Video Transcoding Architectures

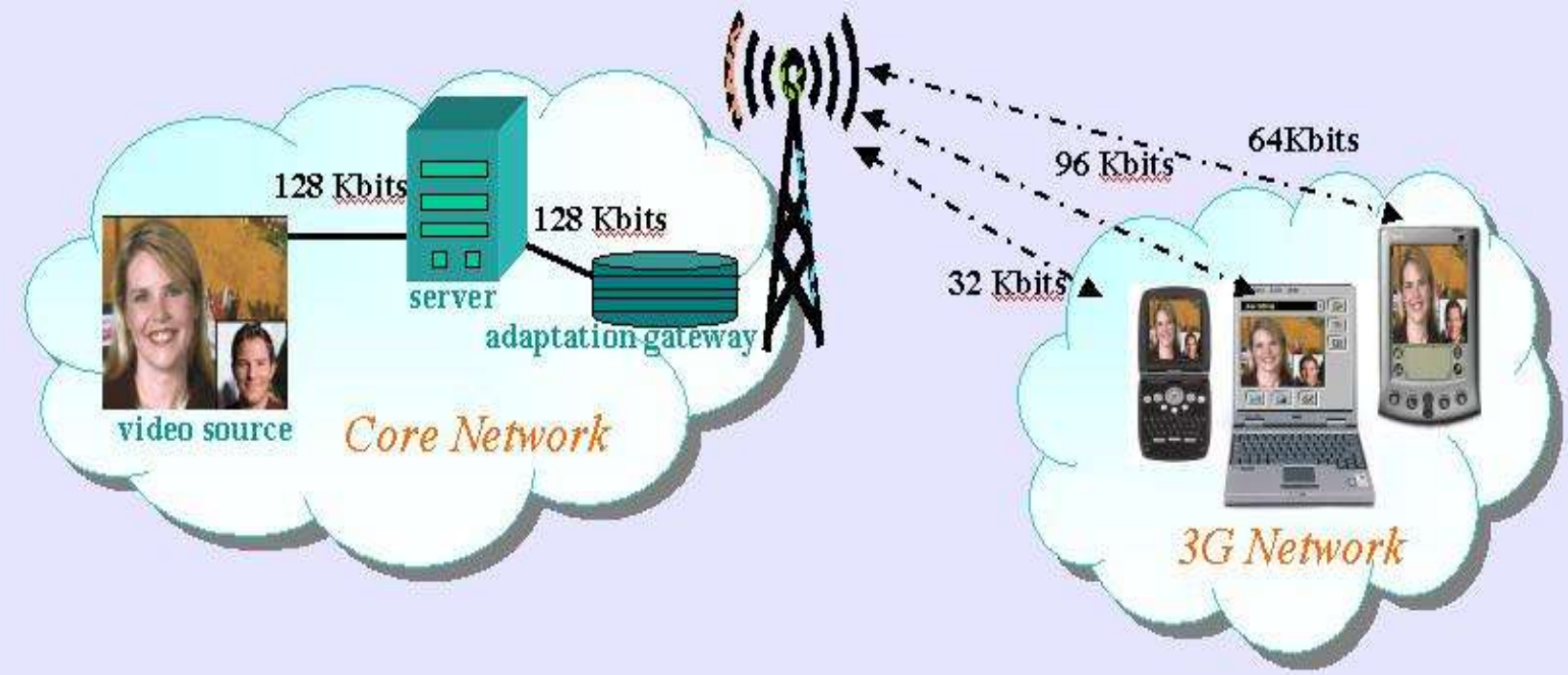
Project Group

- Prof. Maurizio A. Bonuccelli
- Francesca Martelli
- Francesca Lonetti

Former Collaborators

- Luigi D'Amaro
- Gianni Rosa
- Luca Leonardi

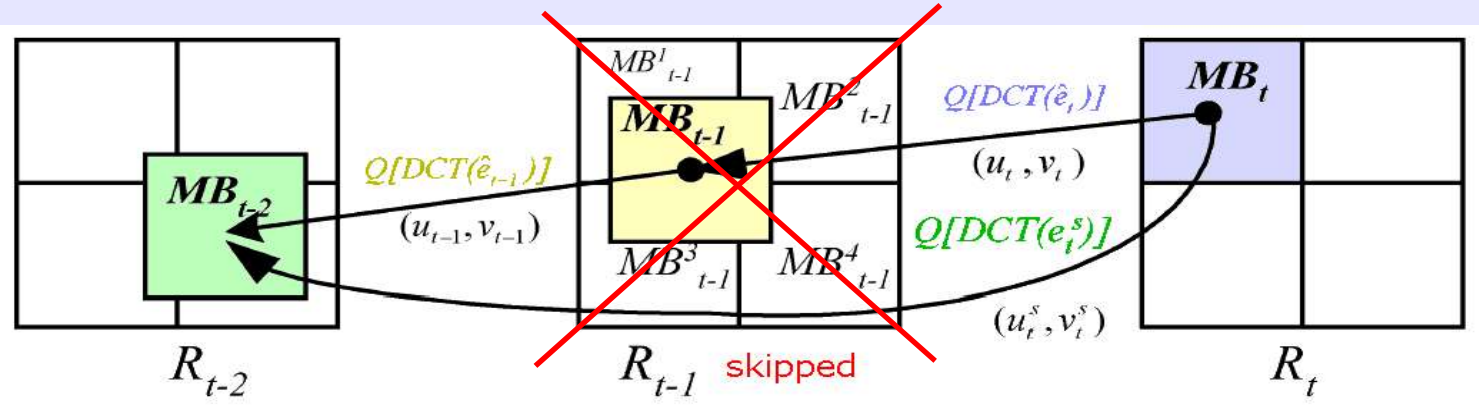
Project Context



Video Transcoding

- Format conversion
- Bit rate conversion (quality transcoding)
- Size conversion (spatial transcoding)
- Frame rate conversion (temporal transcoding)

Temporal Transcoding



- To choose the frames to be skipped (frame skipping policy)
- To recompute the motion vectors not still valid:
  - Motion Estimation
  - Motion Vector Composition
  - Motion Vector Composition+Refined Search
- To recompute the prediction errors

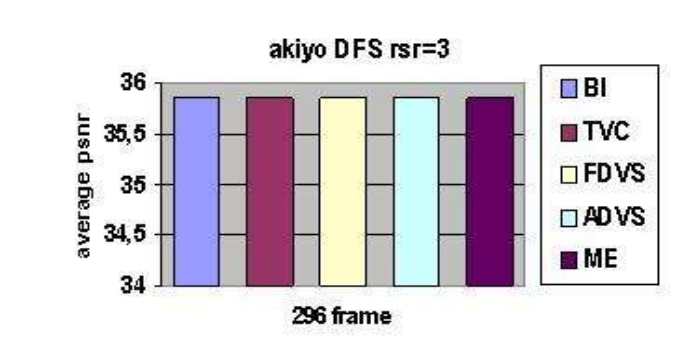
Past Work (1)

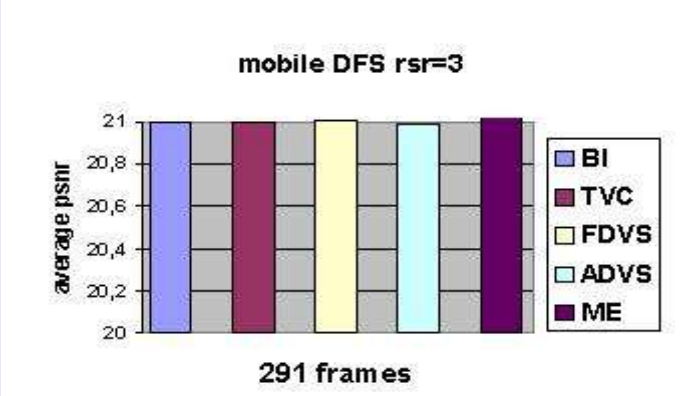
MPEG4 Temporal Transcoder:

- 4 Motion Vector Composition Algorithms:
  - BI; TVC; FDVS; ADVS
- 2 Architectures
  - DFS
    - Prediction Errors re-encoding: Standard Way
    - Frame Skipping: Motion Activity
  - FSC
    - Prediction Errors re-encoding: Fast Way
    - Frame Skipping: Motion Activity + Prediction Errors

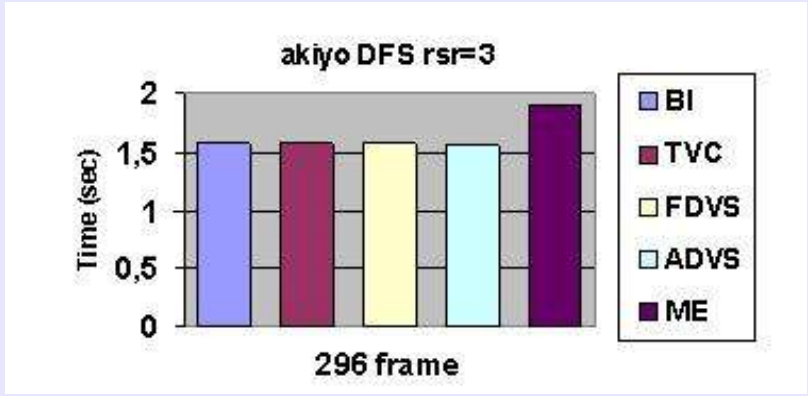
Lesson Learned: MVC

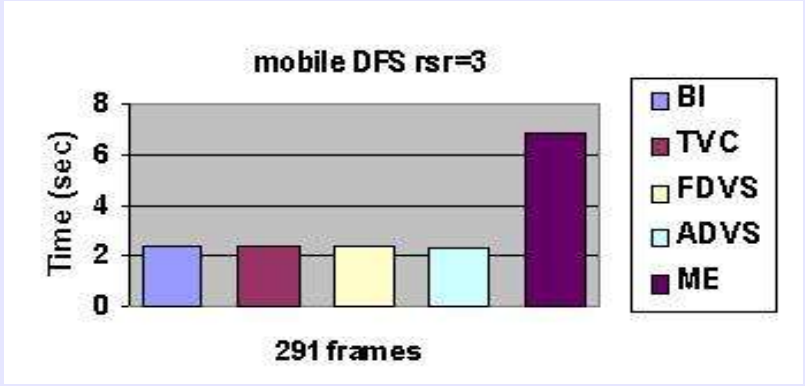
PSNR



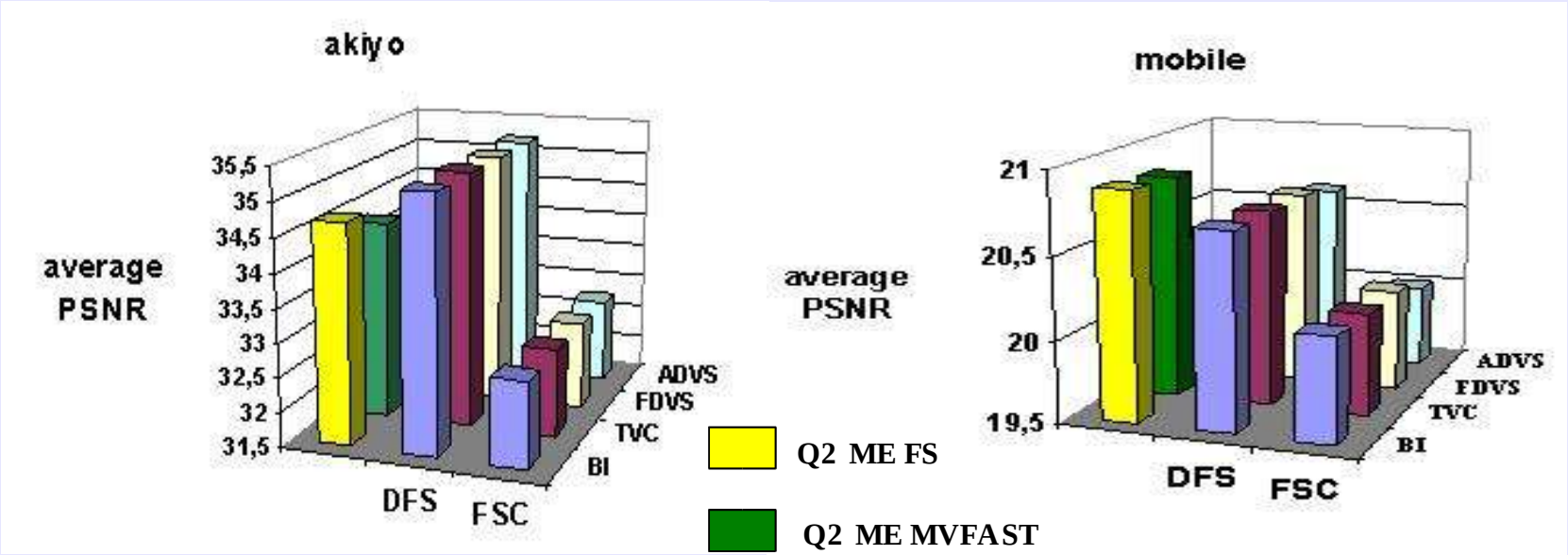


TIME





Lesson Learned: Transcoding Architectures



DFS achieves better performance

Past Work (2)

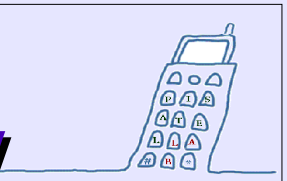
H.263 Temporal Transcoder:

- 4 Motion Vector Composition Algorithms:
  - BI; TVC; FDVS; ADVS
- Architecture: DFS
- Frame Skipping Policies:
  - Buffer
  - Motion Activity
  - Weighted Random
  - Energy Measure
  - Logarithmic Prediction
  - Consecutive Skipping

Basic Skipping Policy

- Our context is a real-time one: max admitted delay = 500 ms ⇒buffer size = ½ output bit rate
- For achieving it, buffer overflow is to be avoided
- Then BUFFER strategy is ALWAYS adopted:
  - UNDERFLOW: if Occupancy<20% then transcode
  - OVERFLOW: if Occupancy>80% then skip

Motion Activity Policy



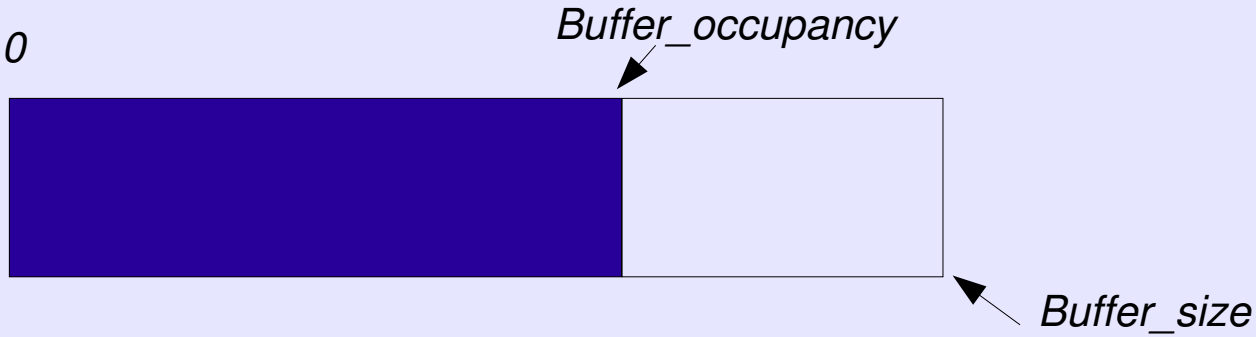
- A frame is skipped if it does not have considerable motion
- Frame motion is quantified as the sum of motion vectors:

$$MA = \sum_i mv_i$$

Weighted Random Policy

A random number *r* is generated, such that $r \in$ [0..buffer_size]:

- If *r<buffer_occupancy* then frame skipped
- If *r>buffer_occupancy* then frame transcoded



Energy Measure Policy

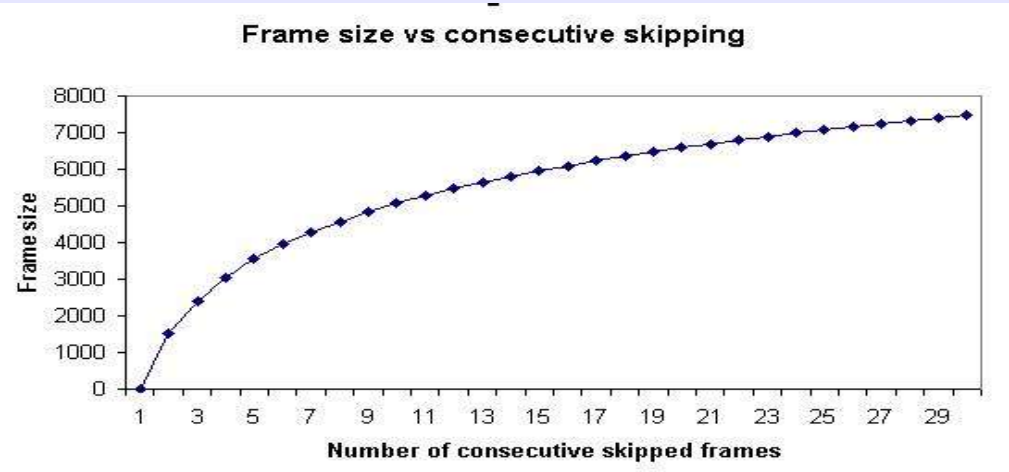
$$ave_{var} = 1/N \sum_{i=0}^{N-1} \sigma_i^2 = 1/(6x64xN) \sum_{i=0}^{N-1} \sum_{j=0}^{6x64} (p_j - p_a)^2$$

- If that value is greater than that one of the previous frame then frame is transcoded, else it is skipped

Consecutive Skipping Policy

- We define Γ = input bit-rate/output bit_rate
- If the number of consecutive skipped frames is less than Γ, then the current frame is skipped

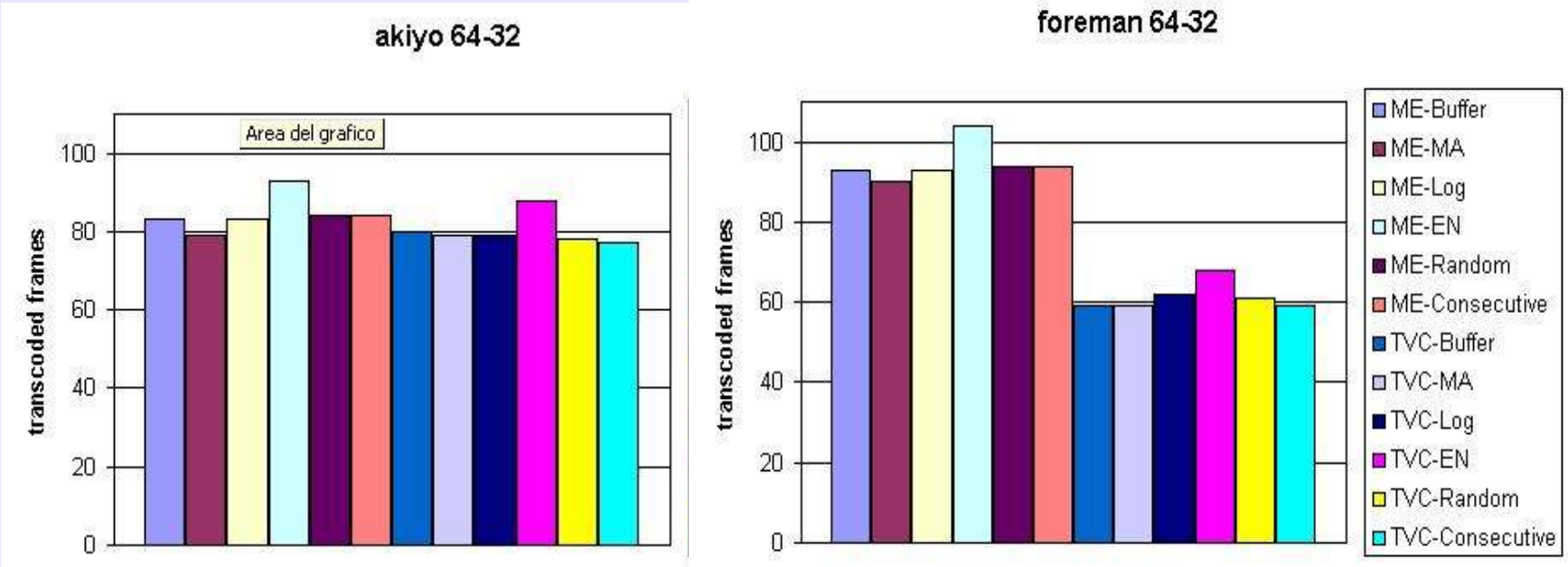
Logarithmic Prediction Policy



- If frame *t* is skipped, due to overflow condition, the policy predicts the next frame to be transcoded, say *t+k*, without transcoding the *t+1*,*t+2*,...,*t+k-1*

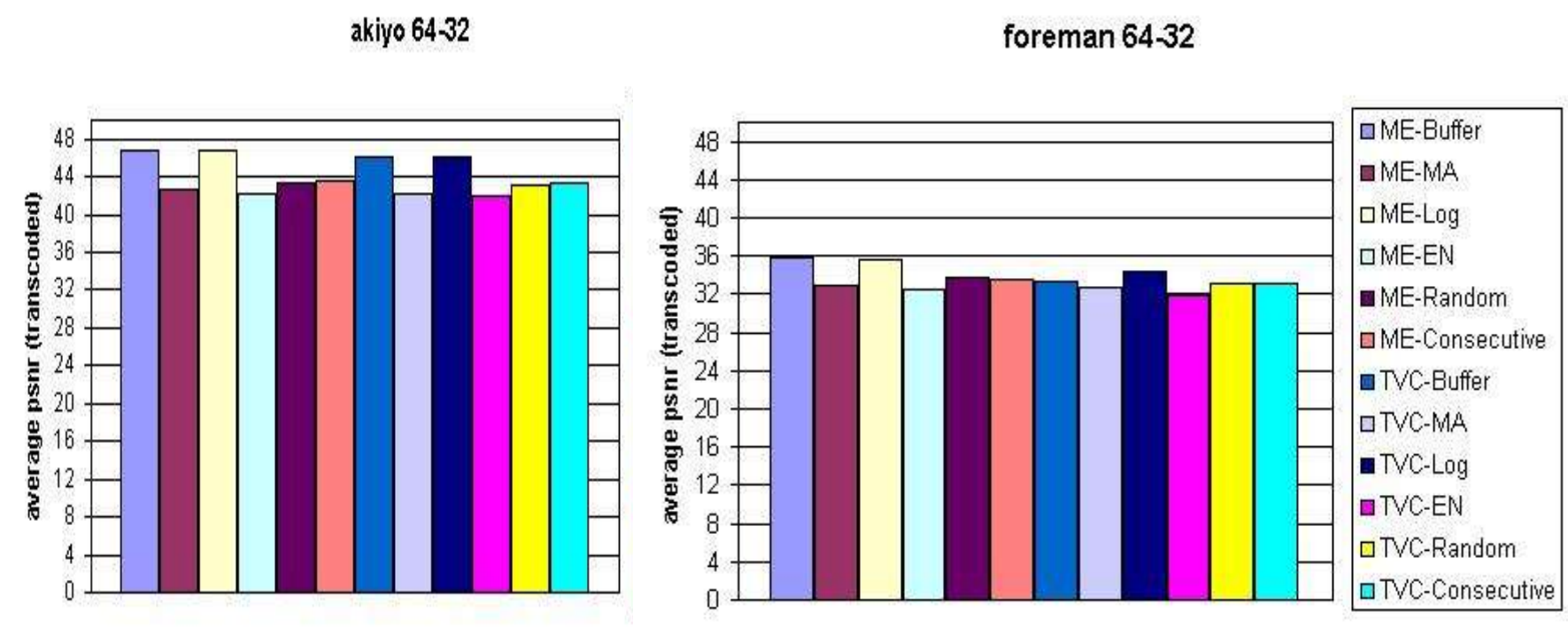
H.263 Transcoder Performance

NUMBER OF TRANSCODED FRAMES



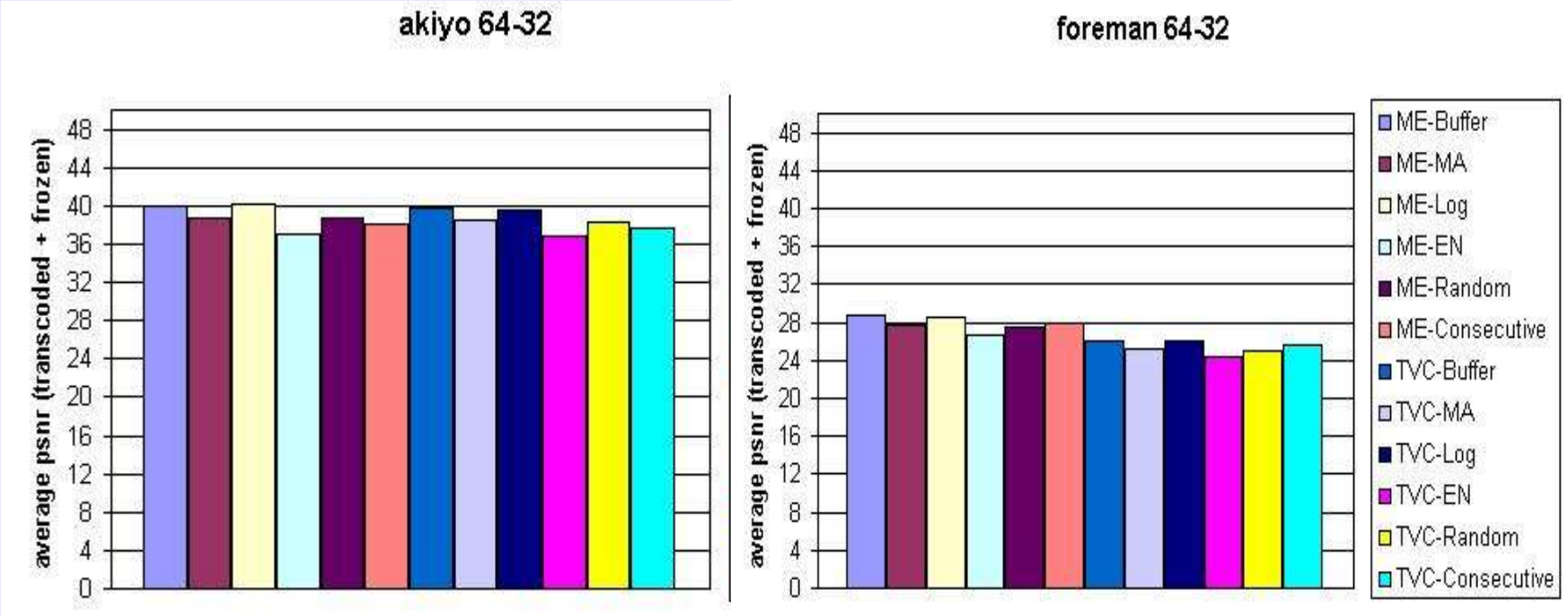
H.263 Transcoder Performance

AVERAGE PSNR (transcoded)



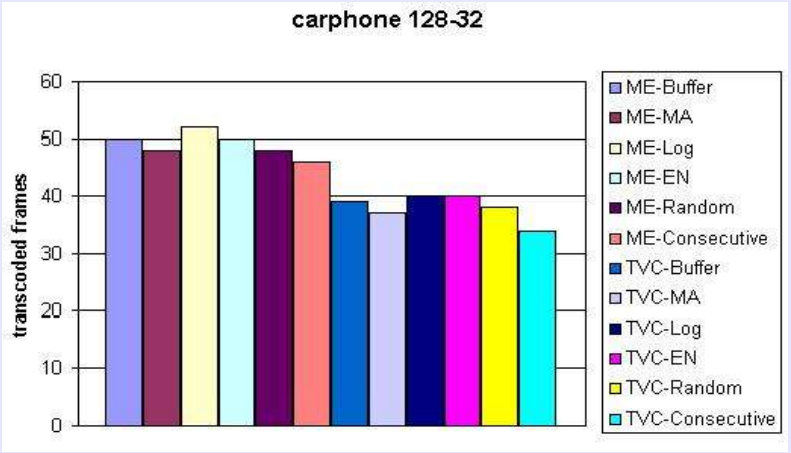
H.263 Transcoder Performance

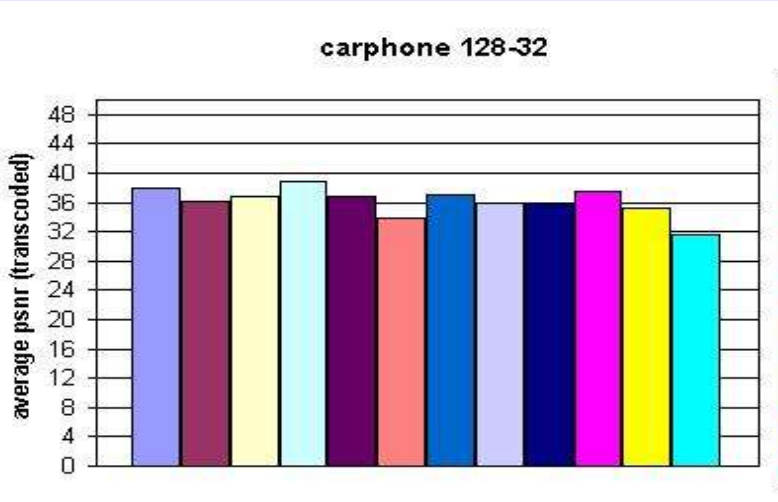
AVERAGE PSNR (transcoded+frozen)

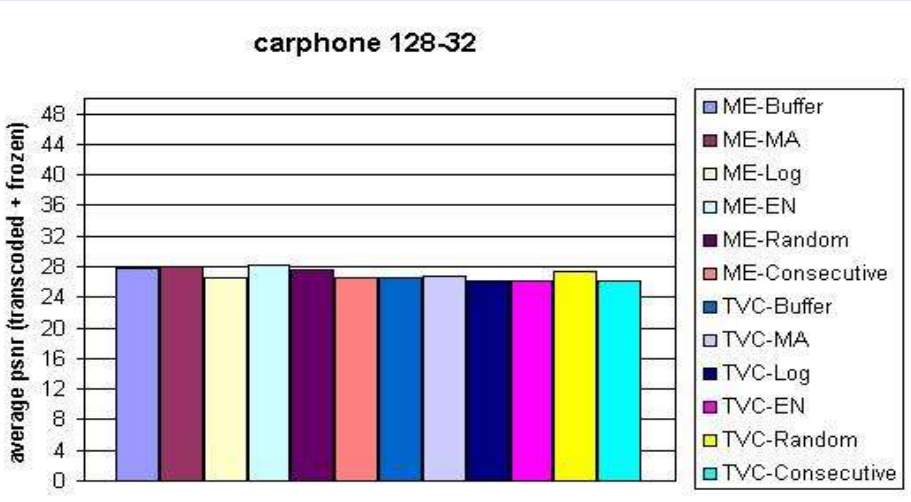


H.263 Transcoder Performance

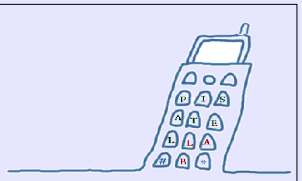
HARD TRANSCODING







Demo



- Two test cases:
  - Video sequence with little motion (*akiyo*)
  - Video sequence with a lot of motion and scene changes (*foreman*)
- For each one of them:
  - original video sequence
  - transcoded video sequence (128/64 Kbits)
  - transcoded video sequence (256/64 Kbits)

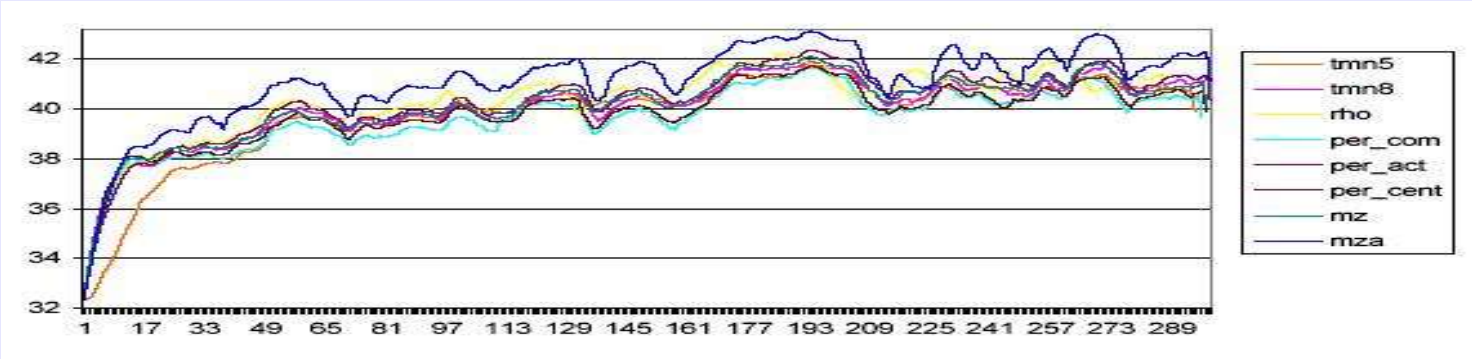
Lesson Learned: Frame Quality

- By observing the test results, we realize that video quality is influenced by the rate control of the front encoder
- Then, we tested and implemented several rate control algorithms for the front encoder:
  - TMN5
  - TMN8
  - ρ-domain
  - Perceptual rate control
  - Multiple zone (Activity)

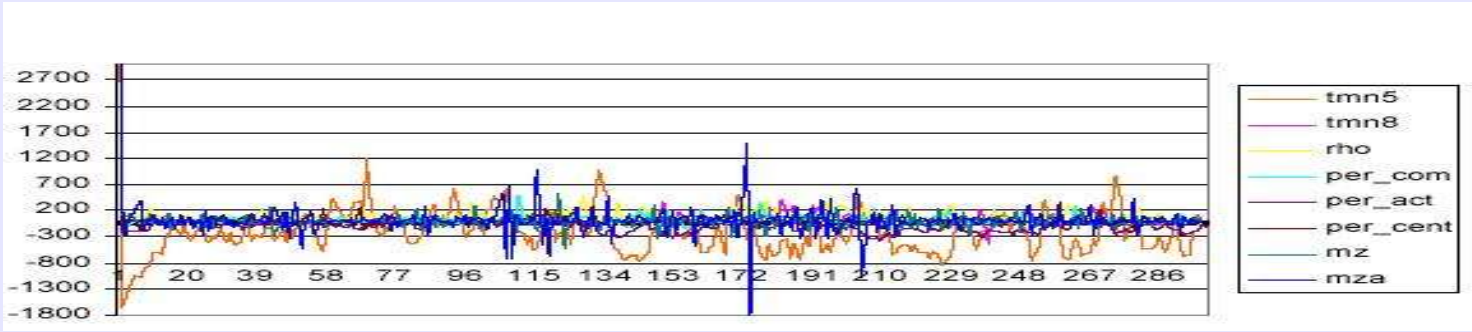
Rate control performance

Akiyo

PSNR



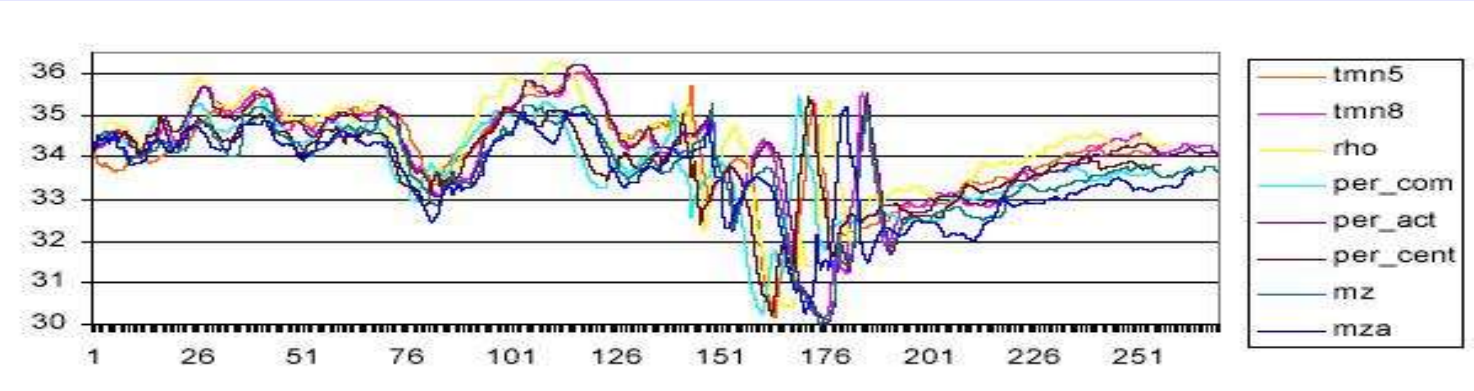
effective bits vs average bits



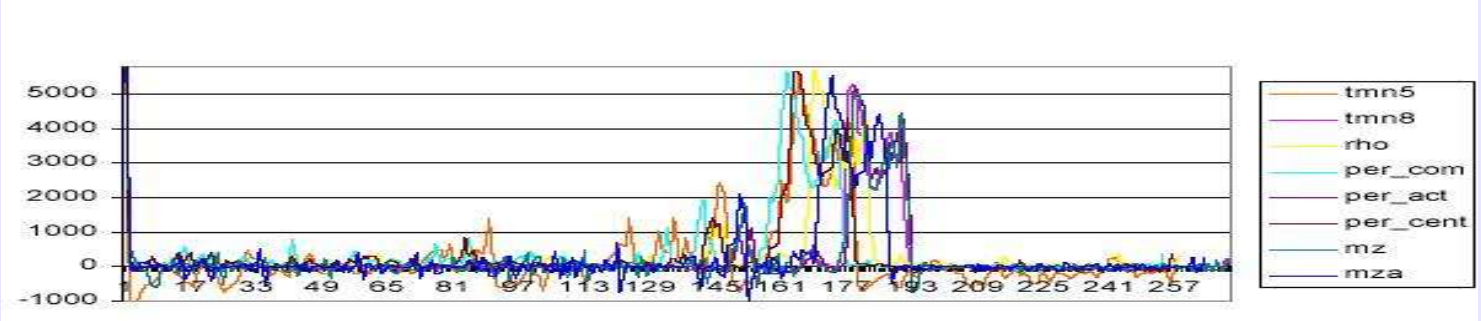
Rate control performance (2)

Foreman

PSNR



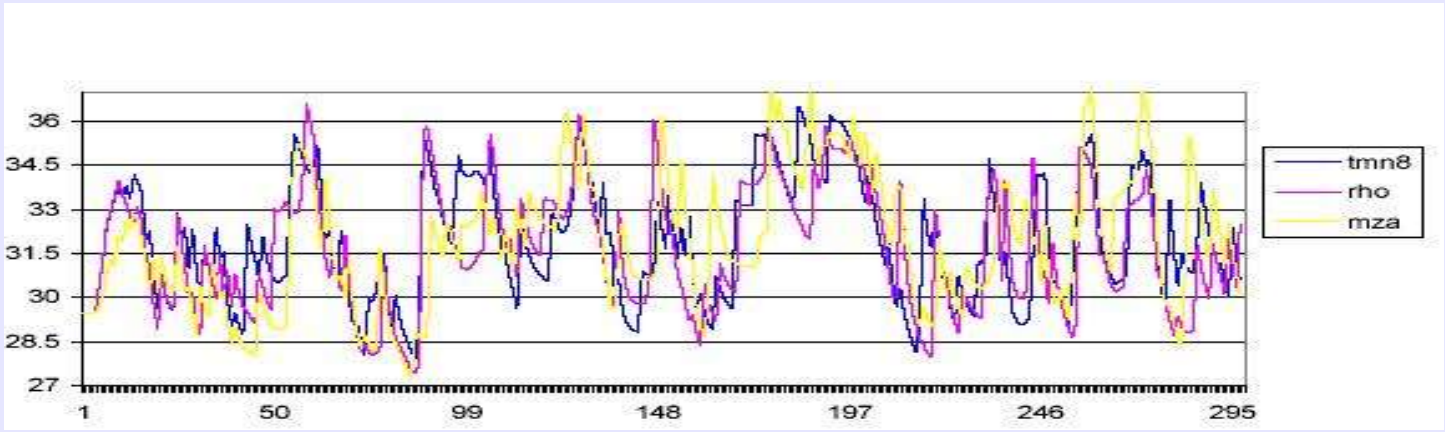
effective bits vs average bits



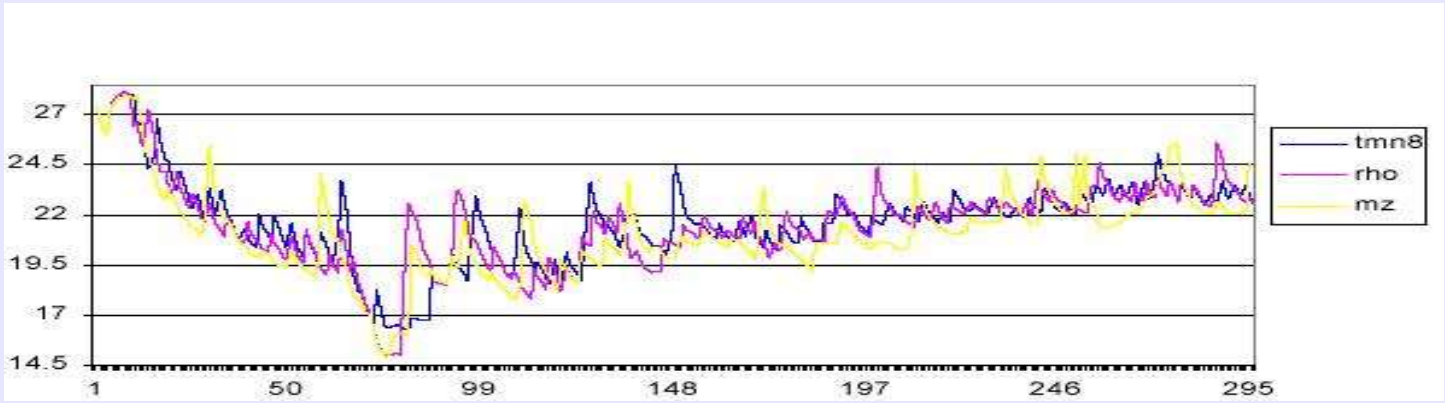
Transcoder performance

PSNR

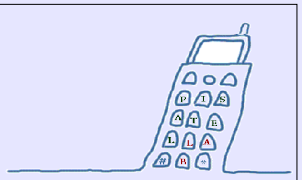
Akiyo 64-32



Costguard 64-32



Purely Temporal Transcoding: Problems



In hard transcoding conditions (high input/output bit rate ratio), a very critical situation can occur:

Frame size is greater than the free buffer space

- IF that frame is transmitted ⇒buffer overflow
- IF that frame is skipped ⇒buffer occupancy decreases, but the size of the next frame is larger ⇒ the same situation can occur⇒ irreversible buffer underflow

Problem Solution

- A purely temporal transcoder can not decrease the frame size by increasing the quantization parameters (Quality Transcoding)
- We can face the critical case only by decreasing the frame size

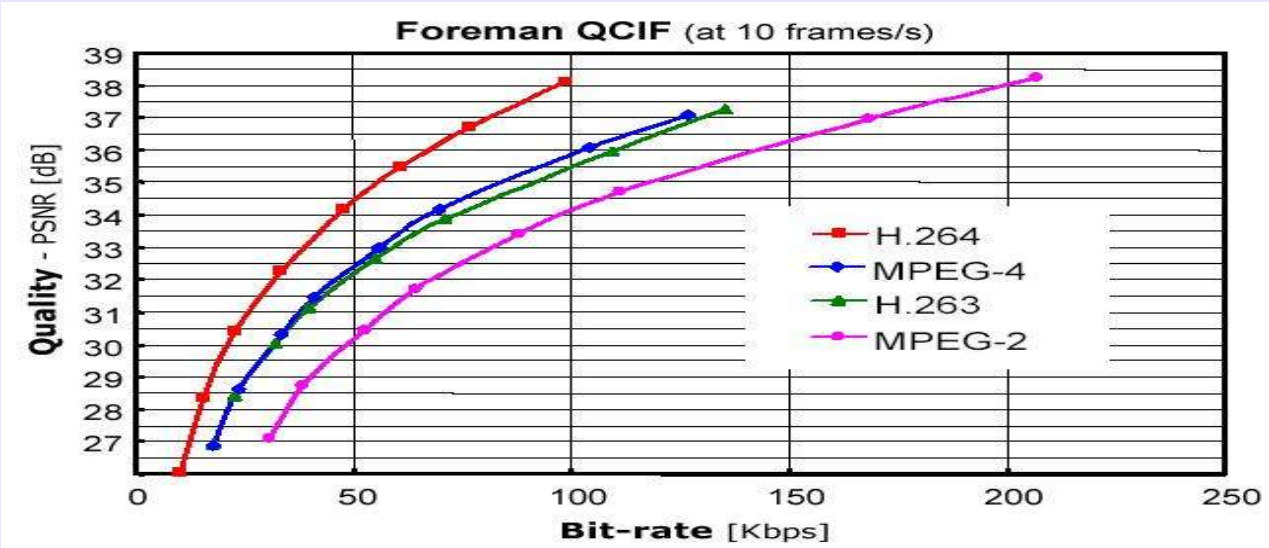
  $\Rightarrow$ Hybrid Transcoder: **Temporal + Quality**
- To avoid long computation time, a simple rate control algorithm can be adopted when the buffer occupancy level is close to the critical case
- Forcing intra frame

Current Work: H.264

- Many advanced compression techniques are adopted:
  - Advanced Intra-Prediction
  - Strong Motion Isolation (4×4, ¼-pel resolution)
  - Multiple Reference Frames
  - Context-adaptive VLC

⇒ *Very high complexity!*

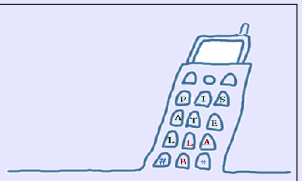
- Average bit rate reduction of 50%!



Current & Future Work: H.264

- H.264 encoder setting and optimization
- H.264 transcoder implementation
- Rate control algorithms for H.264

Future Work



- MPEG4 – H.263 transcoders refined comparison
  - Investigate why size of MPEG4 transcoded frames is smaller than H.263 ones
- H.263 hybrid transcoder implementation
- Buffer sizing vs transmission delay
- New skipping policies
- New motion vector composition algorithms