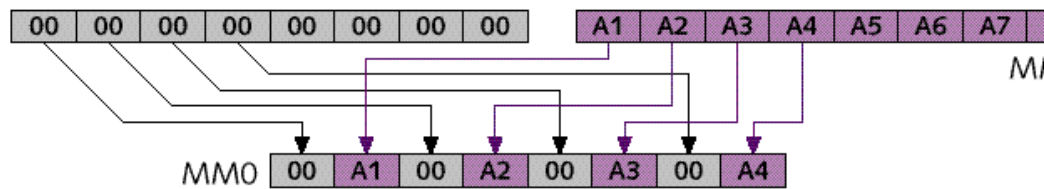
Figure 1. MDMX instructions operate in (a) vector-to-vector mode, (b) vector-to-scalar mode, and (c) vector-to-immediate mode.

Similar to MIPS

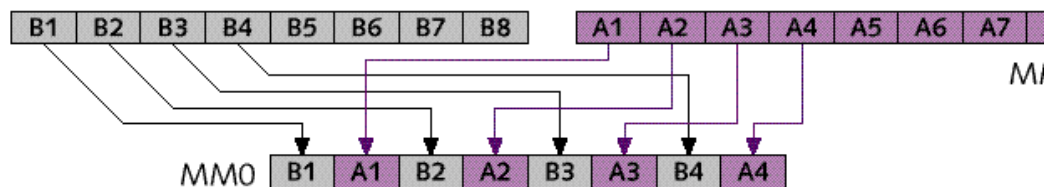
- **Overload use of FP registers**
 - Only 8 available
- **View as vector of integers**
 - 1, 2, or 4 bytes each

Intel MMX

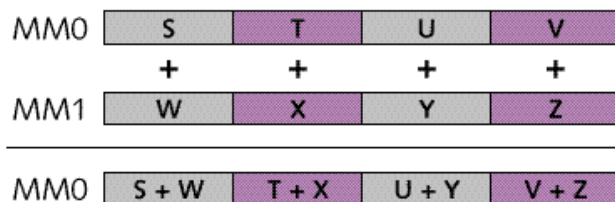
(a) PUNPCKHBW MM0,ZERO



(b) PUNPCKHBW MM0,MM1



(c) PADDW MM0,MM1



Microprocessor Report 3/5/96

Figure 3. (a) Unpack operation converts packed bytes to words with zero extension. (b) PUNPCK can also interleave bytes from two MM registers. (c) Parallel add instruction calculates four 32-bit sums simultaneously.

Digital MVI

Standard Alpha

- **Can do most multimedia tasks in software**
 - Sledgehammers make good fly swatters
- **Cannot do MPEG-2 encoding in real time**
 - Motion estimation is very demanding
 - Must compare 16 X 16 blocks with blocks from other frames

Extensions

- **Packing & unpacking smaller (1, 2, 4 byte) integers**
- **Special instruction used in motion estimation**
 - Perr Ra, Rb, Rc
 - Ra, Rb vectors of single byte words
 - $Rc = \text{SUM}(i = 0, 7) \mid Ra_i - Rb_i \mid$