Block-Based Hybrid Video Coding Using Motion-Compensated Long-Term Memory Prediction

Thomas Wiegand, Xiaozheng Zhang, and Bernd Girod

ABSTRACT: *Our new approach extends the spatial displacement vector utilized in block-based hybrid video coding by a variable time delay permitting the use of more frames than the previously decoded one for motion compensation. The long-term memory covers the decoded frames of some seconds at the encoder's as well as the decoder's side. This scheme is well suited in cases of repetition in the sequence, e.g. a head is rotating back into its old position, or if the camera is shaking. However, transmission of the variable time delay requires additional bit-rate which may be prohibitive when the size of the long-term memory increases. Therefore, we control the bit-rate of the motion information by employing rate-constrained motion estimation. Simulation results are obtained by integrating long-term memory prediction into an H.263 codec. PSNR improvements up to 2 dB for the Foreman sequence and 1.5 dB for the Mother-Daughter sequence are demonstrated in comparison to the TMN-2.0 H.263 coder.*

1. INTRODUCTION

In recent years, numerous standards such as H.261, H.263, MPEG-1, and MPEG-2 have been introduced for compression of video data for digital storage and communication services. Together, the applications for these standards span the gamut from low bit-rate video telephony to high quality HDTV with the most recent standard H.263 [7], targeting the low bit-rate end. H.263 as well as the other standards utilize hybrid video coding schemes which consist of motion-compensated prediction (MCP) and DCT-based transform quantization of the prediction error. It is highly likely that also the future MPEG-4 standard will follow the same approach.

In most cases, the motion compensation (MC) is carried out by employing the immediately preceding frame which is available as a reconstructed frame at the encoder and the decoder. Long-term statistical dependencies in the coded video sequence are not exploited in existing international standards. An overview of techniques in video which use multiple reference frames can be found in [1].

We can beneficially view MCP a source coding problem with a fidelity criterion. For a certain bit-rate MCP provides a version of the video signal with a certain distortion. The bit-rate/distortion trade-off can be controlled by various means. One approach is to treat MCP as special case of entropy-constrained vector quantization (ECVQ) [2, 3]. The image blocks to be encoded are quantized using their own code books that consist of image blocks of the same size in the previously decoded frames. A code book entry is addressed by the translational motion parameters which are entropy-coded. Hence, the criterion for the block motion search is the minimization of the Lagrangian cost function wherein a distortion measure is weighted against a rate measure using a Lagrange multiplier. The Lagrange multiplier imposes the rate-constraint as for ECVQ, and its value directly controls the rate distortion trade-off.

Another approach to control the rate distortion performance of MCP is to use variable block sizes or, in general, variable shaped segments of the image [4, 5]. However, in most cases the segmentation has to be transmitted as side information, which may be prohibitive for low bit-rates. Therefore, the accuracy of the segmentation has to weighted against the distortion gains by MCP [6]. H.263 decoders can only handle images segmented either into blocks of 16×16 or 8×8 pels. This very coarse structure of segmentation requires only a small amount of bit-rate to be transmitted and appears to be efficient.

In this work we utilize both tools: rate-constrained bit allocation and the H.263 segmentation architecture to control the rate distortion performance of long-term memory MCP.

2. MOTION-COMPENSATED LONG-TERM MEMORY PREDICTION

Our approach for exploiting long-term statistical dependencies is to extend the motion vector utilized in hybrid video coding by a variable time delay permitting the use of more frames than the previously decoded one for block-based motion compensation. With that, a long-term memory containing several seconds of the reconstructed image sequence can be used by the MCP. The frames inside the long-term memory which is simultaneously built at encoder and decoder are addressed by a combination of the codes for the motion vector and the variable time delay. Hence, the transmission of the variable time delay causes additional bit-rate that has to be justified by improved MCP. This trade-off limits the efficiency of the proposed approach.

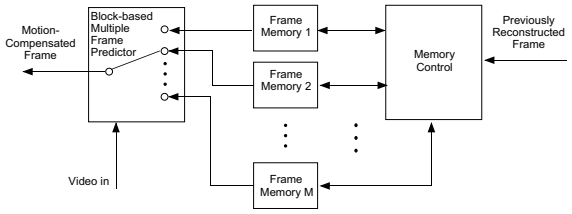


Figure 1: Motion-Compensated Long-Term Memory Predictor.

The architecture of the long-term memory predictor is depicted in Fig. 1. This figure shows an interframe predictor which uses a number of frame memories that are arranged using the memory control. The memory control may work in several modes of operation. A sliding window over time may be accommodated by the memory control unit as depicted in Fig. 1. Past decoded and reconstructed frames starting with the immediately preceding one ending with the frame which is decoded M time instants before are collected in the frame memories 1 to M. Alternatively, the set of past decoded and reconstructed frames may be subsampled using a scheme presumed by encoder and decoder. In general, several memory control modes of operation may be defined and the one which is used may be negotiated between encoder and decoder. In this work we will use the sliding window approach because of its simplicity.

Improvements when using long-term memory prediction can be expected in case of repetition of image sequence content. Examples for such an effect are moving video content with repetition in orientation or shape, covered and uncovered objects, shaking of the camera forth and back, etc. Also, the effect of sampling the video signal at various positions may contribute benefits in favor of long-term memory prediction. Hence, we expect the gains obtainable to increase with frame rate decrease.

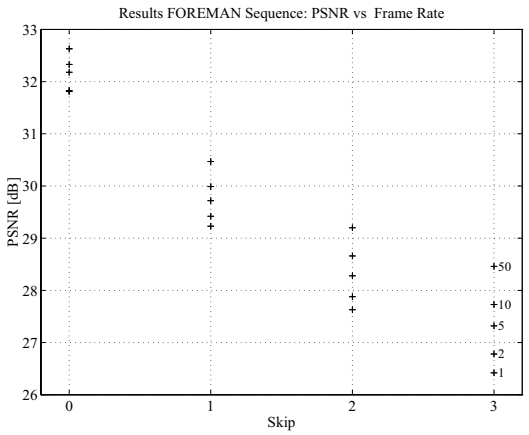


Figure 2: Prediction gain vs. frame skip for the sequence *Foreman* for memory sizes $M = 1, 2, 5, 10,$ and 50.

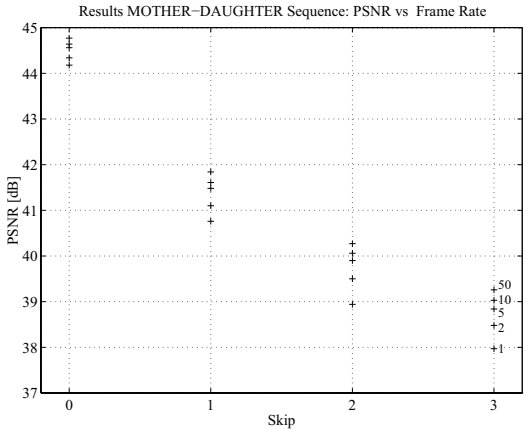


Figure 3: Prediction gain vs. frame skip for the sequence *Mother-Daughter*.

Figs. 2 and 3 show results of motion-compensated prediction experiments for the test sequences *Foreman* and *Mother-Daughter* respectively. The plots show the prediction performance in terms of PSNR vs. frame skip measured over 100 frames. The numbers 1, 2, 5, 10, and 50 relate to the various memory sizes. The long-term memory is built up by original frames sampled at the frame rate corresponding to the various frame skips using the sliding window memory control approach as described above. The block-matching is conducted by full search in the long-term memory buffer in the range $M \times [-16...15] \times [-16...15]$ on integer-pel positions followed by half-pel refinement using 16×16 blocks. As criterion for the block motion search we use the sum of the squared differences (SSD) between displaced and original frame.

2.1. Prediction of the Motion Vector

The spatial displacement vectors and the time delay have to be transmitted as side information requiring additional bit-rate. In order to control the bit-rate for the motion information the criterion for the block motion search is the minimization of the Lagrangian cost function

$$J(\boldsymbol{d}) = D(\boldsymbol{d}) + \lambda R(\boldsymbol{d} - \boldsymbol{p}), \qquad (1)$$

where $D(\boldsymbol{d})$ is a distortion measure for a given motion vector $\boldsymbol{d} = (d_x, d_y, d_t)$, such as the L_1 norm of the displaced frame difference, and $R(\boldsymbol{d} - \boldsymbol{p})$ is the bit-rate associated with a particular choice of the spatial displacement and time delay given its predictor $\boldsymbol{p} = (p_x, p_y, p_t)$. In this work, we set $p_t = 0$ for simplicity reasons.

The predictor for the spatial displacement vector (p_x, p_y) is computed using displacement vectors taken from a region of support (ROS). The ROS includes previously coded blocks that are close spatially and temporally as shown in Fig. 4. The block in the center of the five blocks in frame $t-1$ is located on the same position as the block with displacement vector DV is in frame t.

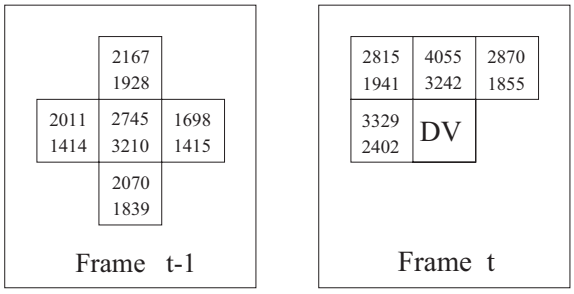


Figure 4: Region of support for predicting DV.

We measured correlation coefficients between the x (top) as well as the y component (bottom) of DV and the displacement vectors of the ROS for a set of training sequences by conventional block matching with the immediately preceding frames.

In order to determine (p_x, p_y), the time delay d_t for the current block is transmitted first. Then, the spatial displacements assigned to blocks in the ROS are selected in case their time delay coincides with the time delay of the current block. The result is sorted in descending order of the correlations between the spatial displacement parameters of the current block and the blocks of the ROS.

The predictor is formed by taking the median from the first three of the sorted spatial displacement vectors. In case there are less than three displacement vectors available, only the first displacement vector is used as predictor if it exists. Otherwise we set the predictor $(p_x, p_y) = (0, 0)$.

2.2. Huffman codes for the time delay

In order to transmit the time delay d_t, we have generated a Huffman code table for each memory size. In [5] the entropy-constrained design of a complete quadtree video codec is presented. The impact of the rate distortion optimized bit allocation on the Huffman code design is demonstrated. From the results in [5], we conclude that if we are using a rate-constrained bit allocation we should include it into the design procedure resulting in an iterative algorithm similar to that of [2]. For further details on entropy-constrained Huffman code design please refer to [5].

In our design algorithm, a set of 10 QCIF training sequences each with 10 seconds of video is encoded at 10 frames/s. The Lagrange multiplier is chosen to $\lambda = 150$ while using SSD as distortion measure and the overall motion vector bit-rate including the bit-rates of the spatial displacement and time delay. During encoding, histograms are gathered on the time delay parameter to design the Huffman codes which are employed in the next encoding step. The loop is performed until convergence is reached, i.e., the changes in the overall Lagrangian costs become small. The spatial displacements (d_x, d_y) are transmitted using the H.263 MVD table [7].

3. INTEGRATION INTO H.263

In order to evaluate the proposed technique the long-term memory MCP is integrated into an H.263 video codec. The H.263 inter-prediction modes INTER, INTER-4V, and UNCODED[1] are extended to long-term memory MC. The INTER and UNCODED mode are assigned one code word representing the variable time delay for the entire macroblock. The INTER-4V mode utilizes four time parameters each associated to one of the four 8×8 motion vectors.

To run our H.263 as well as our long-term memory coder, we have implemented a modified encoding strategy as utilized by the TMN-2.0 coder, the test model for the H.263 standard.[2] Our encoding strategy differs for the motion estimation and the mode decision, where our scheme is motivated by rate-distortion theory.

The problem of optimum bit allocation to the motion vectors and the residual coding in any hybrid video coder is a non-separable problem requiring a high amount of computation. To circumvent this joint optimization, we split the problem

[1] The UNCODED mode is an INTER mode for which the COD bit indicates copying the macroblock from the previous frame without residual coding [7].

[2] The TMN-2.0 codec is available via anonymous ftp to `bonde.nta.no`.

into two parts: motion estimation and mode decision.

The motion estimation is performed as described above using the minimization of the Lagrangian cost function. For each frame the best motion vector is found by full search on integer-pel positions followed by half-pel refinement. The integer-pel search is conducted over the range $[-16 \ldots 15] \times [-16 \ldots 15]$ pels. The distortion is computed by the sum of the absolute differences (SAD) between displaced and original frame and the rate is computed by the bit-rate occupied for the motion vector. The impact of overlapped block motion compensation is neglected in the motion estimation.

Given the displacements for each particular mode we are computing the overall rate distortion costs. The distortion is computed by SSD between reconstructed and original frame, and the rate is computed including the rates of macroblock headers, motion parameters, and DCT quantization coefficients. The mode with the smallest Lagrangian cost is selected for transmission to the decoder. In case of long-term memory MCP, the motion estimation followed by the mode decision as described is conducted for each frame in the frame buffer.

Since there are now two Lagrangian cost functions to be minimized, we employ two different Lagrange multipliers: one for the motion search (λ_{motion}), the other one for the mode decision (λ_{mode}). Furthermore, the distortion measures differ because of complexity reasons. Hence, the selection of the Lagrange parameters remains rather difficult in our coder. In this work, we employ the heuristic $\lambda_{motion} = \sqrt{\lambda_{mode}}$, which appears to be sufficient. The parameter λ_{mode} itself is derived from the rate distortion curve that we computed using the TMN-2.0 H.263 coder.

4. SIMULATION RESULTS

In this section we demonstrate the performance of the proposed approach. Figs. 5 and 6 show the results obtained for the test sequences *Foreman* and *Mother-Daughter*, respectively. These sequences where not part of the training set. The coder is run with constant quantizer when coding 100 frames at 10 frames/s. All results are generated from decoded bit streams.

Figs. 5 and 6 show the average PSNR from reconstructed frames produced by the TMN-2.0 codec, our rate distortion optimized H.263 codec and the long-term memory prediction codec vs. overall bit-rate.

The size of the long-term memory is selected as 2, 5, 10, and 50 frames. The curve is generated by varying the Lagrange parameter and the

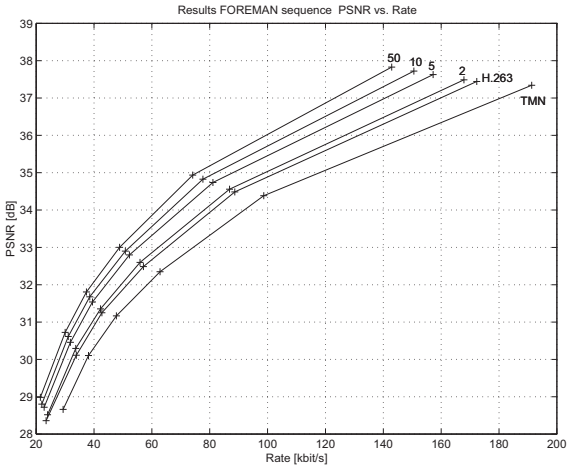


Figure 5: PSNR vs. overall bit-rate for the sequence *Foreman*.

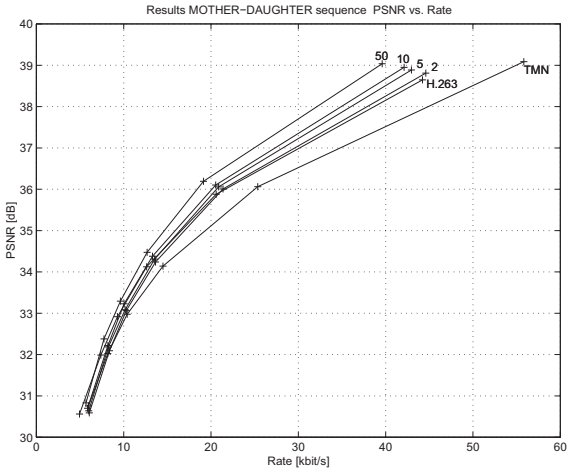


Figure 6: PSNR vs. overall bit-rate for the sequence *Mother-Daughter*.

DCT quantization parameter accordingly. Hence, the points marked with "+" in the plots relate to values computed from entire sequences. The long-term memory buffer is built up simultaneously at encoder and decoder by reconstructed frames. The results are obtained by measuring over frames 50...100, in order to avoid the effects at the beginning of the sequence.

The impact of rate-constrained encoding strategy is visible when comparing our H.263 codec with TMN-2.0. For both sequences, a PSNR gain of 0.6 dB is due to our rate distortion optimization. We noticed that the usage of the full motion estimation search range $[-16 \ldots 15] \times [-16 \ldots 15]$ for the 8×8 block displacement vectors provides most of the gain for our H.263 codec. The TMN-2.0 coder only permits the use of half-pel positions for the 8×8 block displacement vectors that surround the previously found 16×16 block dis-

placement vector, which is searched in the range $[-16 \ldots 15] \times [-16 \ldots 15]$. We have observed that using the full search range for the 8×8 block displacement vectors leads to improved coding performance for our rate-constrained motion estimation, whereas for the TMN-2.0 we get worse results, since no rate-constraint is employed. This effect is even stronger, in case of long-term memory MCP where we have much more search positions: $M \times [-16 \ldots 15] \times [-16 \ldots 15]$.

When comparing long-term memory MCP to our own rate-distortion-optimized H.263 coder, the PSNR gains achieved are about 1.4 dB for the *Foreman* sequence and 0.9 dB for the *Mother-Daughter* sequence at the high bit-rate end, when using a memory of 50 frames. These results demonstrate that utilizing the long-term memory we get an improved motion-compensating prediction scheme in terms of rate distortion performance. The gains tend to vanish for very low bit-rates which is in line with our interpretation of MCP as ECVQ.

Figs. 7 a) and 8 a) show the bit-rate for the motion vectors including the bit-rates for the spatial displacements and the time delay. Two tendencies can be observed for our H.263 and long-term memory coder: the motion vector bit-rate increases as overall bit-rate increases, and the curves merge at the very low bit-rate end for the various memory sizes.

However, the curve for the TMN-2.0 coder shows a completely different behavior. For the sequence *Foreman*, the motion vector bit-rate decreases as overall bit-rate increases. This results from the fact that the TMN-2.0 does not employ a rate-constraint and motion estimation is performed using the reconstructed frames (for TMN-2.0 as well as for our coder). As bit-rate decreases these reconstructed frames get noisier and since the regularization by the rate-constraint is missing for the TMN-2.0, the estimates for the motion data get noisier requiring a higher bit-rate.

Figs. 7 b) and 8 b) show the amount of bit-rate used for the spatial displacement vectors. Note that for our rate-constrained motion estimation the bit-rate occupied by the spatial displacement vectors is roughly independent from the size of the long-term memory. This indicates that our prediction scheme for the spatial displacement vectors works sufficiently for all memory sizes.

Finally, Figs. 7 c) and 8 c) depict the bit-rate required to transmit the time delay. As the long-term memory size increases, the bit-rate for the time delay increases. But this increase is well compensated by the reduction in bit-rate that is used for coding of the MCP residual.

5. CONCLUSIONS

By using motion-compensated long-term memory prediction we obtain a significantly improved video codec in terms of rate distortion performance. The gains are achieved at the expense of increased computational complexity and memory requirement. Main ingredient for the successful use of motion-compensated long-term memory prediction is the rate-constrained encoding strategy. In comparison to TMN-2.0, rate-constrained motion estimation and mode decision provides PSNR gains up to 0.6 dB. On top of that, our long-term memory prediction coder yields additional PSNR gains of 1.4 dB for the *Foreman* sequence and 0.9 dB for *Mother-Daughter* when using a long-term memory of 50 frames.

6. ACKNOWLEDGEMENTS

Thanks to Eckehard Steinbach, Uwe Horn, Niko Färber, and Klaus Stuhlmüller for helpful discussions.

7. REFERENCES

[1] T. Wiegand, X. Zhang, and B. Girod, "Motion Compensating Long Term Memory Prediction", in *Proceedings of the IEEE International Conference on Image Processing*, Santa Barbara, USA, Oct. 1997, To be published.

[2] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy Constrained Vector Quantiza tion", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 1, pp. 31–42, Jan. 1989.

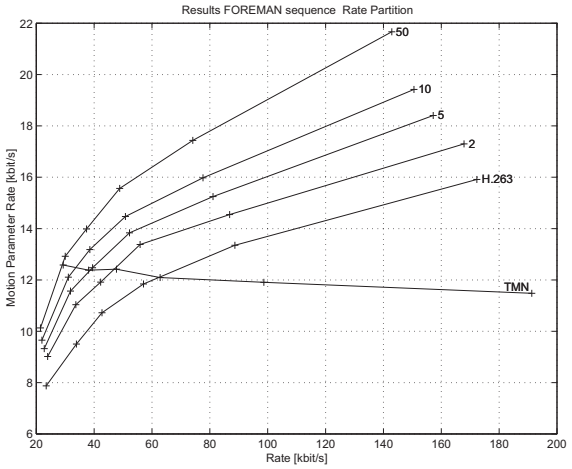
[3] B. Girod, "Rate Constrained Motion Estimation", in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Chicago, USA, Sept. 1994, pp. 1026–1034, (Invited paper).

[4] G. J. Sullivan and R. L. Baker, "Efficient Quadtree Coding of Images and Video", *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 327–331, May 1994.

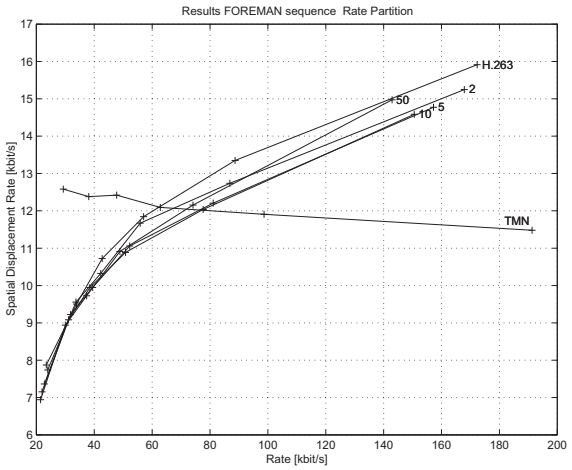
[5] T. Wiegand, M. Flierl, and B. Girod, "Entropy Constrained Design of Quadtree Video Coding Schemes", in *Proceedings of the IEE International Conference on Image Processing and its Applications*, Dublin, Ireland, July 1997, pp. 36–40.

[6] K. W. Stuhlmüller, A. Salai, and B. Girod, "Rate Constrained Contour Representation for Region Based Motion Compensation", in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Orlando, USA, Mar. 1996.

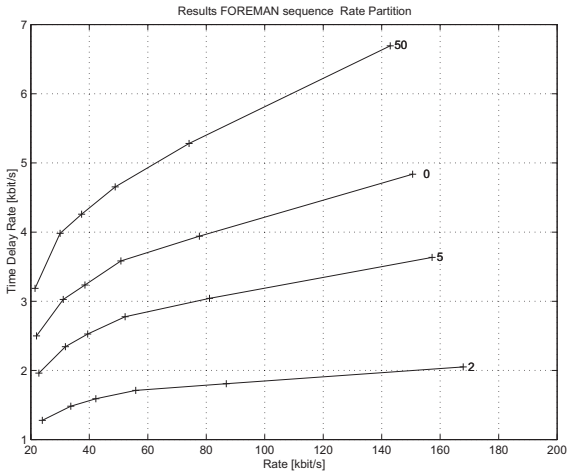
[7] ITU T Recommendation H.263, "Video Coding for Low Bitrate Communication", Draft, Dec. 1995.



a)

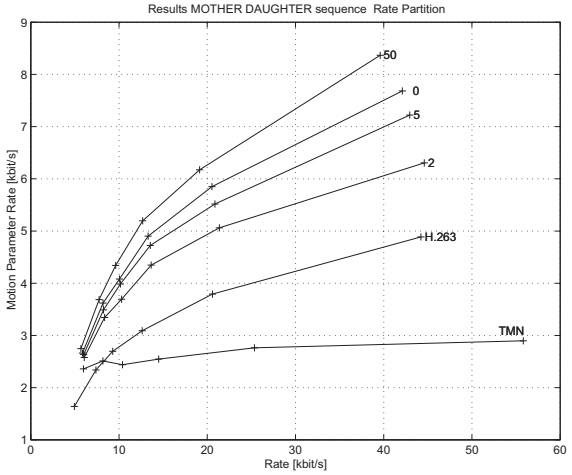


b)

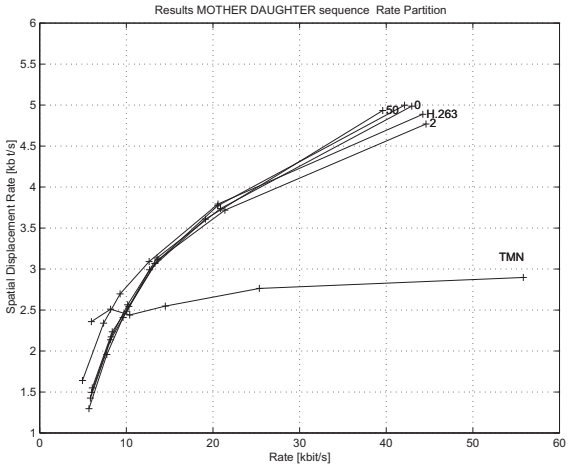


c)

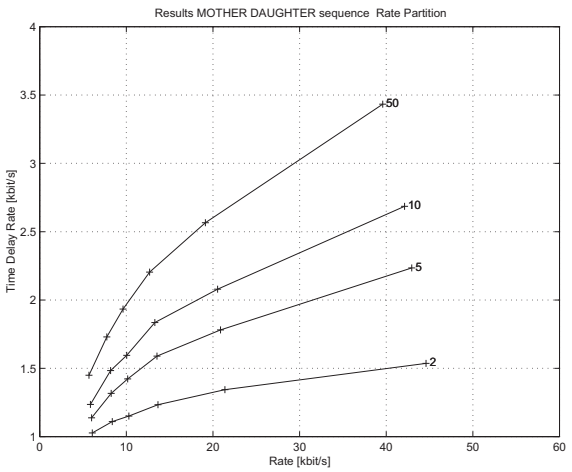
Figure 7: Motion rate vs. overall bit-rate for the sequence *Foreman*.



a)



b)



c)

Figure 8: Motion rate vs. overall bit-rate for the sequence *Mother-Daughter*.