Development of a Transcoding Algorithm from MPEG to H.263

Nick Feamster
Development of a Transcoding Algorithm from MPEG to H.263

Motivation:

MPEG
- High bitrate
- DVD, Digital Television
- 1.5 Mbps, 30 fps

Transcode

H.263
- Low bitrate
- Wireless, etc.
- 500 kbps, 10 fps
Transcoding Issues

MPEG vs. H.263

- I, P, B Frames
- Restricted Motion Vectors on boundaries
- More I macroblocks per frame
- I, P Frames
- Unrestricted Motion Vectors on boundaries
- Fewer I macroblocks per frame
Computation vs. Quality

**Goal:** A fast transcoding algorithm which preserves picture quality.

**Method:** Make use of available decoded information to avoid unnecessary computation.
MPEG vs. H.263

- Y Component for MPEG- PSNR vs. Frame
- Y Component for H.263- PSNR vs. Frame
MPEG vs. H.263

Conclusion: 1.5 Mbps MPEG Source minimizes loss due to recoding.
Initial Method

- Drop B Frames
- MPEG Decode
- H.263 Motion Estimation
- H.263 Encode

- 9-10 Seconds per frame
- Too much computation for real time transcoding
Alternative: Reuse Information

What is the best way to do this?
Issue: MPEG I Frames

Problem: Each MPEG GOP has an I frame with no motion vectors.

Solution: Copy motion vectors from preceding MPEG P frame.
**Issue:** Small MV Errors

**Problem:** Quantization losses result in motion estimation discrepancies

**Solution:** Half pel search to refine motion vectors
Issue: MPEG Intra Macroblocks

- Often more useful to code an H.263 macroblock as an intra block, even if the block is coded inter in MPEG

Solution: *Use the minimum error threshold specified by the H.263 encoder to correct modes.*
Resulting Method

1. Drop B frames → MPEG Decode
2. I frame? → MV’s from P frame
   - Yes: Half Pel Correct
   - No: MV’s from current frame → Mode Correct → H.263 Encode
Other Considerations

• Boundary Error
  – 0.5 dB loss, worst case (sequence dependent)
  – Much of this loss can be recovered with mode correction and half pel search
  – At a lower spatial resolution, boundary motion vectors take on increasing importance.
H.263 Boundary Motion Vectors
Other Considerations

• MPEG I Frames
  – Tried weighted average of neighboring P frame MV’s based on overlap, not as effective as simple replacement.
  – Replacement works well on sequences with correlated motion vectors
  – High quality P frame from MPEG I frame → Less degradation of subsequent P frame quality
Results

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Method 1 (sec)</th>
<th>Method 2 (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carousel (30 IP)</td>
<td>323</td>
<td>26</td>
</tr>
<tr>
<td>Bus (30 IP)</td>
<td>254</td>
<td>26</td>
</tr>
<tr>
<td>Girl (18 IP)</td>
<td>106</td>
<td>15</td>
</tr>
<tr>
<td>Bus (10 IP)</td>
<td>80</td>
<td>9</td>
</tr>
<tr>
<td>Carousel (10 IP)</td>
<td>90</td>
<td>8</td>
</tr>
<tr>
<td>Football (10 IP)</td>
<td>92</td>
<td>8</td>
</tr>
</tbody>
</table>

~ 10x Efficiency Improvement
Results

**Bus**

PSNR vs. I/P Frame for Various H.263 Streams

**Carousel**

PSNR vs. I/P Frame for Various H.263 Streams

**Girl**

PSNR vs. I/P Frame for Various H.263 Streams
Conclusions

• Reusing motion vector information from decoded MPEG stream saves 85-90% of computation time with an average of 1.4 dB loss.
• Much of the loss can be attributed to motion vector errors resulting from requantization of the original frame pixel values.
Future Work

- Algorithmic research (motion vector search for I frames, etc.)
- Real-time implementation
- Allow MPEG interlace as transcoder input