Performance of a Software MPEG Decoder

ACM Multimedia 93

Ketan Patel, Brian C. Smith, and Lawrence A. Rowe
Computer Science Division - EECS
University of California at Berkeley
(larry@cs.berkeley.edu)

Outline

MPEG video compression
Software decoder
Performance
Future plans

MPEG Video Stream

Different types of frames
I - intracoded frame
P - forward predicated frame
B - bi-directional/interpolated frame
Frames delivered in decode order

Frame Compression

I frames:

P/B frames:
**Decoder**

Algorithm - parse bitstream and undo compression

Written in C (15K lines of code)

Ported everywhere
- Unix/X Windows
- PC/Windows 3.x
- Macintosh

Code freely distributed
FTP from toe.cs.Berkeley.edu: pub/multimedia

**Relative Performance**

Original code spent 60-80% time dithering

Using ordered dither into a fixed color map ...

<table>
<thead>
<tr>
<th>Operation</th>
<th>%Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parsing</td>
<td>17%</td>
</tr>
<tr>
<td>Inverse DCT</td>
<td>14%</td>
</tr>
<tr>
<td>Reconstruction</td>
<td>32%</td>
</tr>
<tr>
<td>Dithering</td>
<td>24%</td>
</tr>
<tr>
<td>Misc arithmetic</td>
<td>10%</td>
</tr>
<tr>
<td>Other</td>
<td>3%</td>
</tr>
</tbody>
</table>

IDCT is not the bottleneck
Reconstruction and dithering are bottleneck, problem is memory bandwidth

**Real Time?**

Canyon: 144x112, 49:1 compression (1:1:4, 0.49 bpp, 0.24Mbs)
Flower: 320x240, 50:1 compression (1:4:10, 0.49 bpp, 1.00Mbs)
Berkeley MPEG decoder v2.0 running on Unix

<table>
<thead>
<tr>
<th>Machine</th>
<th>Canyon</th>
<th>Flower</th>
<th>Clock</th>
<th>Cache (I/D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC AXP 3k/500</td>
<td>43.1</td>
<td>8.9</td>
<td>150 MHz</td>
<td>?</td>
</tr>
<tr>
<td>HP 9k/750</td>
<td>74.7</td>
<td>15.4</td>
<td>66 MHz</td>
<td>256/256</td>
</tr>
<tr>
<td>Intel 486DX2</td>
<td>13.4</td>
<td>3.3</td>
<td>66 MHz</td>
<td>8/0</td>
</tr>
<tr>
<td>SGI Indigo</td>
<td>54.6</td>
<td>11.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sun Sparc 10/30</td>
<td>38.1</td>
<td>8.2</td>
<td>36 MHz</td>
<td>16/20</td>
</tr>
<tr>
<td>Intel 486DX2</td>
<td>22.0</td>
<td>5.5</td>
<td>running v1.2 on Windows</td>
<td></td>
</tr>
<tr>
<td>DEC AXP 3k/500</td>
<td>68.3</td>
<td>16.3</td>
<td>running DEC decoder</td>
<td></td>
</tr>
</tbody>
</table>

Small sized images can play real time
Medium sized images are within a factor of 2
CIF format (352x288) in real time on Phillips 50 MHz
PRPA VLIW processor

**The Global Village**

Internet distribution has been very successful
First release in November 1991
Many contributions by others: bug fixes, feature extensions, and performance improvements
Over 1500 copies distributed per month (7/93)
Over 8000 mpeg movies distributed per month (7/93)

Special acknowledgements...
- Toshihiko Kawai of Sony
- Tom Lane of the Independent JPEG Group
- Reid Judd of Sun Microsystems
- Todd Brunhoff of NVR
- Earl Killian of IDT, Inc.
- Chad Fogg of U. of Washington
- Paulo Villegas of Telefonica I+D
- Arian Koster of PTT
Future Plans

Integrate MPEG video stream into CMPlayer
   Full-function VCR commands with frame drops
   caused by decoder CPU and network limitations
Distribute portable, parallel MPEG video encoder
Experiment with other compression algorithms
   MPEG-2, wavelets, 3D subband coding, ...
Complete Berkeley Video-on-Demand server, meta
database, and archive server

Conclusions

MPEG-1 decoding is not that difficult
   ... within a factor of 2 for CIF images
   ... low cost chips/boards will be released real soon
Implementation experience
   Biggest problem is memory bandwidth, not CPU
   IDCT is only 15% of time
   Dithering and reconstruction over 50% of time
Playing movies on your screen is great fun, try it!