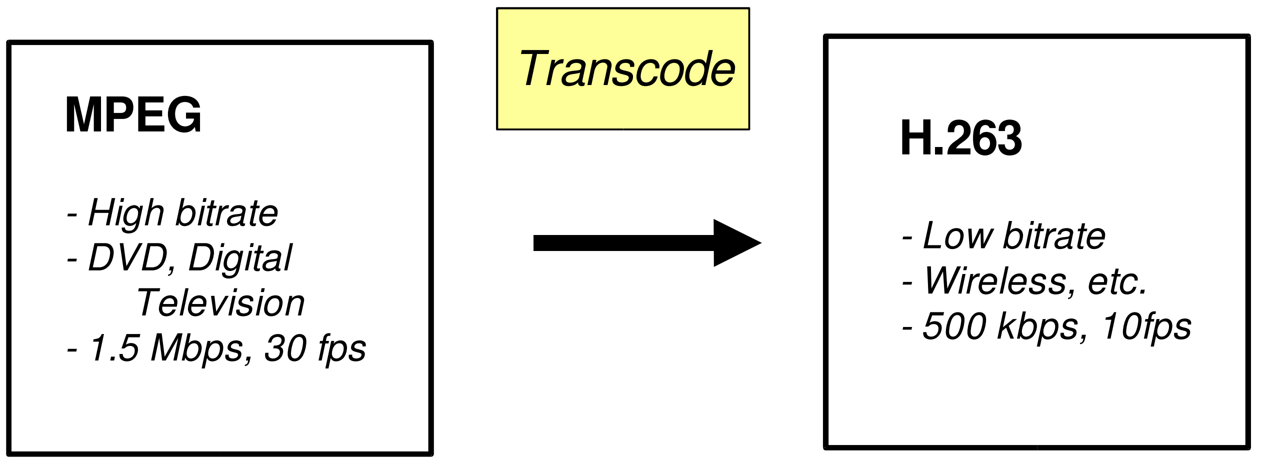
Development of a Transcoding Algorithm from MPEG to H.263

Nick Feamster

Development of a Transcoding Algorithm from MPEG to H.263

Motivation:



Transcoding Issues

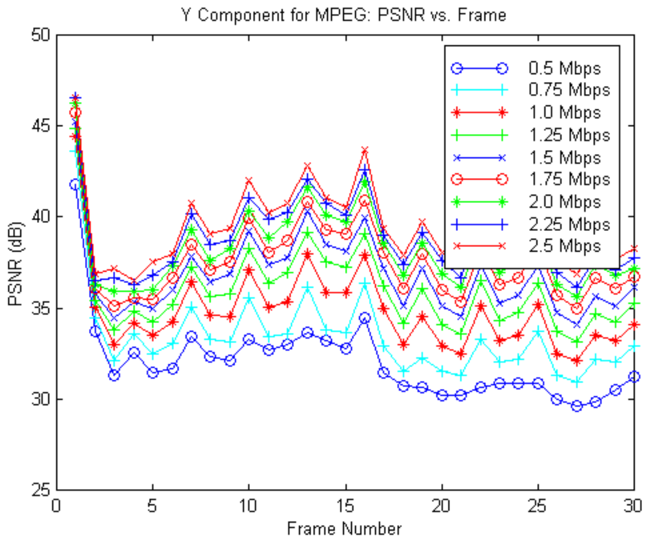
MPEG	*vs.*	**H.263**
- I, P, B Frames		- I, P Frames
- Restricted Motion Vectors on boundaries		- Unrestricted Motion Vectors on boundaries
- More I macroblocks per frame		- Fewer I macroblocks per frame

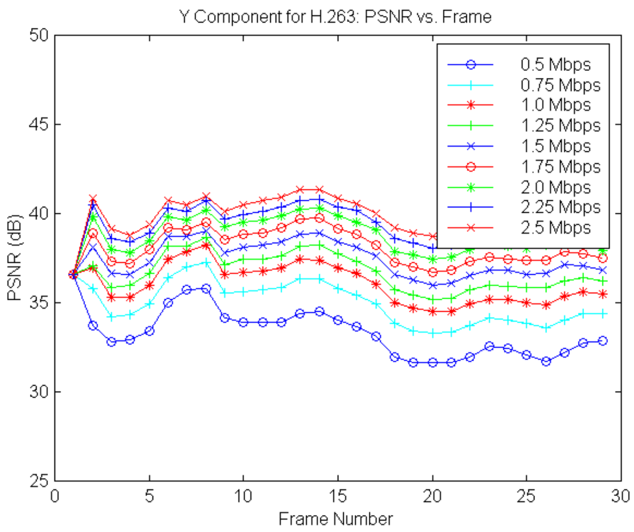
Computation vs. Quality

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

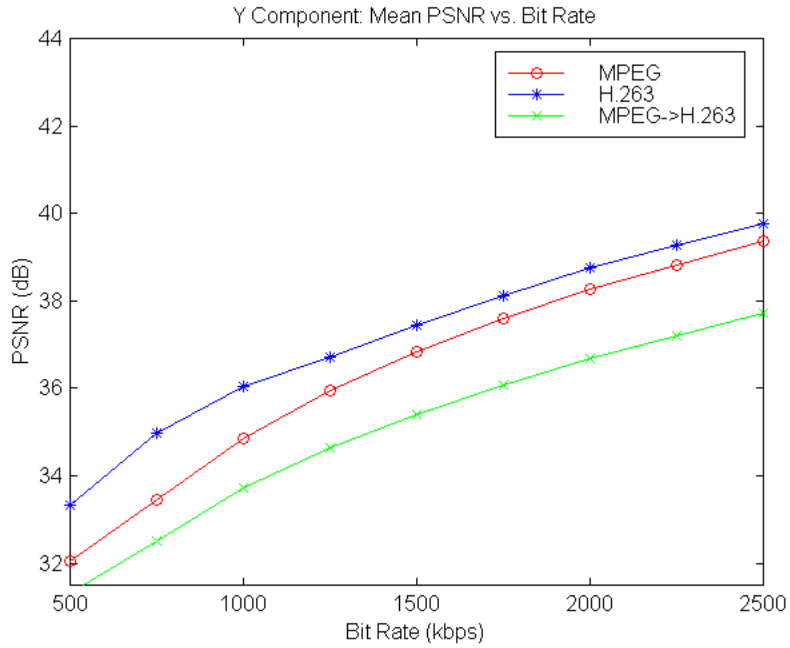
Method: *Make use of available decoded information to avoid unnecessary computation.*

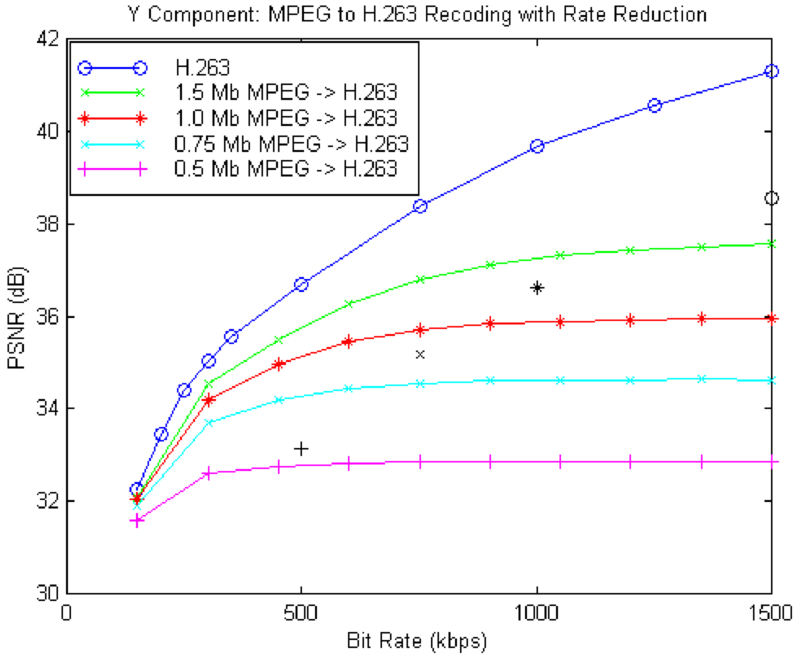
MPEG vs. H.263


Y Component for MPEG: PSNR vs. Frame
PSNR (dB)
Frame Number
0.5 Mbps
0.75 Mbps
1.0 Mbps
1.25 Mbps
1.5 Mbps
1.75 Mbps
2.0 Mbps
2.25 Mbps
2.5 Mbps


Y Component for H.263: PSNR vs. Frame
PSNR (dB)
Frame Number
0.5 Mbps
0.75 Mbps
1.0 Mbps
1.25 Mbps
1.5 Mbps
1.75 Mbps
2.0 Mbps
2.25 Mbps
2.5 Mbps

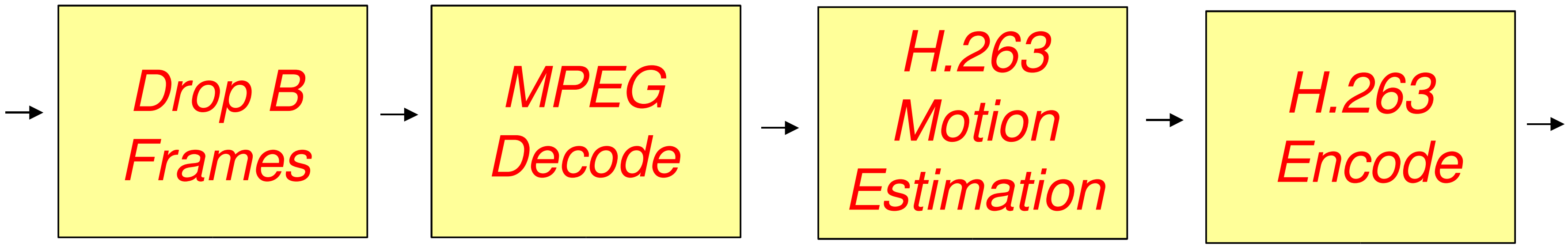
MPEG vs. H.263





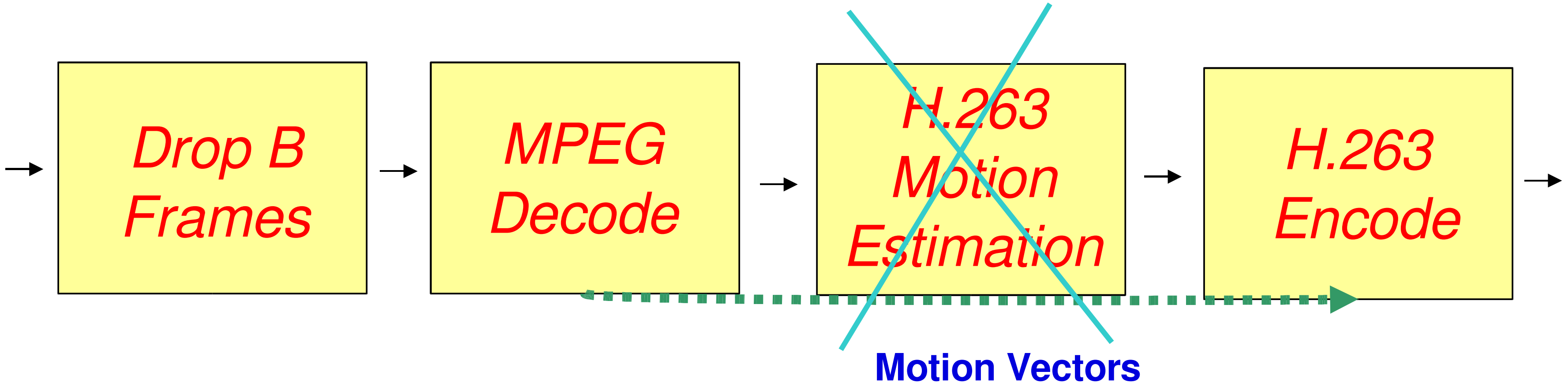
Conclusion: *1.5 Mbps MPEG Source minimizes loss due to recoding.*

Initial Method



- *9-10 Seconds per frame*
- *Too much computation for real time transcoding*

Alternative: Reuse Information

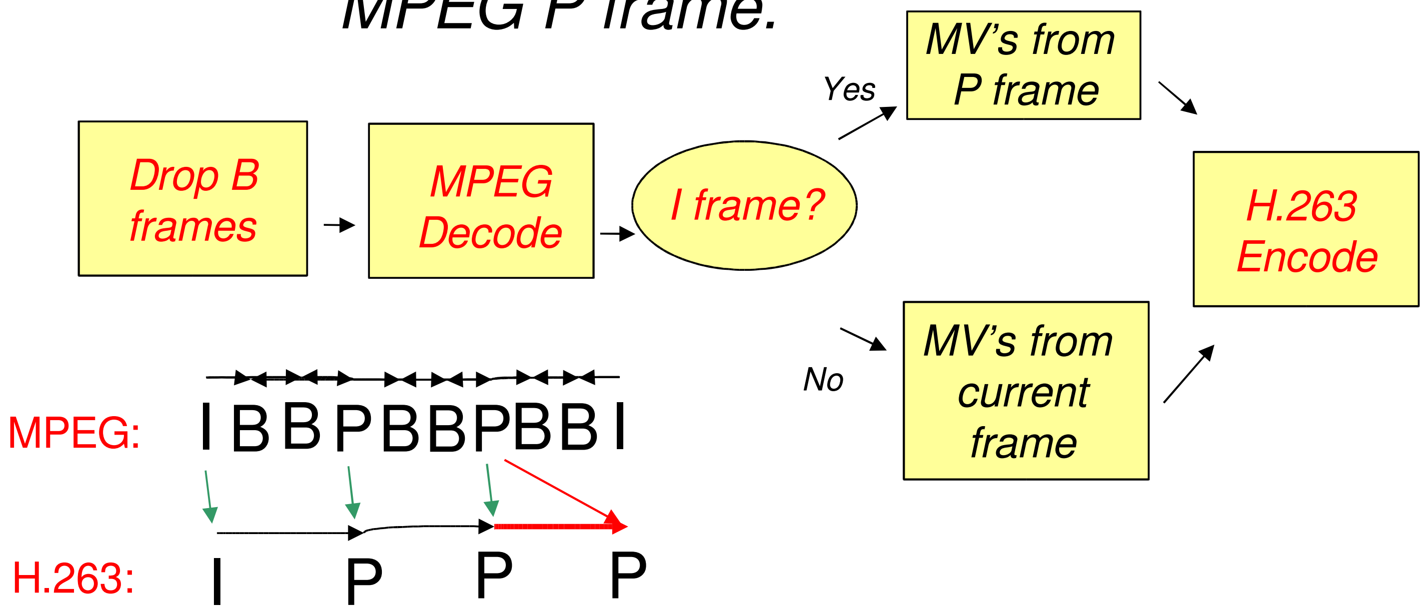


What is the best way to do this?

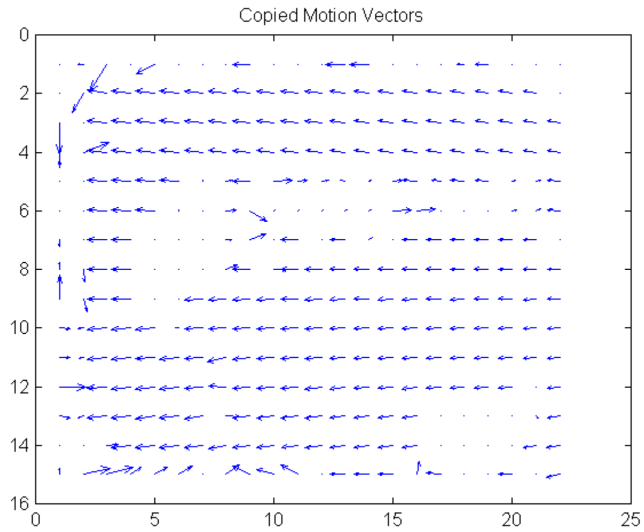
Issue: MPEG I Frames

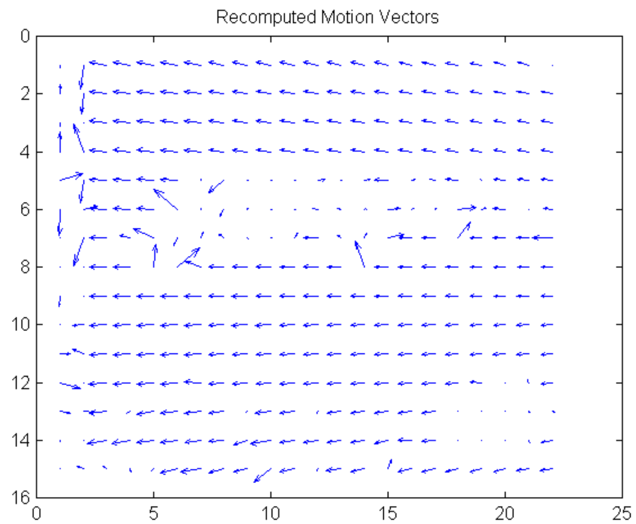
Problem: *Each MPEG GOP has an I frame with no motion vectors.*

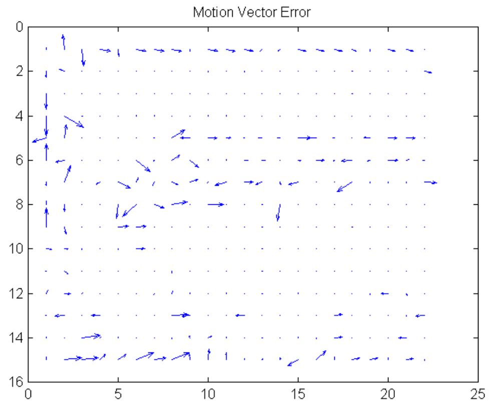
Solution: *Copy motion vectors from preceding MPEG P frame.*



Issue: Small MV Errors







Problem: *Quantization losses result in motion estimation discrepancies*

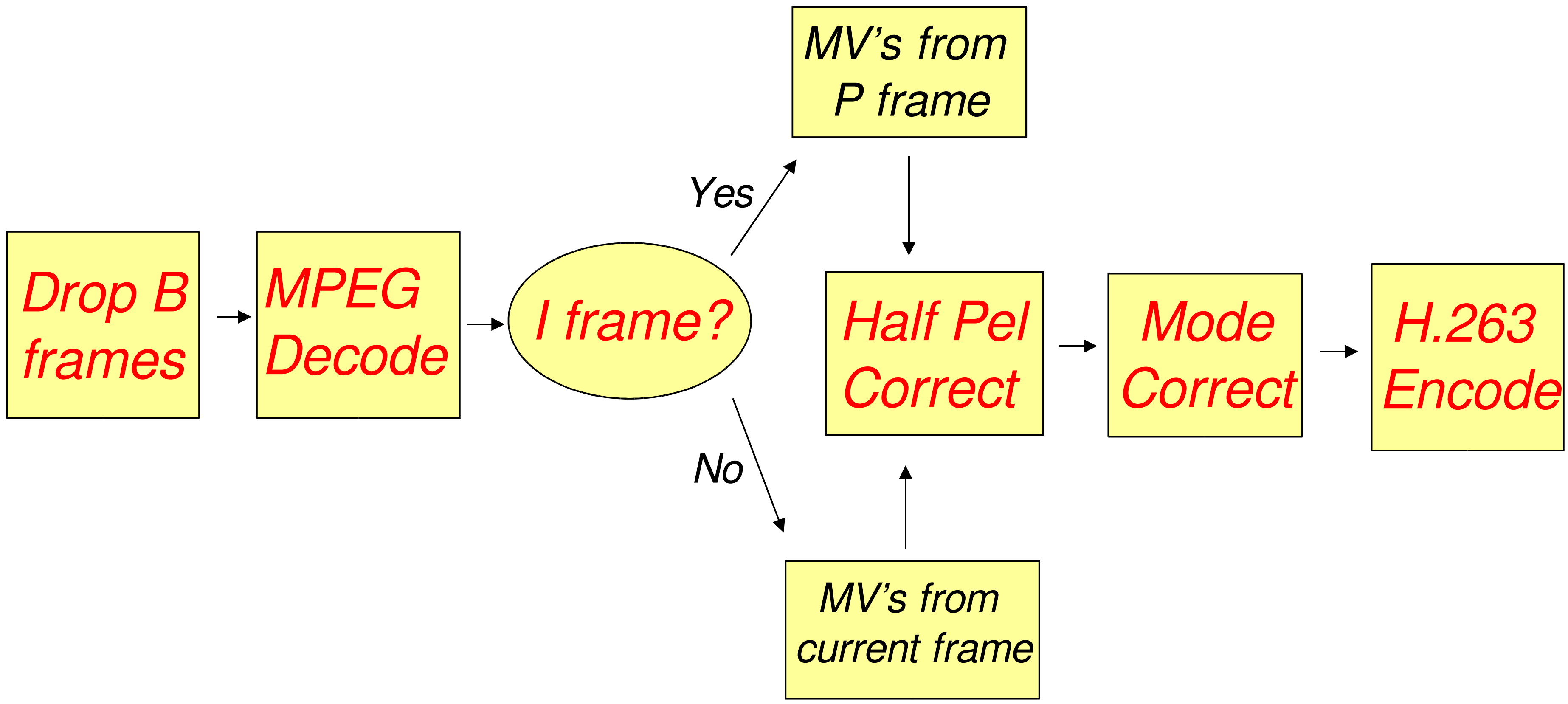
Solution: *Half pel search to refine motion vectors*

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

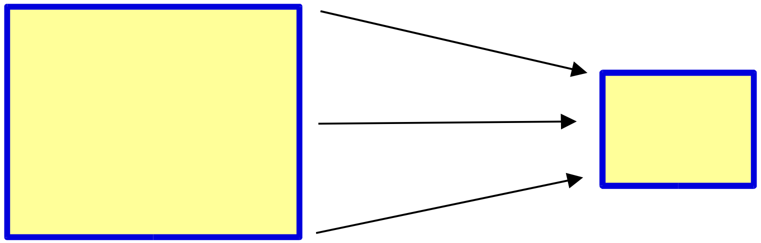
Solution: *Use the minimum error threshold specified by the H.263 encoder to correct modes.*

Resulting Method

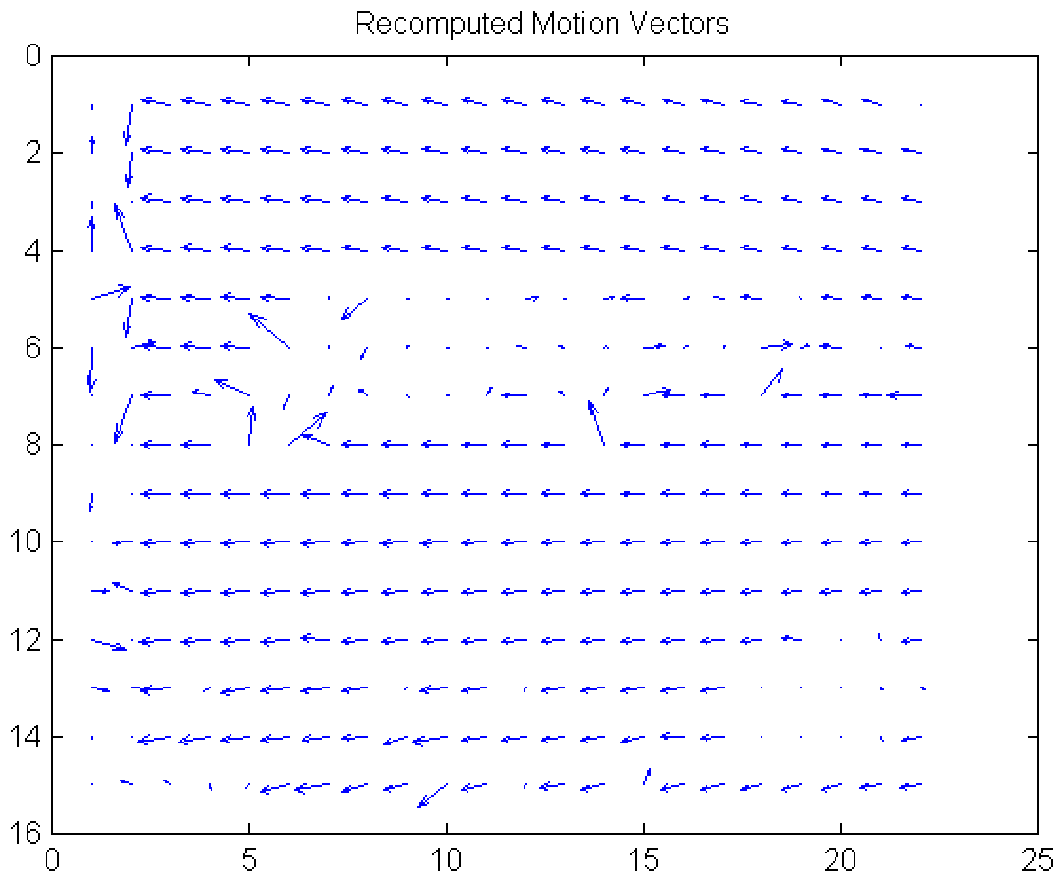


Other Considerations

- 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 $\rightarrow$ Less degradation of subsequent P frame quality

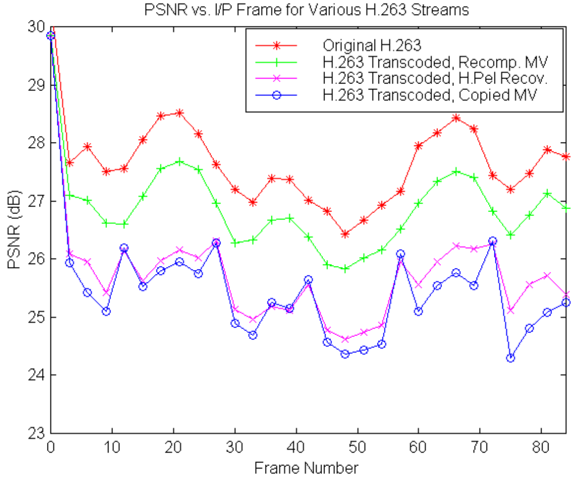
Results

Sequence	Method 1 (sec)	Method 2 (sec)
Carousel (30 IP)	323	26
Bus (30 IP)	254	26
Girl (18 IP)	106	15
Bus (10 IP)	80	9
Carousel (10 IP)	90	8
Football (10 IP)	92	8

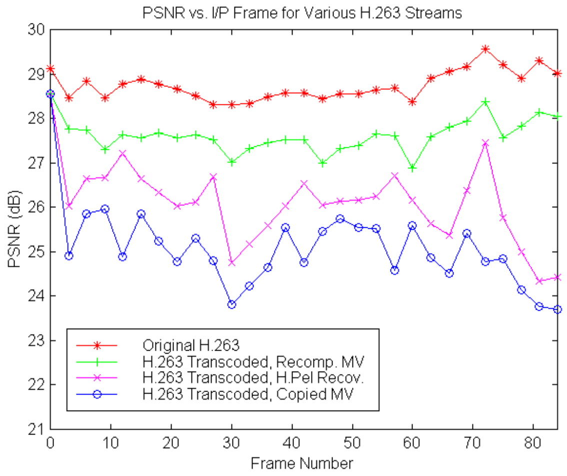
~ 10x Efficiency Improvement

Results

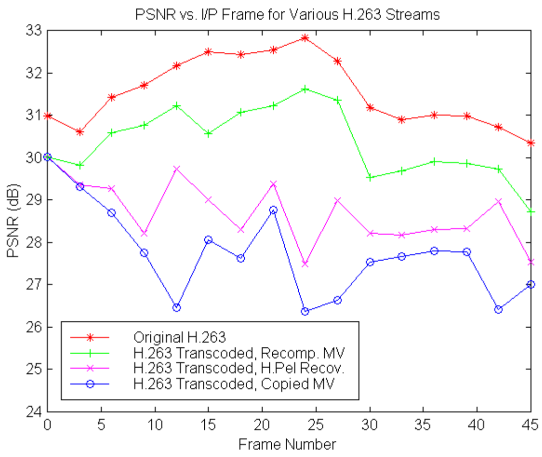
Carousel



Bus

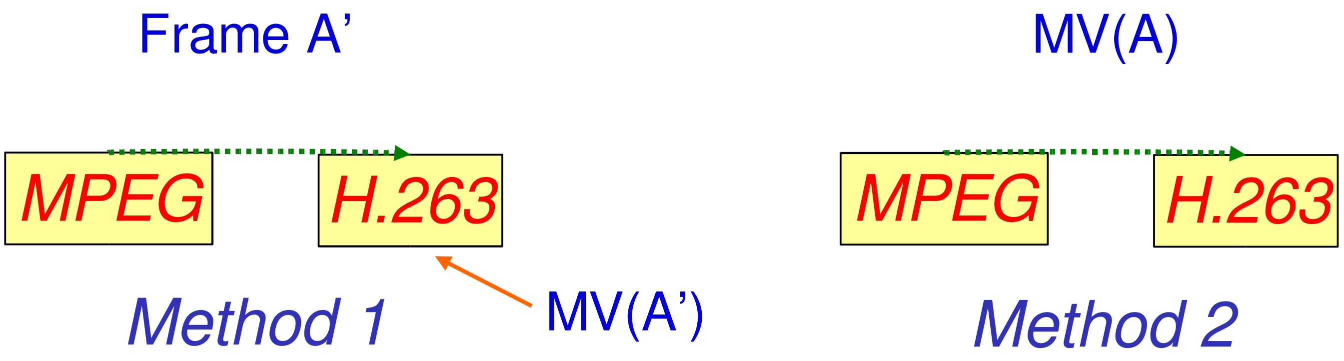


Girl



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