Video-Conferencing System

Evan Broder
Chris Post
Goals

- Video conferencing system
- Transmit video and audio on low-bandwidth serial line
- Compress video
- Downsample audio
- Resistant to dropped packets
Specifications

- **Video**
  - 320x240
  - 15 fps

- **Audio**
  - 16 kHz
  - 8 bit

- **250 kbps serial**
  - Approx. 56x compression
Compression: What is JPEG

- Frame-based compression
- YCrCb color space
- Chroma subsampling
- Encoding
  - DCT
Compression: What is JPEG

- Frame-based compression
- YCrCb color space
- Chroma subsampling
- Encoding
  - DCT
  - Quantization
Compression: What is JPEG

- Frame-based compression
- YCrCb color space
- Chroma subsampling
- Encoding
  - DCT
  - Quantization
  - Entropy/Huffman encoding
- Decoding
Logic Overview
Logic Overview

Unpacker (broder) → Huffman Decoder (ccpost) → Dequantizer (broder) → iDCT (broder) → Block Reassembler (broder)

Physical connection

AC97 Interpolate/Playback (ccpost) → Speaker

Upsampler (broder)

YCrCb to RGB Converter (broder)

Memory Interface (Bank 1) (ccpost)

VGA Driver (ccpost) → Monitor
Memory Interface

Downsampler
- [7:0] data
- [8:0] x
- [7:0] y
- [1:0] channel
- write_ready

Memory Interface (Bank 0)

Block Splitter
- [8:0] x
- [7:0] y
- [1:0] channel
- [63:0] row
- read_request
- read_done
DCT

- Input is one row per clock cycle
- Output is one column per clock cycle
- Width changes because DCT outputs more information

2-D DCT

[63:0] row_in
valid_in

[95:0] column_out
Valid_out
DCT

- Only one 1-D DCT module
- Latency is approximately 20 clock cycles
1-D DCT

- Using Winograd algorithm
- Original algorithm
  - 64 multiplies
  - 64 adds
- Winograd
  - 6-stage pipelined
  - 29 add/subs
  - 5 multiplies

<table>
<thead>
<tr>
<th>TABLE 1 – 1-D DCT algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STEP 1</strong></td>
</tr>
<tr>
<td>b0=a0+a7</td>
</tr>
<tr>
<td>b1=a1+a6</td>
</tr>
<tr>
<td>b2=a3−a4</td>
</tr>
<tr>
<td>b3=a1−a6</td>
</tr>
<tr>
<td>b4=a2+a5</td>
</tr>
<tr>
<td>b5=a3+a4</td>
</tr>
<tr>
<td>b6=a2−a5</td>
</tr>
<tr>
<td>b7=a0−a7</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Entropy/Huffman Encoder

- Zigzag ordering from upper-left
- Result is Huffman coded
- Less bits to store 0 and small numbers
- Run-length to pack runs of same value
- Special end of block code when only 0s left
Packer

- Buffers output from each channel
- When it has all the data it needs for one packet, it frames and transmits the packet
- Keep track of coordinates of video blocks

**Video:**

<table>
<thead>
<tr>
<th>Start</th>
<th>Channel</th>
<th>Len</th>
<th>X</th>
<th>Y</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
</table>

**Audio:**

<table>
<thead>
<tr>
<th>Start</th>
<th>Channel</th>
<th>Len</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
</table>
Reception and Decoding

- Blocks decoded in order received
- Block decoding operations are simple inverses of the encoding counterparts
- Each block gets coordinates transmitted with it through decoding
- Block reassembler uses these coordinates to do block buffering
Timeline

- **Now**: First version of 2-D DCT
- **After Thanksgiving**
  - Final version of 2-D (i)DCT
  - (De)Quantizer
  - NTSC Decoder
  - Downsampler
  - VGA Driver
- **Nov 28**
  - Entropy/Huffman Encoder/Decoder
  - Packer/Unpacker
- **Dec 5**
  - Memory Interfaces
  - Splitter
  - Reassembler
  - Upsampler
  - Color Space Converter
Questions?