

Poznań University of Technology



Chair of Multimedia Telecommunications and Microelectronics

Rafał Lange

**Multiresolution Representation of Motion Vectors
in Video Compression**

Doctoral Dissertation

Advisor: Prof. Marek Domański

Poznań, 2006

Politechnika Poznańska



Katedra Telekomunikacji Multimedialnej i Mikroelektroniki

Rafał Lange

**Wielorozdzielczościowe reprezentacje
wektorów ruchu dla zastosowań w kompresji
sekwencji wizyjnych**

Rozprawa Doktorska

Promotor: prof. dr hab. inż. Marek Domański

Poznań, 2006

This dissertation is dedicated to my beloved parents Irena and Edward, who gave me wonderful childhood and all opportunities.

I would like to thank to all important people, who have always been in the right place, especially to my wife Kasia, and to my wonderful sisters: Renata and Elżbieta.

I would like to express special thanks and appreciation to Professor Marek Domański, for his time, help and ideas that guided me towards completing this dissertation.

Rozprawa ta dedykowana jest moim ukochanym rodzicom Irenie i Edwardowi, którzy dali mi cudowne dzieciństwo oraz wszelkie możliwości.

Chciałbym podziękować wszystkim ważnym osobom, które zawsze były we właściwym miejscu, szczególnie mojej żonie Kasi oraz moim cudownym siostrom: Renacie i Elżbiecie.

Chciałbym również wyrazić szczególne podziękowania oraz wyrazy wdzięczności panu profesorowi Markowi Domańskiemu, za jego czas, pomoc oraz pomysły, które doprowadziły mnie do ukończenia tej rozprawy.

Table of contents

List of tables	8
List of figures	12
List of symbols and abbreviations	23
Abstract	27
Streszczenie	27
Chapter 1. Introduction	29
1.1. Multiresolution video representation	31
1.2. Motion vectors in hybrid video coding.....	38
1.3. Goals and thesis of the work.....	38
1.4. Research methodology.....	40
1.5. Quality measurement	42
1.6. Thesis overview	45
Chapter 2. Video coding with motion-compensated prediction	47
2.1. Hybrid video coding	49
2.2. The paradigm of motion-compensated prediction	52
2.3. Motion estimation	53
2.3.1. Block-matching algorithm of motion estimation.....	53
2.3.2. Pel-recursive algorithms	54
2.4. Scalable video coding with motion compensated-prediction	55
2.4.1. Wavelet codecs	56
2.4.2. Hybrid scalable codec with DCT-based transform.....	57
2.5. Summary	59
Chapter 3. Motion model and its representation in non-scalable video coding	61
3.1. Introduction.....	63
3.2. Motion compensation in advanced hybrid coders	64
3.2.1. Motion model.....	64
3.2.1.1. Variable block size.....	64
3.2.1.2. Overlapped Motion Compensation.....	65
3.2.1.3. Fractional motion vectors	69
3.2.1.4. Motion vectors over pictures boundaries.....	72
3.2.1.5. Global Motion Compensation.....	72

3.2.1.6. Motion compensation using control-grid interpolation.....	74
3.2.2. Temporal prediction modes.....	75
3.2.3. Multiframe motion compensation – multi-hypothesis prediction	76
3.2.4. Inference of motion information	77
3.3. Motion vectors representation	79
3.3.1. Motion vector field.....	79
3.3.2. First-order prediction.....	81
3.3.3. Median prediction	83
3.3.4. Motion vector coding using vector quantization.....	85
3.3.5. Motion vector coding based on Minimum Bitrate Prediction.....	87
3.4. Conclusions and summary	89
Chapter 4. Advanced coding of motion vectors	91
4.1. Introduction	93
4.2. Component-wise prediction	96
4.2.1. Median prediction of motion vectors	96
4.2.2. Directional predictions of motion vectors.....	98
4.2.3. Component-wise motion vector prediction – experimental results.....	99
4.2.3.1. Average absolute values of motion vector prediction residuals in 4CIF video sequences.....	99
4.2.3.2. Average absolute values of motion vector prediction residuals in CIF video sequences (P-frames).....	101
4.2.3.3. Average absolute values of motion vector prediction residuals in CIF video sequences (B-frames)	104
4.2.4. Component-wise motion vector prediction – conclusions	106
4.3. Vector median prediction	108
4.3.1. Median filtering of vector-valued signals	108
4.3.2. Vector median motion vector prediction - experimental results	109
4.3.2.1. Bitrate and distortion.....	110
4.3.2.2. Average absolute values of components of motion vector residual.....	111
4.3.3. Vector median motion vector prediction - conclusions	118
4.4. Entropy coding of motion vectors	119
4.4.1. Coding of motion vectors using Exp-Golomb codes	120
4.4.2. Coding of motion vectors using CABAC	122

4.4.3. Comparison of Exp-Golomb coding against CABAC coding of motion vectors – experimental results	123
4.4.4. Comparison of Exp-Golomb coding against CABAC coding of motion vectors – conclusions	128
4.5. Summary	130
Chapter 5. Multiresolution motion fields	131
5.1. A problem of multiresolution motion representation	133
5.2. Multiresolution motion estimation in scalable video coding.....	134
5.3. Correlation of multiresolution motion vectors.....	135
5.4. Multiresolution estimation of motion vectors – experimental results	138
5.4.1. Length of differential motion vectors in P-frames.....	138
5.4.2. Length of differential motion vectors in B-frames	142
5.4.3. Standard deviation of the differential motion vector length in P-frames	146
5.4.4. Standard deviation of the differential motion vector length in B-frames	149
5.4.5. Average differential motion vector length and average standard deviation of the differential motion vector length	153
5.4.6. Mutual matching of motion vector fields	156
5.4.7. Visualization of motion vector fields	160
5.5. Conclusions.....	165
5.6. Summary	167
Chapter 6. Joint encoding of multiresolution motion vectors	169
6.1. Towards new standard of scalable video coding	171
6.2. Introduction to joint encoding of multiresolution motion vectors.....	174
6.3. Modification of scalable coder	175
6.3.1. Joint motion estimation.....	176
6.3.2. Joint multiresolution motion representation	178
6.4. Joint encoding of multiresolution motion vectors - experimental results.....	180
6.4.1. Motion estimation using optical flow technique	180
6.4.2. Joint encoding of motion vectors in scalable video codec	183
6.5. Conclusions.....	187
6.6. Summary	188
Chapter 7. Multiresolution prediction of motion vectors	189
7.1. Introduction.....	191
7.2. Difficulties in spatial prediction of motion vectors	191

7.2.1. Boundary of frame or slice.....	192
7.2.2. Intra-frame coding in adjacent macroblocks.....	194
7.2.3. Different types of prediction in adjacent blocks.....	194
7.2.4. Different reference frames in adjacent blocks.....	195
7.3. Inter-layer motion vector prediction.....	196
7.3.1. Original proposal of inter-layer motion vector prediction – Implicit Inter-Layer Prediction (IILP).....	196
7.3.2. Possible approaches to inter-layer motion vector prediction.....	197
7.3.2.1. IILP using directly co-located block.....	197
7.3.2.2. IILP using the best block.....	198
7.3.2.3. The impact of reduced accuracy of interpolated motion vectors from the base layer.....	199
7.3.3. Proposed variants of IILP.....	199
7.4. Modification of the scalable coder.....	201
7.5. IILP prediction of motion vectors – experimental results.....	202
7.5.1. Bitrate and distortion.....	203
7.5.2. Average absolute values of motion vector prediction residuals.....	208
7.5.3. The decrease of average absolute value of motion vector prediction residuals.....	212
7.5.4. Subjective evaluation of quality.....	215
7.6. IILP prediction of motion vectors – conclusions.....	217
7.7. Inter-layer prediction with explicit signaling of prediction mode – MPEG approach.....	218
7.7.1. MPEG proposal of advanced scalable video coding - SVC.....	218
7.7.2. Inter-layer motion vector prediction in SVC.....	219
7.8. Comparison of inter-layer motion prediction techniques in SVC codec.....	220
7.8.1. Comparison of inter-layer motion prediction techniques in SVC codec – experimental results.....	220
7.8.1.1. Bitrate and distortion.....	221
7.8.1.2. Average absolute values of motion vector prediction residuals.....	225
7.8.1.3. Subjective evaluation of the quality.....	228
7.8.1.4. Complexity estimation.....	230
7.8.2. Comparison of inter-layer motion prediction techniques in SVC codec – conclusions.....	231

7.9. Summary	233
Chapter 8. Joint multiresolution coding of motion vectors in temporally scalable codec	235
8.1. Temporal scalability	237
8.2. Original proposal of joint multiresolution coding of motion vectors in B-frames	238
8.3. Modification of SVC codec	240
8.4. Joint multiresolution coding of motion vectors in B-frames– experimental results	241
8.4.1. Bitrate and distortion	242
8.4.2. The bitrate of motion vector residuals sub-bitstream	248
8.4.3. The bitrate of transform coefficients sub-bitstream.....	251
8.4.4. The bitrate of control data sub-bitstream.....	253
8.4.5. Subjective evaluation of the quality	256
8.4.6. Complexity estimation.....	258
8.5. Conclusions.....	259
8.6. Summary.....	262
Chapter 9. Results and conclusions	263
9.1. Results.....	265
9.2. Conclusions.....	268
Bibliography	271
Author’s contributions	271
Other references used in the dissertation	273
Annex – Test sequences	287

List of tables

Tab. 2.1. Available prediction modes in enhancement layer of the scalable video coder from Fig. 2.4.....	58
Tab. 3.1. Maximum number of motion vectors per macroblock for various partitioning in AVC codec.	65
Tab. 3.2. The influence of f_code in MPEG-2 on extreme values of motion vector component residual in full-pel units.....	83
Tab. 4.1. Inter-frame coded macroblock types in P-pictures in AVC/H.264.....	93
Tab. 4.2. The result of motion vector prediction, depending on availability of a adjacent macroblock and reference frame used in inter-frame prediction.	97
Tab. 4.3. Average prediction error module for motion vector components.....	100
Tab. 4.4. Bitrate (kbps) and PSNR (dB) in <i>Bus</i> (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.....	110
Tab. 4.5. Bitrate (kbps) and PSNR (dB) in <i>Football</i> (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.	110
Tab. 4.6. Bitrate (kbps) and PSNR (dB) in <i>Foreman</i> (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.	111
Tab. 4.7. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.	111
Tab. 4.8. Number of operations depending on median type for two-dimensional vector.	119
Tab. 4.9. The examples of Exp-Golomb codewords and their mapping onto signed values. First 9 codewords are given.	121
Tab. 4.10. Average difference between length of the codeword using CABAC and length of the codeword using Exp-Golomb coding for corresponding values of motion vector residuals.....	128
Tab. 5.1. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>Bus</i> sequence for different quantization parameter Q_p	153
Tab. 5.2. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>Football</i> sequence for different quantization parameter Q_p	154

Tab. 5.3. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>Foreman</i> sequence for different quantization parameter Q_p .	154
Tab. 5.4. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>Mobile</i> sequence for different quantization parameter Q_p .	155
Tab. 5.5. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>City</i> sequence for different quantization parameter Q_p .	155
Tab. 5.6. Average values of differential motion vector length and standard deviation of differential motion vector length in <i>Crew</i> sequence for different quantization parameter Q_p .	156
Tab. 5.7. Average mutual covering of motion vectors in <i>Bus</i> sequence for different quantization parameter Q_p .	157
Tab. 5.8. Average mutual covering of motion vectors in <i>Football</i> sequence for different quantization parameter Q_p .	157
Tab. 5.9. Average mutual covering of motion vectors in <i>Foreman</i> sequence for different quantization parameter Q_p .	158
Tab. 5.10. Average mutual covering of motion vectors in <i>Mobile</i> sequence for different quantization parameter Q_p .	158
Tab. 5.11. Average mutual covering of motion vectors in <i>City</i> sequence for different quantization parameter Q_p .	159
Tab. 5.12. Average mutual covering of motion vectors in <i>Crew</i> sequence for different quantization parameter Q_p .	159
Tab. 6.1. The timeline of MPEG works on Scalable Video Coding compared against author's activities.	173
Tab. 7.1. Techniques of the proposed multiresolution motion vectors prediction.	200
Tab. 7.2. The result of inter-layer motion vector prediction, depending on availability of adjacent macroblocks and reference frame used in inter-frame prediction.	200
Tab. 7.3. Bitrate (kbps) and PSNR (dB) in <i>Bus</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.	203

Tab. 7.4. Bitrate (kbps) and PSNR (dB) in <i>Football</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.....	203
Tab. 7.5. Bitrate (kbps) and PSNR (dB) in <i>Foreman</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.....	204
Tab. 7.6. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.....	204
Tab. 7.7. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.....	204
Tab. 7.8. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.....	205
Tab. 7.9. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>Bus</i> sequence (352×288).....	212
Tab. 7.10. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>Football</i> sequence (352×288).....	213
Tab. 7.11. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>Foreman</i> sequence (352×288).....	213
Tab. 7.12. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>Mobile</i> sequence (352×288).....	213
Tab. 7.13. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>City</i> sequence (704×576).....	214
Tab. 7.14. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in <i>Crew</i> sequence (704×576).....	214
Tab. 7.15. Prediction of the motion vectors in SVC codec.....	219

Tab. 7.17. Bitrate (kbps) and PSNR (dB) in <i>Football</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.....	222
Tab. 7.18. Bitrate (kbps) and PSNR (dB) in <i>Foreman</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.....	222
Tab. 7.19. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.....	223
Tab. 7.20. Execution time of encoder for various techniques of inter-layer prediction of motion vectors in SVC video codec	230
Tab. 7.21. Execution time of decoder for various techniques of inter-layer prediction of motion vectors in SVC video codec.	231
Tab. 8.1. Multiresolution coding of motion vectors in B-frames, techniques and bitstream syntax.	241
Tab. 8.2. Bitrate (kbps) and PSNR (dB) in <i>Bus</i> (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.....	242
Tab. 8.3. Bitrate (kbps) and PSNR (dB) in <i>Football</i> (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.	243
Tab. 8.4. Bitrate (kbps) and PSNR (dB) in <i>Foreman</i> (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.	243
Tab. 8.5. Bitrate (kbps) and PSNR (dB) in <i>Mobile</i> (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.	243
Tab. 8.6. Average decrease of PSNR in B-frames as compared with SVC codec for various techniques of motion vectors coding.	248
Tab. 8.7. Execution time of encoder for various techniques of inter-layer prediction of motion vectors in SVC video codec	258
Tab. 8.8. Execution time of decoder for various techniques of inter-layer prediction of motion vectors in SVC video codec.	259

List of figures

Fig. 1.1. The resolution of a video sequence and total number of luminance and chrominance samples for YUV 4:2:0 sampling scheme.	32
Fig. 1.2. Multiresolution video transmission over heterogeneous networks and protocols.	33
Fig. 1.3. <i>Simulcast</i> video coding.	34
Fig. 1.4. Scalable video coding.	35
Fig. 1.5. Multiresolution video representation using scalable video coding.	36
Fig. 1.6. Comparison of quality decrease. Single-layer coding versus scalable coding with unequal error protection [Gir95].	37
Fig. 1.7. The voting scale of Single Stimulus MultiMedia method.	43
Fig. 1.8. Hypothetical rate-distortions curves.	44
Fig. 2.1. Hybrid video encoder with motion compensated prediction.	49
Fig. 2.2. Block-matching algorithm of motion estimation.	54
Fig. 2.3. Scalable hybrid coder with layered multiresolution video representation.	57
Fig. 3.1. Segmentation of a frame in video sequence using blocks with variable size (from 16×16 to 4×4 luminance sample). Frame 58 from <i>Mobile</i> sequence, frame 44 from <i>Foreman</i> sequence.	64
Fig. 3.2. Overlapped Motion Compensation scheme, as proposed by Nogaki and Ohta.	66
Fig. 3.3. Motion vectors used for overlapped motion compensation in H.263 codec. Four cases are shown, for each of the 8×8 block of the macroblock.	67
Fig. 3.4. The example of motion compensated prediction using integer and fractional motion vectors.	69
Fig. 3.5. Interpolation of fractional luma samples in MPEG-2 and H.263 video coding algorithms.	70
Fig. 3.6. Interpolation of fractional luma samples in AVC/H.264 video coding algorithm.	71
Fig. 3.7. Control-grid interpolation scheme.	74
Fig. 3.8. Forward, backward and bidirectional motion-compensated prediction.	76
Fig. 3.9. Motion-compensated prediction using long-term memory.	77
Fig. 3.10. Derivation of motion vectors in temporal motion inference mode in AVC/H.264.	78

Fig. 3.11. Visualizations of motion vector fields estimated with block matching algorithm.....	80
Fig. 3.12. Predictive coding of motion vector.	80
Fig. 3.13. Macroblock used in DPCM motion vector prediction.	81
Fig. 3.13. Macroblocks used in median motion vector prediction.	83
Fig. 3.15. Bits required for encoding motion vector residual in H.263.....	84
Fig. 3.16. Motion vector representation using vector quantization.	86
Fig. 3.16. Blocks and motion vectors used in Minimum Bitrate Prediction method of motion vector coding.	87
Fig. 3.18. Determination of the value of component of current motion vector in MBP method.	88
Fig. 4.1. Partitioning of macroblock in AVC/H.264 video coder.....	94
Fig. 4.2. Motion vector assignment in the fragment of frame 32 from sequence <i>Mobile</i> 176×144 in AVC/H.264 video codec.	95
Fig. 4.3. Blocks used for motion vector prediction when median motion vector prediction is performed for various sizes of partition.....	97
Fig. 4.4. Directional prediction of motion vectors in 16×8 and 8×16 macroblock partitioning modes.	98
Fig. 4.5. Histogram of motion vector prediction residuals for various bitrate	100
Fig. 4.6. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	102
Fig. 4.7. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	102
Fig. 4.8. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	103
Fig. 4.9. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	103
Fig. 4.10. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	104

Fig. 4.11. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	105
Fig. 4.12. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	105
Fig. 4.13. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	106
Fig. 4.14. Average absolute values of components of motion vector residual for various median predictions in P-frames, <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	112
Fig. 4.15. Average absolute values of components of motion vector residual for various median predictions in P-frames, <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	112
Fig. 4.16. Average absolute values of components of motion vector residual for various median predictions in P-frames, <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	113
Fig. 4.17. Average absolute values of components of motion vector residual for various median predictions in P-frames, <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	113
Fig. 4.18. Average absolute values of components of motion vector residual for various median predictions in B-frames, <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	114
Fig. 4.19. Average absolute values of components of motion vector residual for various median predictions in B-frames, <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	114
Fig. 4.20. Average absolute values of components of motion vector residual for various median predictions in B-frames, <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	115
Fig. 4.21. Average absolute values of components of motion vector residual for various median predictions in B-frames, <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	115

Fig. 4.22. Average number of bits per motion vector component for various median predictions, <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	116
Fig. 4.23. Average number of bits per motion vector component for various median predictions, <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	117
Fig. 4.24. Average number of bits per motion vector component for various median predictions, <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	117
Fig. 4.25. Average number of bits per motion vector component for various median predictions, <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	118
Fig. 4.26. Block diagram of motion vector residual encoding using CABAC entropy coder in AVC/H.264.	122
Fig. 4.27. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Bus</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	124
Fig. 4.28. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Foreman</i> sequence (352×288, IBPBP), AVC/H.264 video codec.	125
Fig. 4.29. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Mobile</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	125
Fig. 4.30. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	126
Fig. 4.31. Number of bits for residual motion vector component in P-frames. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	126
Fig. 4.32. Number of bits for residual motion vector component in B-frames. Comparison of Exp-Golomb codes against CABAC entropy coding in <i>Football</i> sequence (352×288, IBPBP), AVC/H.264 video codec.....	127
Fig. 5.1. Multiresolution video coding using pyramid of video coders.....	133
Fig. 5.2. Two-layer scalable video encoder, which has been used in experiments.	135
Fig. 5.3. Average length of differential motion vector in P-frames in <i>Bus</i> sequence for different quantization parameter Q_p	139
Fig. 5.4. Average length of differential motion vector in P-frames in <i>Football</i> sequence for different quantization parameter Q_p	139

Fig. 5.5. Average length of differential motion vector in P-frames in <i>Foreman</i> sequence for different quantization parameter Q_p .	140
Fig. 5.6. Average length of differential motion vector in P-frames in <i>Mobile</i> sequence for different quantization parameter Q_p .	140
Fig. 5.7. Average length of differential motion vector in P-frames in <i>City</i> sequence for different quantization parameter Q_p .	141
Fig. 5.8. Average length of differential motion vector in P-frames in <i>Crew</i> sequence for different quantization parameter Q_p .	141
Fig. 5.9. Average length of differential motion vector in B-frames in <i>Bus</i> sequence for different quantization parameter Q_p .	143
Fig. 5.10. Average length of differential motion vector in B-frames in <i>Football</i> sequence for different quantization parameter Q_p .	143
Fig. 5.11. Average length of differential motion vector in B-frames in <i>Foreman</i> sequence for different quantization parameter Q_p .	144
Fig. 5.12. Average length of differential motion vector in B-frames in <i>Mobile</i> sequence for different quantization parameter Q_p .	144
Fig. 5.13. Average length of differential motion vector in B-frames in <i>City</i> sequence for different quantization parameter Q_p .	145
Fig. 5.14. Average length of differential motion vector in B-frames in <i>Crew</i> sequence for different quantization parameter Q_p .	145
Fig. 5.15. Standard deviation of differential motion vector length in P-frames in <i>Bus</i> sequence for different quantization parameter Q_p .	146
Fig. 5.16. Standard deviation of differential motion vector length in P-frames in <i>Football</i> sequence for different quantization parameter Q_p .	147
Fig. 5.17. Standard deviation of differential motion vector length in P-frames in <i>Foreman</i> sequence for different quantization parameter Q_p .	147
Fig. 5.18. Standard deviation of differential motion vector length in P-frames in <i>Mobile</i> sequence for different quantization parameter Q_p .	148
Fig. 5.19. Standard deviation of differential motion vector length in P-frames in <i>City</i> sequence for different quantization parameter Q_p .	148
Fig. 5.20. Standard deviation of differential motion vector length in P-frames in <i>Crew</i> sequence for different quantization parameter Q_p .	149
Fig. 5.21. Standard deviation of differential motion vector length in B-frames in <i>Bus</i> sequence for different quantization parameter Q_p .	150

Fig. 5.22. Standard deviation of differential motion vector length in B-frames in <i>Football</i> sequence for different quantization parameter Q_p	150
Fig. 5.23. Standard deviation of differential motion vector length in B-frames in <i>Foreman</i> sequence for different quantization parameter Q_p	151
Fig. 5.24. Standard deviation of differential motion vector length in B-frames in <i>Mobile</i> sequence for different quantization parameter Q_p	151
Fig. 5.25. Standard deviation of differential motion vector length in B-frames in <i>City</i> sequence for different quantization parameter Q_p	152
Fig. 5.26. Standard deviation of differential motion vector length in B-frames in <i>Ice</i> sequence for different quantization parameter Q_p	152
Fig. 5.27. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Bus</i> sequence.	161
Fig. 5.28. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Bus</i> sequence.	161
Fig. 5.29. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Football</i> sequence.	161
Fig. 5.30. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Football</i> sequence.	162
Fig. 5.31. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Foreman</i> sequence.	162
Fig. 5.32. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Foreman</i> sequence.	162
Fig. 5.33. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Mobile</i> sequence.	163
Fig. 5.34. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in <i>Mobile</i> sequence.	163
Fig. 5.35. Interpolated motion vectors estimated for resolution 352×288 , motion vectors estimated for resolution 704×576 and differential motion vectors in <i>City</i> sequence. ...	163

Fig. 5.36. Interpolated motion vectors estimated for resolution 352×288, motion vectors estimated for resolution 704×576 and differential motion vectors in <i>City</i> sequence.....	164
Fig. 5.37. Interpolated motion vectors estimated for resolution 352×288, motion vectors estimated for resolution 704×576 and differential motion vectors in <i>Crew</i> sequence...	164
Fig. 5.38. Interpolated motion vectors estimated for resolution 352×288, motion vectors estimated for resolution 704×576 and differential motion vectors in <i>Crew</i> sequence...	164
Fig. 6.1. The loss of Mean Opinion Score (MOS) against AVC/H.264 in subjective quality comparison of scalable codecs, March 2004.....	172
Fig. 6.2. Basic structure of scalable hybrid video coder with independent motion estimation and compensation.	175
Fig. 6.3. Structure of scalable coder with joint motion estimation and motion vector encoding.	176
Fig. 6.4. Interpolation of the base-layer motion vectors.	179
Fig. 6.5. Motion vectors from base layer and enhancement layer used in joint motion vectors representation.....	179
Fig. 6.6. Bitrates of motion vectors, transform coefficients and total bitrate in <i>Basket</i> sequence (352×288, IPPP), AVC/H.264 video codec. Various algorithms of motion estimation have been used.....	181
Fig. 6.7. Bitrates of motion vectors, transform coefficients and total bitrate in <i>Fun</i> sequence (352×288, IPPP), AVC/H.264 video codec. Various algorithms of motion estimation have been used.....	182
Fig. 6.8. Motion vectors estimated using block matching (on the left) and optical flow (on the right). Frame number 17 from <i>Basket</i> sequence.	182
Fig. 6.9. Motion vectors estimated using block matching (on the left) and optical flow (on the right). Frame number 10 from <i>Fun</i> sequence.	183
Fig. 6.10. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in <i>Basket</i> sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.....	184
Fig. 6.11. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in <i>Stefan</i> sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.	185
Fig. 6.12. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in <i>Fun</i> sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.	185

Fig. 6.13. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in <i>Football</i> sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.....	186
Fig. 6.14. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in <i>Cheer</i> sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.	186
Fig. 7.1. Examples of grouping macroblocks into slices using FMO.....	193
Fig. 7.2. Boundary macroblocks.....	194
Fig. 7.3. Neighboring macroblocks use various prediction modes, fragment of hypothetical video frame.	195
Fig. 7.4. Blocks used in inter-layer prediction of motion vectors.	196
Fig. 7.5. IILP motion vector prediction using directly co-located block.....	197
Fig. 7.6. IILP motion vector prediction using the best motion vector from the low-resolution layer.	198
Fig. 7.7. Modification of scalable video hybrid coder with inter-layer motion vector prediction.	201
Fig. 7.8. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Bus</i> sequence (352×288).....	205
Fig. 7.9. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Football</i> sequence (352×288).	206
Fig. 7.11. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Mobile</i> sequence (352×288).	207
Fig. 7.12. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>City</i> sequence (704×576).	207
Fig. 7.13. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Crew</i> sequence (704×576).	208
Fig. 7.14. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Bus</i> sequence (352×288).	209
Fig. 7.15. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Football</i> sequence (352×288).	209
Fig. 7.16. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Foreman</i> sequence (352×288).	210

Fig. 7.17. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Mobile</i> sequence (352×288). ..	210
Fig. 7.18. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>City</i> sequence (704×576).	211
Fig. 7.19. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Crew</i> sequence (704×576).	211
Fig. 7.20. The results of subjective assessment of the quality of <i>Bus</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	215
Fig. 7.21. The results of subjective assessment of the quality of <i>Football</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	216
Fig. 7.22. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Bus</i> sequence (352×288) ; SVC video codec.	223
Fig. 7.23. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Football</i> sequence (352×288) ; SVC video codec.	224
Fig. 7.24. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Foreman</i> sequence (352×288) ; SVC video codec.	224
Fig. 7.25. R-D curves for various techniques of inter-layer prediction of motion vectors in <i>Mobile</i> sequence (352×288) ; SVC video codec.	225
Fig. 7.26. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Bus</i> sequence (352×288); SVC video codec.	226
Fig. 7.27. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Football</i> sequence (352×288); SVC video codec.	226
Fig. 7.28. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Foreman</i> sequence (352×288); SVC video codec.	227
Fig. 7.29. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, <i>Mobile</i> sequence (352×288); SVC video codec.	227
Fig. 7.30. The results of subjective assessment of the quality of <i>Bus</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	228

Fig. 7.31. The results of subjective assessment of the quality of <i>Football</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	229
Fig. 8.1. Using of B-frames in order to achieve temporal scalability.....	237
Fig. 8.2. The structure of GOP that allows for 3 levels of temporal scalability.	238
Fig. 8.3. Motion prediction used in modified joint multiresolution coding of motion vectors in B-frames.....	239
Fig. 8.4. R-D curves for various techniques of motion vectors coding in B-frames in <i>Bus</i> sequence (352×288).....	244
Fig. 8.5. R-D curves for various techniques of motion vectors coding in B-frames in <i>Football</i> sequence (352×288).	244
Fig. 8.6. R-D curves for various techniques of motion vectors coding in B-frames in <i>Foreman</i> sequence (352×288).	245
Fig. 8.7. R-D curves for various techniques of motion vectors coding in B-frames in <i>Mobile</i> sequence (352×288).....	246
Fig. 8.8. The values of PSNR in B-frames for various techniques of motion vectors coding in <i>Bus</i> sequence (352×288).....	246
Fig. 8.9. The values of PSNR in B-frames for various techniques of motion vectors coding in <i>Football</i> sequence (352×288).	246
Fig. 8.10. The values of PSNR in B-frames for various techniques of motion vectors coding in <i>Foreman</i> sequence (352×288).	247
Fig. 8.11. The values of PSNR in B-frames for various techniques of motion vectors coding in <i>Mobile</i> sequence (352×288).	247
Fig. 8.12. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Bus</i> sequence (352×288).....	248
Fig. 8.13. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Football</i> sequence (352×288).....	249
Fig. 8.14. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Foreman</i> sequence (352×288).....	249

Fig. 8.15. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Mobile</i> sequence (352×288).	250
Fig. 8.16. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Bus</i> sequence (352×288).	251
Fig. 8.17. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Football</i> sequence (352×288).	251
Fig. 8.18. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Foreman</i> sequence (352×288).	252
Fig. 8.19. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Mobile</i> sequence (352×288).	253
Fig. 8.20. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in B-frames in <i>Bus</i> sequence (352×288).	254
Fig. 8.21. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Football</i> sequence (352×288).	254
Fig. 8.22. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Foreman</i> sequence (352×288).	255
Fig. 8.23. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in <i>Mobile</i> sequence (352×288).	255
Fig. 8.24. The results of subjective assessment of the quality of <i>Bus</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	256
Fig. 8.25. The results of subjective assessment of the quality of <i>Mobile</i> sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288).	257

List of symbols and abbreviations

2D	– two-dimensional,
3D	– three-dimensional,
4CIF	– 4:2:0 progressive video sequence with resolution 704x576 pixels of luminance component,
AHG	– Ad Hoc Group,
AMA	– Adaptive Motion Accuracy,
ASP	– Advanced Simple Profile,
B	– bitrate,
B_B	– bitrate of base layer bitstream,
B_E	– bitrate of enhancement layer bitstream,
BER	– Bit Error Rate,
B-frame	– frame coded using bidirectional (forward and backward) motion-compensated prediction,
BMA	– Block-Matching Algorithm,
CABAC	– Context-Adaptive Arithmetic Coding,
CGI	– Control-Grid Interpolation,
CIF	– Common Intermediate Format, 4:2:0 progressive video sequence with resolution 352x288 pixels of luminance component,
DCT	– Discrete Cosine Transform,
DFD	– Displaced Frame Difference,
DLP	– Digital Light Processing,
Δ_{CABAC}	– average difference between lengths of Exp-Golomb codeword and CABAC codeword for given value of motion vector residual,
Δmv	– motion vector residual,
$\Delta mv_x, \Delta mv_y$	– horizontal and vertical components of motion vector residual,
Δmv_{HL}	– differential motion vector, calculated as a difference between motion vector from high-resolution video sequence and corresponding motion vector from low-resolution video sequence,
DPCM	– Differential Pulse-Code Modulation,
DPSNR	– the differential value of PSNR,
DSCQS	– Double Stimulus Continuous Quality-Scale Method,
DVB	– Digital Video Broadcasting,
e_i	– difference between corresponding pixels in original picture and distorted one,
ϵ_{N+1}	– prediction residual signal in motion-compensated prediction,
FIR	– Finite Impulse Response,
FMO	– Flexible Macroblock Ordering,

fps	– frames per second,
GMC	– Global Motion Compensation,
GPRS	– General Packet Radio Service,
H	– vertical dimension of a video frame (height),
$H_0..H_4$	– weighting arrays used in OMC,
HDD	– Hard Disc Drive,
HDTV	– High Definition Television,
h_{HL}	– mutual matching parameter, describes matching of motion vector fields from low-resolution video sequence and high-resolution video sequence,
I	–array containing luminance samples of a video frame, original video frame,
\tilde{I}	– prediction of a video frame in motion-compensated prediction,
\hat{I}	– reconstructed video frame in motion-compensated prediction,
I-frame	– frame coded without reference to any picture except itself,
IILP	– Implicit Inter-Layer Prediction,
IPTV	– Internet Protocol Television
I_x, I_y, I_t	– partial derivatives of luminance in horizontal, vertical and temporal directions respectively,
JVT	– Joint Video Team, established by VCEG and MPEG,
kbps	– kilobit per second,
λ	– optical flow field smoothness parameter,
LAN	– Local Area Network,
LCD	– Liquid Crystal Display,
MBP	– Minimum Bitrate Prediction,
MCTF	– Motion-Compensated Temporal Filtering,
$med()$	– median operator,
MOS	– Mean Opinion Score,
MPEG	– Moving Pictures Experts Group,
mv	– motion vector,
mv_0, mv_C	– motion vector of the currently coded block,
mv_B	– backward motion vector, a vector used in backward motion-compensated prediction,
mv_B	– motion vector from base layer of scalable codec,
mv_E	– motion vector from enhancement layer of scalable codec,
mv_F	– forward motion vector, a vector used in forward motion-compensated prediction,
mv_P	– predicted motion vector,

MVQ	– Motion Vector Quantization,
mv_x, mv_y	– horizontal and vertical component of motion vector, respectively,
OMC	– Overlapped Motion Compensation,
p	– prediction signal,
$p_0..p_4$	– intermediate prediction arrays used in OMC,
PDA	– Personal Digital Assistant,
P-frame	– frame coded using unidirectional (forward) motion-compensated prediction,
PSNR	– Peak Signal-to-Noise Ratio,
QCIF	– Quarter Common Intermediate Format, 4:2:0 progressive video sequence with resolution 176x144 pixels of luminance component,
Q_P	– quantization parameter, the parameter that controls the value of quantizer in hybrid video coding; its value influences the quality and the bitrate of compressed video sequence,
R-D	– rate-distortion,
SDTV	– Standard Definition Television,
σ_{HL}	– standard deviation of differential motion vector field (see Δmv_{HL}),
SSCQE	– Single Stimulus Continuous Quality Evaluation,
SVC	– Scalable Video Coding,
SVM	– Scalable Video Model,
t_D	– decoding time,
t_E	– encoding time,
TMC	– Triangle Motion Compensation,
UVLC	– Universal Variable-Length Codes,
v	– optical flow vector,
VBSMC	– Variable Block Size Motion Compensation,
VCEG	– Video Coding Experts Group,
VLC	– Variable Length Codes,
VOP	– Video Object Plane,
VQ	– Vector Quantization,
W	– horizontal dimension of a video frame (width).

Abstract

The dissertation deals with the problem of representation of motion vectors in scalable video codecs. Motion model and existing techniques of representation of motion vectors in advanced hybrid video coders are thoroughly discussed. The problem of multiresolution motion vectors estimation and motion vectors coding in scalable video codec is stated. The possible solutions are presented. Similarities and correlations in multiresolution motion vector fields are researched.

Author proposes several techniques of multiresolution motion vector coding, including joint multiresolution representation and inter-layer prediction. The very fast and simple mode of motion vector representation in temporally scalable codec is also presented in the thesis.

Proposed algorithms have been experimentally tested and compared against other methods. Obtained results are presented in this dissertation.

Streszczenie

Rozprawa dotyczy problemu reprezentacji wektorów ruchu w skalowalnych kodekach wizyjnych. W pracy, omówiono model ruchu i istniejące techniki reprezentacji wektorów ruchu w zaawansowanych hybrydowych koderach wizyjnych. Sformułowany został problem wielorozdzielczościowej estymacji wektorów ruchu oraz ich kodowania w koderze skalowalnym. Przedstawiono możliwe rozwiązania tego problemu. Przebadano podobieństwa i korelacje występujące w wielorozdzielczościowych polach wektorów ruchu.

Autor zaprezentował kilka technik wielorozdzielczościowego kodowania wektorów ruchu, takich jak łączna reprezentacja wielorozdzielczościowa czy też międzywarstwowa predykcja wektorów ruchu. Zaprezentowano również bardzo szybki i prosty tryb reprezentacji wektorów ruchu w skalowalnym koderze wizyjnym ze skalowalnością czasową.

Zaproponowane algorytmy zostały sprawdzone eksperymentalnie i porównane z innymi używanymi metodami. W rozprawie przedstawiono dokładne rezultaty eksperymentów.

Chapter 1

Introduction

1.1. Multiresolution video representation

In 1929 Ray Kell, the future head of television research at RCA Laboratories, patented the idea of video compression in temporal domain using conditional update of changeable areas of consecutive video frames [Rea02, Web05, Ced06]. Although it is not sure whether he managed to implement the proposed technique, the idea of sending through a transmission channel only the variable part of video sequence is the fundamental of today's digital video representation.

Since its beginnings, video compression developed rapidly. Many techniques and algorithms of video compression have been proposed over the time. The first generation of the video codecs involved algorithms of intra-frame coding and simple inter-frame coding techniques [Ska98, Gha98, Dom05]. Along with the progress in science, new, more sophisticated tools and techniques have been proposed in order to achieve better compression of video. Among others, motion-compensated predictive coding proved to be extremely efficient in compression of video signals.

New, advanced techniques and tools of video coding have been proposed recently. They resulted in developing the state-of-the-art video codecs which are often called advanced video codecs, such as AVC/H.264 [ISO06], VC-1 [SMP05] or AVS [AVS06]. They significantly outperform previous algorithms of video coding in terms of compression efficiency [Rib03, Ric03, Oel04, Fan04].

A very important issue is standardization of video coders [Sch95]. In many cases, encoders and decoders are offered by different vendors. Moreover, encoder and decoder usually operate far away from each other. Thanks to video coding standards, they can interact properly. The development of video coding standards is a reflection of scientific progress. Motion-compensated prediction has been exploited in all recent video coding standards such as MPEG-2, H.263 or MPEG-4 [Hoa02, Wan02]. These standards were successfully introduced into industry and are widely used for compression of video content.

Advanced video codec AVC/H.264 was established as international standard in 2003 [Wie03]. However, Version 1 of AVC/H.264 did not support scalable video coding that is currently considered as an important functionality for many applications [Ohm01, Dom03].

A scalable bitstream allows for representation of a video sequence with various spatial and temporal resolutions. A low-resolution bitstream is embedded in a high-resolution bitstream. In order to obtain low-resolution video it is not needed to decode the whole bitstream [Nav94]. With the high resolution one can achieve more detailed video representation, while the lower resolutions reduce the amount of data that forms sampled video.

Many techniques were proposed for multiresolution video compression, including wavelet coding and hierarchical hybrid video coding. Former video codecs allow for multiresolution video representation: in MPEG-2 certain profiles introduce spatial and temporal scalability [ISO94], annex O in H.263 recommendation permits for spatial and temporal scalability as well. However, these extensions of existing techniques were never widely used because of its high complexity and high scalability overhead [Dom04].

At the beginning of 21st century the video science community started to work towards the development of a new standard for multiresolution video representation [MP02-35, MP03-25, MP03-93]. While existing tools and techniques are well-suited for common video sequences with a single spatial and temporal resolution, they are not efficient enough for compression of a material with various resolutions. On the other hand, recently, the heterogeneous telecommunication networks have spread rapidly. The maximum transmission speed often varies within a network and there is a need to ensure an appropriate technology for video compression in such a heterogeneous environment.

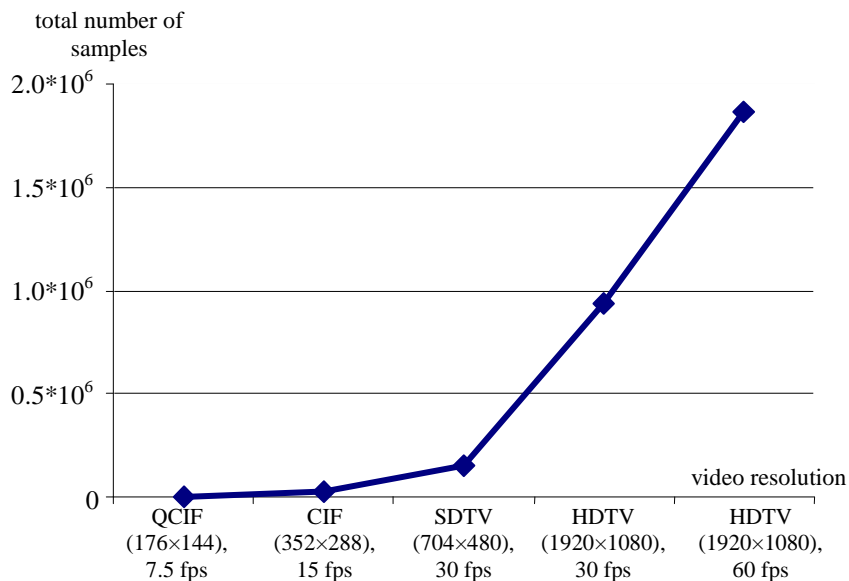


Fig. 1.1. The resolution of a video sequence and total number of luminance and chrominance samples for YUV 4:2:0 sampling scheme.

Users need a wide variety of video resolutions for specific purposes: low spatial resolution for small screen of Personal Digital Assistant (PDA) and cellular phone, standard television resolution for a terrestrial television and very high resolutions for High Definition Television (HDTV) and digital cinema. There are around 81 times more pixels in a single video frame of the high-definition television than in PDA-resolution format (Fig. 1.1). It is a real challenge for telecommunication systems to process, transmit and store such a variety of video signals.

Furthermore, the same video sequence often should be transmitted through various transmission channels to clients that use various receivers capable of displaying the specific resolution of the video [Wan02]. The same content is delivered through different protocols, networks and to consumers with various terminal types as depicted in Fig. 1.2.

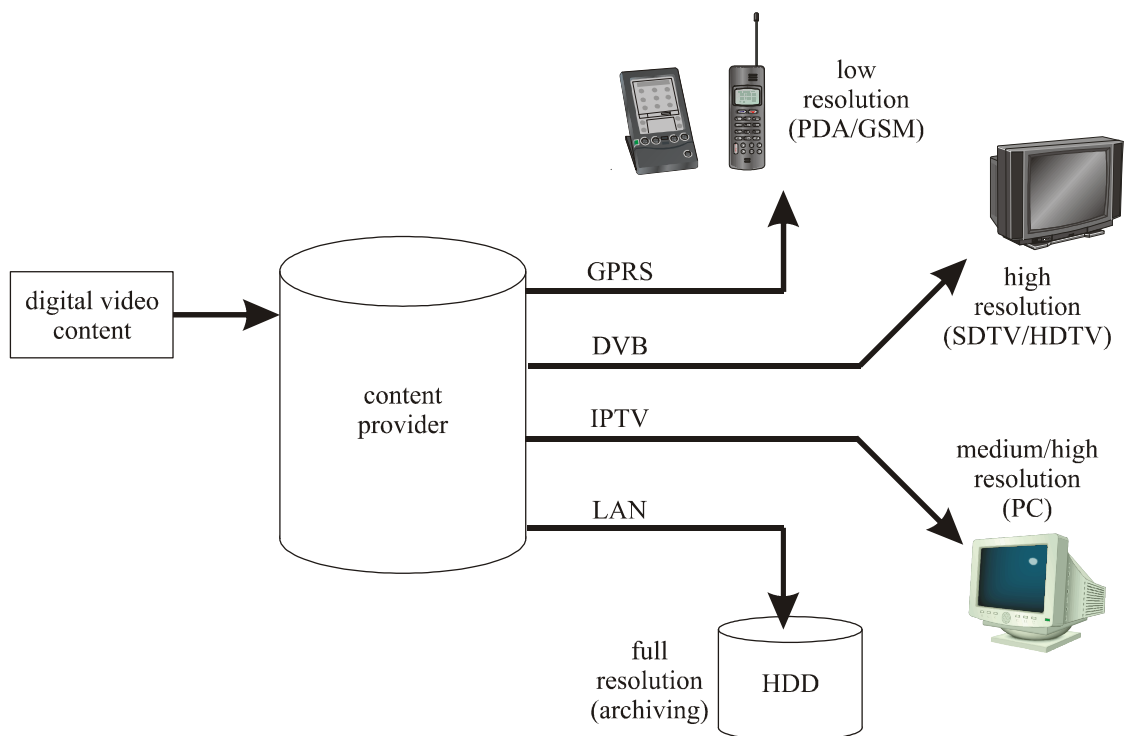


Fig. 1.2. Multiresolution video transmission over heterogeneous networks and protocols.

In order to be stored or transmitted, video data have to be compressed by video encoder, usually using lossy compression scheme [Tek95, Dom98, Ohm04]. When multiresolution representation of video sequence is considered, it is possible to process a video signal by using a set of video encoders [Wei99]. The original video sequence is spatially or temporally downsampled and then compressed many times in order to

achieve multiple bitstreams containing video of various resolutions, as depicted in Fig. 1.3. Such a technique is called *simulcast* [Mac02].

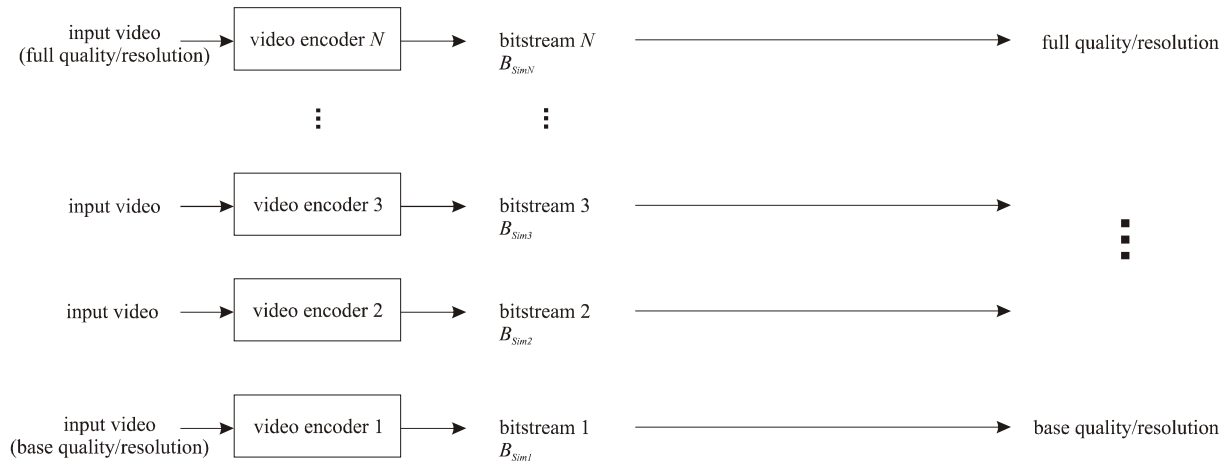


Fig. 1.3. *Simulcast* video coding.

The simulcast approach allows for flexible choosing of the encoding algorithm, resolution of the output video or the target bitrate for each resolution. However, simulcast technique is not efficient enough, because a video sequence is encoded independently for each resolution. In simulcast, the overall bitrate B_{SimT} is a sum of bitrates of each individual bitstream with the specific resolution ($B_{Sim1}, B_{Sim2}, \dots$) (1.1):

$$B_{SimT} = B_{Sim1} + B_{Sim2} + \dots + B_{SimN} . \quad (1.1)$$

While there are many similarities between the same video sequences represented with different resolutions, the simulcast-approach does not allow for exploiting them in order to achieve higher compression ratio. Another disadvantage of simulcast technique is multi-pass encoding: the source sequence has to be encoded as many times as many video resolutions are required.

The other approach, which is much more efficient for multiresolution video representation, is scalable video coding (Fig. 1.4).

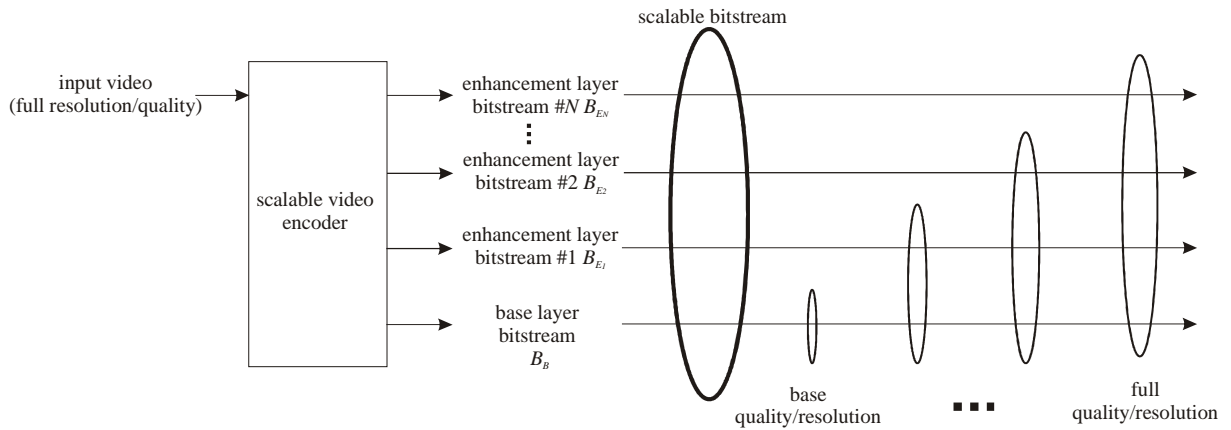


Fig. 1.4. Scalable video coding.

Scalable video encoding provides the ability to produce layered data bitstream [Ohm01, Dom03]. Layers of a scalable bitstream allow for decoding of a video sequence with varying quality (spatial resolution, temporal resolution or signal-to-noise ratio). In other words, a scalable video bitstream is a bitstream that can be partially decoded with reduced video resolution or quality [MP03-25]. The full resolution or full quality can be achieved only when the whole bitstream is properly received and decoded. An access to the appropriate, limited part of a bitstream – called a base layer – enables decoding of a video sequence with only basic quality. The base-layer bitstream is embedded in the total bitstream.

Scalable video coding proved to be very efficient for obtaining multiresolution video representation [Pur94, Ill97, MP04-37]. Layers of the scalable bitstream encode data for particular temporal or spatial resolutions. The idea of multiresolution video representation by the use of scalable video coding has been depicted in Fig. 1.5.

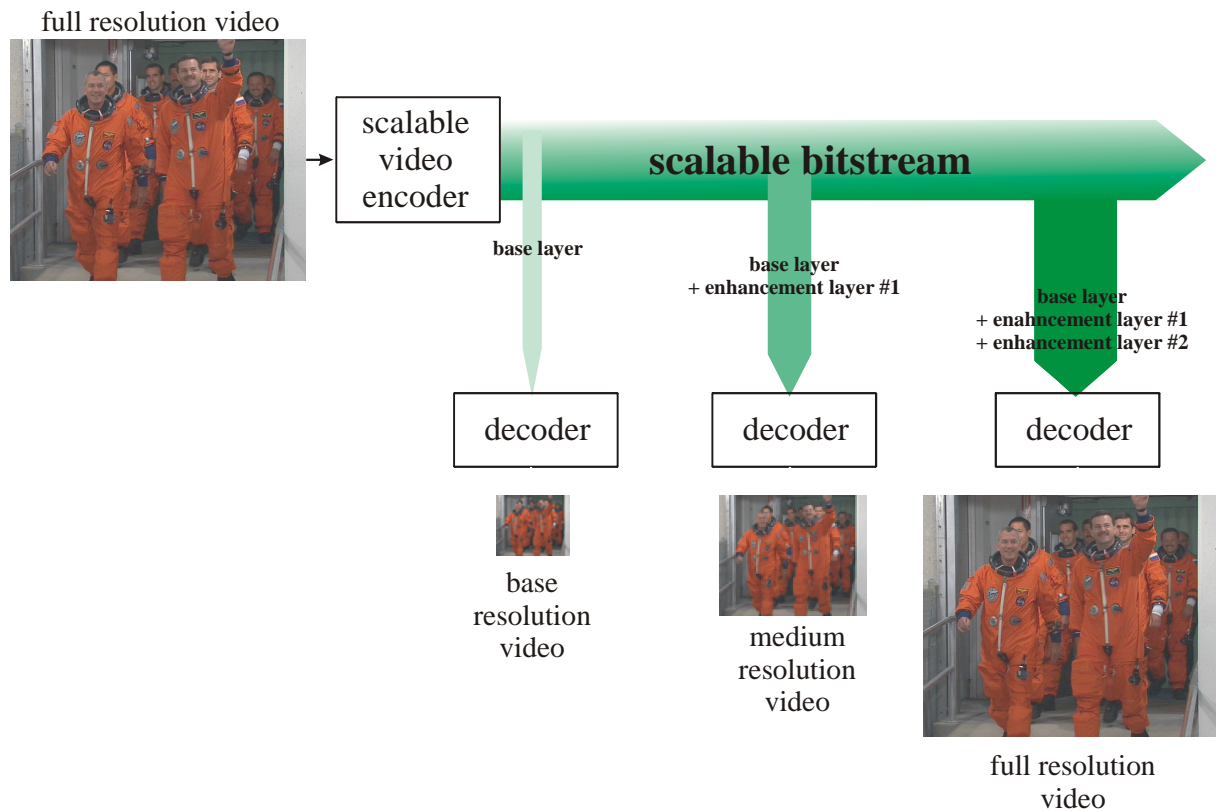


Fig. 1.5. Multiresolution video representation using scalable video coding.

Important application of scalable video coding is video transmission in error-prone environments, such as wireless networks or the Internet. Scalable coding scheme with layered approach can be used for video streaming with unequal error protection [Gir95, Gal01]. The base-layer bitstream is better protected with appropriate algorithms and the bitstreams of the higher layers are worse protected. As a result, under the erroneous transmission conditions, the quality of the decoded video decreases slower than in the case of equal error protection and single layer approach, as depicted in Fig. 1.6.

Scalable video coding introduces analogue-like transmission feature into digital transmission: increasing bit error rate (BER) causes the degradation of quality of video, but still allows for access to the video content with reduced quality [Dom04].

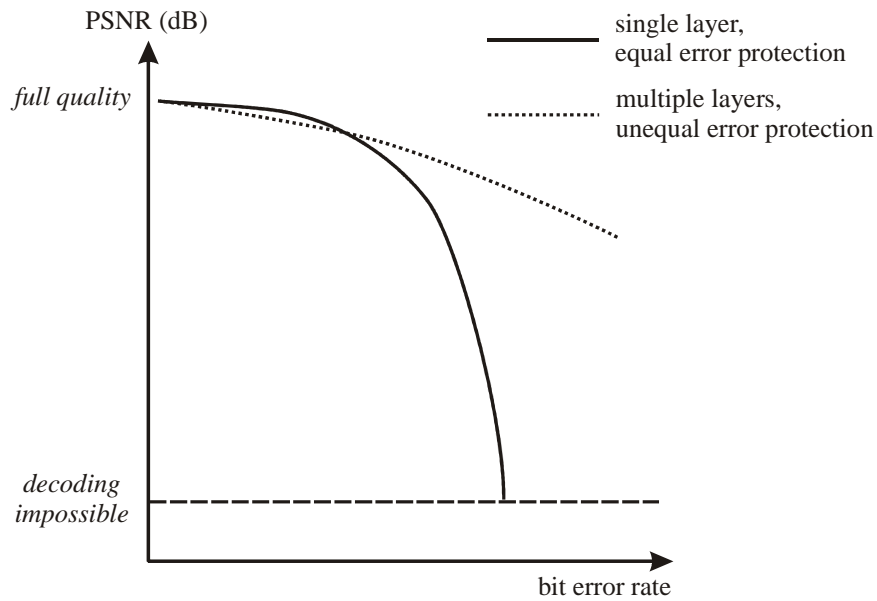


Fig. 1.6. Comparison of quality decrease. Single-layer coding versus scalable coding with unequal error protection [Gir95].

Scalable video coding permits to encode input video at the highest resolution once, and enables to decode a portion of the bitstream depending on specific resolution or rate that is required. This allows for simple and flexible transmission over heterogeneous networks and provides adaptability of systems for bandwidth variations and error conditions [Ohm05]. The potential applications of scalable video coding are [Gha02, Sma02]:

- video telecommunications in heterogeneous networks,
- video-surveillance systems,
- video database browsing,
- multiresolution playback of video in multimedia environments,
- multicasting of video over the internet,
- video service hierarchies with multiple spatial, temporal and quality resolutions,
- high-definition television (HDTV) with embedded standard-definition television (SDTV).

Multiresolution video representation has been recognized as extremely useful functionality. Therefore scalability is one of the most important challenges of video compression [Dom04].

1.2. Motion vectors in hybrid video coding

Natural video sequences contain high level of spatial and temporal redundancies. In video compression, they are removed with almost negligible impact on subjective quality of the output video. The most successful and wide-spread class of video encoders use motion-compensated prediction and coding of prediction residuals in order to achieve video compression. They are commonly called hybrid codecs [Dom98, Shi00].

In order to perform motion-compensated prediction, a video frame is divided into small regions, usually being rectangular blocks. Motion vectors are estimated at encoder's side for each block and then, they are transmitted to decoder using bitstream. A bitstream produced by a typical hybrid video codec consists of sub-bitstreams of motion vectors, transform coefficients of prediction error and control data [Ric02].

Motion vector fields, obtained with popular algorithms of motion estimation, are highly spatially correlated [Bar94, Li94, Kri97a]. In order to exploit correlations between neighboring vectors, motion vectors are differentially encoded. This technique reduces significantly the amount of data that has to be transmitted. The techniques of residual representation of motion vectors have been widely adopted in video coding algorithms [ISO93, ISO95, ISO98, ITU05, ISO06].

In the very first approaches, the components of motion vectors were coded differentially using simple prediction schemes with respect of the lastly encoded motion vector. This technique has been further improved in more advanced algorithms of video coding –median prediction of motion vector has been introduced. For each component of a motion vector the prediction signal is formed by median filtering of the motion vector components of neighboring macroblocks.

Median prediction exploits spatial correlation between motion vectors better than first-order prediction using single motion vector. Therefore, prediction residual is reduced and can be represented in a bitstream with less number of bits.

1.3. Goals and thesis of the work

To date, there are many proposals of scalable video coders, including hierarchical hybrid scalable coders and scalable wavelet coders. Some of them use Motion-Compensated Temporal Filtering (MCTF) in order to achieve temporal scalability.

However, there is still no ultimate solution for representation of motion vectors in multiresolution video coding.

The goal of this work is to propose new techniques for representation of motion data for multiresolution video compression. The researched algorithms should give better compression efficiency with the lowest possible increase of complexity as compared to present algorithms. Alternatively, modifications of existing tools and techniques are to be proposed in order to decrease their complexity with possible minor impact on compression efficiency.

Algorithms for spatial and temporal scalability have been already researched in video coding. However, the correlations between motion vectors of video sequence with different spatiotemporal resolutions are not exploited in formerly standardized algorithms. The most recent Advanced Video Coding algorithm (AVC/H.264) [ISO06], which is described by an international standard, in its original version had not supported multiresolution video representation.

In 2003, Moving Picture Experts Group (MPEG) of International Organization for Standardization (ISO) began works on new techniques and tools for scalable video coding that should allow for representations of video sequence with various spatiotemporal resolutions. The author of this thesis actively joined the MPEG team in order to work on new algorithms and methods for more efficient encoding of motion vectors in scalable video codec. As result of research, a number of documents and reports were contributed to MPEG and Joint Video Team (JVT). Some ideas presented in this dissertation were incorporated into MPEG works. The techniques of multiresolution motion vector representation were adopted into arising standard of scalable video coding.

The following assumptions are made in this dissertation:

- motion compensated inter-frame coding technique is used as a basic video coding algorithm,
- the proposed technique should assure possibly high compatibility with existing tools and techniques,
- no critical requirements for memory and computational complexity are introduced into the existing algorithms.

The **main theses** of the dissertation are:

- There is the implicit correlation between motion vectors estimated for different resolutions of the same video sequence.
- By exploiting these inter-resolution correlations in motion vector fields, it is possible to improve efficiency of representation of motion data in scalable video coding.
- It is possible to develop techniques of multiresolution representation of motion vectors that are competitive to the methods described in literature, developed simultaneously with the author's investigations.

The **particular goal** of the thesis is to extend techniques of motion vector representation for scalable video coding. New algorithms and tools are to be researched in order to improve overall coding efficiency. The complexity of proposed methods should be the same or lower than the complexity of existing methods. New techniques of motion vector representation should decrease the motion vector residuals with minor impact on complexity and requirements of codecs.

1.4. Research methodology

The starting point for research was related to the existing techniques of representation of motion vectors in non-scalable video coders, with the special attention to the most recent and the most advanced solution used in AVC/H.264 video codec. These techniques have been thoroughly analyzed and their efficiency for motion vectors encoding has been examined and experimentally tested. A problem of motion vector representation in multiresolution coding of video sequences has been formulated. The existing techniques of motion vector representation in non-scalable, single-resolution video coding have been applied into scalable video codec, which produces layered bitstream with multiresolution representation of video sequence.

These existing techniques of single-resolution representations of motion vectors, implemented in scalable video codecs were further developed and improved. New techniques have been proposed as well. These developed methods have been experimentally tested and researched in order to check their usefulness in further algorithms of multiresolution video compression.

As the reference anchor, the state-of-the-art AVC/H.264 video compression algorithm has been used. Two scalable video codecs were used during the experimental verification of research:

- AVC-based video codec, developed originally at Poznań University of Technology, and
- SVC video codec, which has been developed by JVT committee as a future standard for scalable video compression.

The first scalable video codec that was the basis for modifications had been built using AVC/H.264 reference software version 7.3, which is freely available at [ISO06a]. This scalable codec was developed as the answer for MPEG's "Call for Proposals on Scalable Video Coding Technology" [MP03-93]. The codec is briefly presented in Section 2.4.2 and is thoroughly described in [Bła04a, Bła04b].

The second scalable video codec, SVC (Scalable Video Coding) codec was developed later than the codec developed at Poznań University of Technology. When this dissertation was written, the SVC codec was still in development [JVT06-02]. Version 4.0 was used by the author in experimental research. The reference software of this codec is freely available for developers at [ISO06b].

These two codecs have been chosen because the author had free access to their source code, so that modifications could be introduced in their algorithms. Another state-of-the art video codec VC-1 does not allow for scalable representation of video sequence. Furthermore, its reference software is not available freely, thus it could not be used in the experiments.

Efficiency of motion vector coding and efficiency of overall compression have been examined: the existing techniques of motion vector encoding have been compared against the original solutions proposed in the dissertation. For this purposes, residual of motion vector prediction, as well as rate and distortion have been measured for various methods of encoding of motion vectors in tested video codecs. For measuring the distortions, objective quality measure PSNR have been chosen, as discussed in the following section. Additionally, subjective tests for evaluation of the quality of encoded video sequences have been also performed in some cases.

In order to determine the complexity of proposals and compare it with the complexity of existing algorithms, the execution times of the researched video codecs have been measured.

In all experiments, standard video test sequences have been used. These test sequences contain various types of motion and textures. They were chosen among others by JVT and MPEG in order to perform comparisons and experiments during developing of new tools and techniques for scalable video compression [MP03-93]. The bitrate ranges of compressed video sequences were also chosen to meet the requirements announced by MPEG organization during comparison of scalable video coders [MP03-93, Bar04].

1.5. Quality measurement

The ability to measure distortion in image is needed for comparison of efficiency of the different video coding algorithms. However, the perceptual feelings about distortions in visual content are difficult to measure because of the complexity of the human visual system [Sul98]. What is more, the addition of temporal dependencies in consecutive pictures of a video sequence introduces further problems in measuring the perceptual quality.

There are two classes of methods for assessment of video quality:

- subjective quality evaluation by a panel of viewers [ITU94, ITU03, Win05],
- objective quality evaluation using traditional signal distortions measures [Oja03, Win05].

Moreover, there exist techniques aimed at automatic assessment of video quality that produce results highly correlated with results of subjective tests [Xin99, Pas06]. Nevertheless, these methods still have limited scope of applications, therefore they are not used in assessment of new coding algorithms.

In the recommendation of International Telecommunication Union BT.500-11, two classes of subjective methods of quality evaluation are defined: Double Stimulus and Single Stimulus techniques. For example, in Double Stimulus Continuous Quality-Scale Method (DSCQS) the distorted video sequence is compared against the original video sequence. On the other hand, in Single Stimulus Continuous Quality Evaluation (SSCQE) method video sequence is assessed without comparison to the original video. In both methods viewers vote using handset slider with continuous quality scale. Additionally, the variants of the DSCQS and SSCQE methods with discrete scale of 5 grades are also described in the recommendation [ITU03].

Another subjective quality assessment technique – Single Stimulus MultiMedia (SSMM) [Bar04] – has been derived from the Single Stimulus method and the variation of the Single Stimulus method with two repetitions defined in the recommendation BT.500-11 [ITU03, Bla06]. In the SSMM method, judged video sequences are displayed in alternate order on progressively scanned display (LCD display, CRT computer display, LCD or DLP projector). Each video sequence is displayed for 10 seconds, and then the 5-seconds break is given for judging. The assessment scale has 11 grades, as depicted in Fig. 1.7.

In order to avoid the “Contextual effect” (subjective feelings about the quality of the video sequence depend on the quality of a previous video sequence), in SSMM method each video sequence is displayed twice in a different order. The mean grade is then calculated from the two received votes for each tested video sequence.

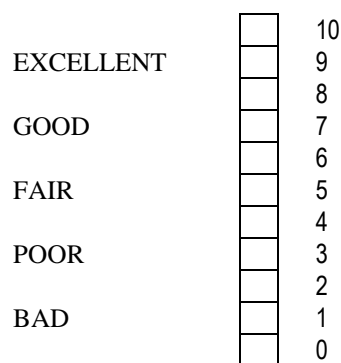


Fig. 1.7. The voting scale of Single Stimulus MultiMedia method.

In this dissertation, the SSMM method of quality evaluation has been used in the experiments depicted in sections 7.5, 7.8 and 8.4. The tests were performed in a dark room, using LCD projector NEC VT770. The panel of 19 observers at the age of 22-28 took part in the experiments. The viewers were not specialists in video coding. The height of the screen was 1.5 meters and the width of the screen was 2 meters. The viewing distance of 3 to 4 times of the screen height was preserved for CIF sequences [Bla06].

The most often used objective measure for comparison of the quality of visual content is peak-signal-to-noise ratio (PSNR) [Dom98, Win05], which is defined by:

$$PSNR[dB] = -10 \log_{10} \left(\frac{\sum_i e_i^2}{N \cdot 255^2} \right) \quad (1.2)$$

where:

- e_i – difference between corresponding pixels in original picture and distorted one,
- N – total number of samples in a picture,
- 255 – magnitude of a sample value, related do the dynamic range (8-bit representation).

The value of PSNR is often measured and compared only for the luminance component of the image, as the distortions in the chrominance components of the picture are less visible, “the chrominance components are often treated as something of a minor nuisance in video coding” [Sul98].

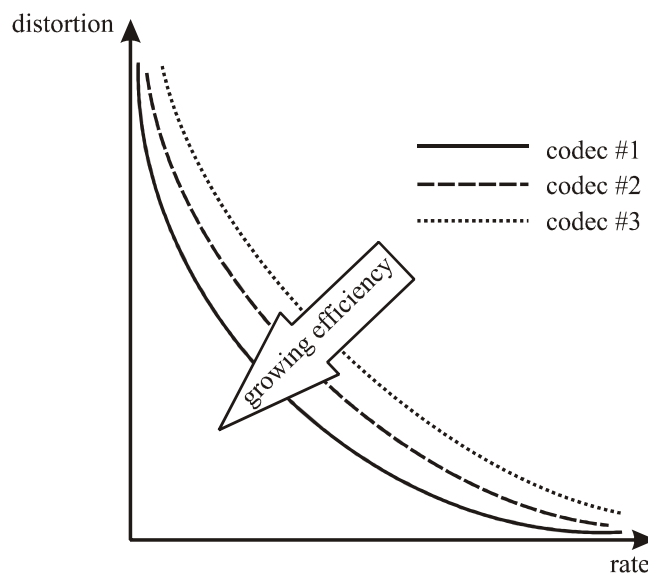


Fig. 1.8. Hypothetical rate-distortions curves.

Efficiency of a video coder is described by rate-distortion (R-D) curve [Ram94, Wie96, Ort98, Ska98, Shi00, Ohm05]. The R-D curve is obtained by plotting the distortion measure (e.g. PSNR) achieved by specific codec for each tested bitrate. Examples of hypothetical R-D curves are given in Fig. 1.8.

The subjective quality evaluation is expensive and difficult to carry out. It requires a number of observers, specialist equipment and a lot of tests. On the other hand, the differences in achieved video quality are often very small, high accuracy and a fine grain of the scale is needed in order to assess the quality of video sequence. Because of these reasons, in this dissertation, an objective measure – PSNR – has been chosen most often for quality evaluation. The subjective method of quality evaluation – SSMM – has been used in a few sections of this thesis in order to verify the results of quality assessment obtained by the PSNR calculation.

1.6. Thesis overview

The thesis is organized as follows: in Chapter 2 we shortly discuss the problem of scalable representation of video sequences. Hybrid video coder is described in detail; the paradigm of motion-compensated prediction and algorithms of motion estimation are also described.

Chapter 3 contains detailed description of techniques used in motion-compensated prediction in video coders. The motion model used in video coders is thoroughly discussed in this chapter as well as methods of representation of motion vectors in a bitstream.

In Chapter 4 the most advanced techniques of motion vector representation are described. Methods of prediction and entropy coding of motion vectors in the state-of-the-art AVC/H.264 video codec are discussed and explained. Experimental results are presented regarding the efficiency of existing techniques of non-scalable encoding of motion vectors. New methods of motion vectors encoding using vector median prediction are proposed and tested also in Chapter 4. The efficiency of adaptive entropy coding of motion vectors is experimentally researched.

Chapter 5 discusses correlations in multiresolution motion fields. The problem of the multiresolution motion vectors estimation and representation is formulated also in this chapter. Measures of correlations of motion vector fields are proposed. Experimental results of examining of multiresolution motion fields are presented in Chapter 5.

In Chapter 6 the original author's method of joint multiresolution motion estimation and differential motion vector encoding is briefly presented together with experimental results achieved by proposed scalable codec.

Chapter 7 contains a description of the IILP technique – an original technique of inter-layer prediction of motion vector in layered scalable video coder. Other techniques of inter-layer motion representation that were proposed lastly are also presented together with comparison of author's technique against later-proposed algorithms.

In Chapter 8 a technique of motion vectors derivation for temporally scalable video codec is presented. The efficiency and complexity of proposed method is discussed in comparison to other algorithms.

Chapter 9 contains a summary of achieved results and conclusions.

Chapter 2.

Video coding with motion-compensated prediction

2.1. Hybrid video coding

Temporal redundancies in video sequences are eliminated by the use of prediction with motion compensation. The very first idea behind motion-compensated prediction was temporal DPCM coding of subsequent video frames that was proposed for analogue television in 1971 in the papers of Limb and Pease [Lim71] and Candy et al. [Can71]. The idea of inter-frame coding was developed further by introducing motion-compensated prediction that highly reduces energy of the prediction residuals [Tek95, Dom98].

Motion-compensated prediction of video content is used in hybrid video coders together with transform encoding of prediction residuals. At present, this is the most often used class of video coding techniques [Tek95, Ska98, Sad02]. A block diagram of a typical modern hybrid video encoder with motion-compensated prediction is given in Fig. 2.1.

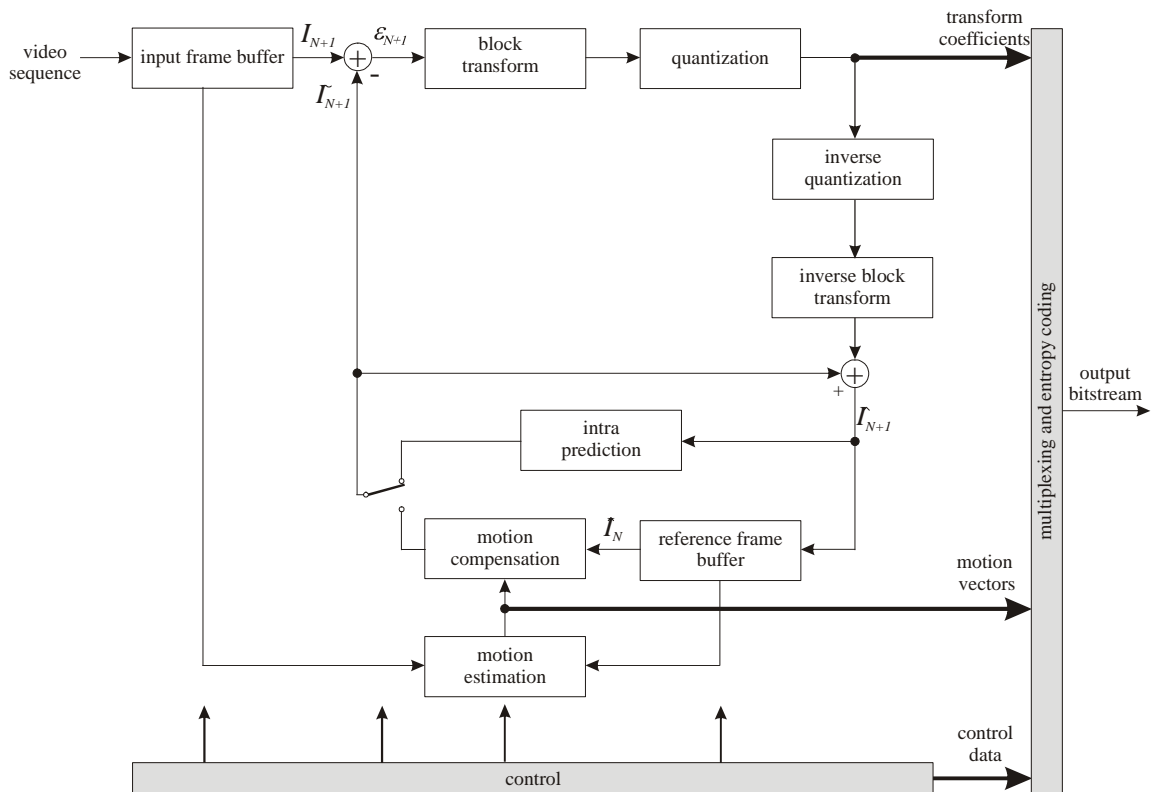


Fig. 2.1. Hybrid video encoder with motion compensated prediction.

A residual signal ϵ_{N+1} is obtained by subtracting a prediction of current frame \tilde{I}_{N+1} from the actual values of samples I_{N+1} . This residual signal is coded using lossy

compression scheme with transform coding and quantization of transform coefficients. Quantized values of transform coefficients are then encoded using entropy coding and form a bitstream.

In the reconstruction loop of the encoder, inverse rescaling and inverse transform are performed. Reconstructed residual information is added to the prediction signal \tilde{I}_{N+1} and forms final reconstruction of the current frame \hat{I}_{N+1} , which can be used further in inter-frame prediction process. Finally the reconstructed frame is stored in the reference frame buffer, thus it can be used as a prediction for subsequent frames.

Prediction signal \tilde{I}_{N+1} is obtained using spatial prediction from a reconstructed current frame (\hat{I}_{N+1}) or using motion-compensated prediction from a previous frame \hat{I}_N as depicted in Fig. 2.1.

Hybrid video coding has evolved over years. The first generation of hybrid video coders employed only simple inter-frame prediction techniques and was never widely used. The progress in science allowed for introducing more sophisticated algorithms, tools and functionalities, which improved the efficiency of video compression in the second and the third generations of video codecs [Dom05, Ohm04]. Representatives of these groups are video coding standards H.261, MPEG-2 or H.263. They have been widely used in telecommunication, multimedia and digital television.

The most efficient and advanced coding algorithms have been developed recently and they outperform previous techniques.

The following features allow for significant reduction of bitrate at the same video quality in forthcoming generation of video codecs:

- new algorithms of intra prediction,
- in-loop deblocking filtering,
- complex and accurate motion model,
- multi-hypothesis prediction,
- advanced motion vector representation,
- context-based entropy coding,
- arithmetic entropy coding.

These advanced tools have been incorporated into the state-of-the-art video codecs like AVC/H.264 [ISO06], VC-1 [SMP05] or AVS [AVS06] codecs. They are often referred to as advanced video coders.

In advanced hybrid video coders input frames of the video sequence are buffered. The order of encoded frames can be reversed. As a result, advanced video coders use forward prediction (prediction from the past), backward prediction (prediction from the future) and bidirectional prediction for decreasing the energy of prediction residuals [Str96]. Frames which are coded with motion-compensated prediction are customarily called P-frames (*predictive*) and B-frames (*bi-directional* or *bi-predictive*). Forward prediction is used in P-frames, while forward, backward and bidirectional prediction is used in B-frames. Efficient inter-frame coding with motion compensation requires accurate estimation of displacement field between two frames [Fli04]. This has been widely described in Section 3.2.2.

On the other hand, the first frame of a sequence has to be encoded independently of other frames, because there is no frame to predict from. In such a case, in advanced video coders, spatial prediction of samples is performed. This technique that is called intra prediction, significantly improves coding efficiency. Frames which are encoded using only intra prediction are called I-frames. Beside the first frame of video sequence, I-frames are used also as random access points to the bitstream.

Advanced video coders may use multiple reference frames for motion compensated prediction. Therefore, motion data that have to be represented in the bitstream consist of forward and backward motion vectors and indices of reference frames that were used for prediction of samples.

In order to decode completely a video frame, a decoder needs a reference frame index, motion vector and transform coefficients of the prediction residual. Therefore, the following information has to be encoded in a bitstream in order to allow for complete reconstruction of video sequence:

- control data (sequence resolution, prediction modes, partitioning, etc.),
- motion data (motion vector, reference frames),
- transform coefficients of prediction residuals.

In Chapter 3, more detailed description is given for motion models and motion data in advanced hybrid video codecs.

2.2. The paradigm of motion-compensated prediction

Temporal redundancies are removed from a video sequence using motion-compensated prediction. Subsequent frames in video sequence are temporally predicted using previously encoded frames, according to the formula:

$$\tilde{I}_{N+1}(x, y) = \hat{I}_N(x + mv_x, y + mv_y), \quad (2.1)$$

where:

- \tilde{I}_{N+1} – prediction of the current frame,
- \hat{I}_N – reconstructed previous frame (reference frame),
- mv_x, mv_y – components of the motion vector (horizontal and vertical respectively) calculated for a given location.

Motion-compensated prediction residual called *displaced frame difference* (DFD) is calculated:

$$DFD_{N+1}(x, y) = I_{N+1}(x, y) - \tilde{I}_{N+1}(x, y), \quad (2.2)$$

where I_{N+1} denotes the original frame. DFD is then encoded, usually using lossy scheme with DCT-based transformation. The final reconstruction of the current frame is obtained according to the equation:

$$\hat{I}_{N+1}(x, y) = \tilde{I}_{N+1}(x, y) + \hat{DFD}_{N+1}(x, y), \quad (2.3)$$

where:

- \hat{I}_{N+1} – reconstructed current frame,
- \hat{DFD}_{N+1} – reconstructed prediction residual.

In order to perform motion-compensated prediction, motion vector components mv_x and mv_y have to be estimated at encoder's side and transmitted to the decoder. The process of searching for motion vectors is called motion estimation.

2.3. Motion estimation

2.3.1. Block-matching algorithm of motion estimation

In video compression, the most widely used scheme of motion estimation is block matching algorithm (*BMA*) [Jai81, Tek95, Ska98, Sad02]. In block-matching approach, motion vectors for video frame are calculated in reference to previously encoded frame called a reference frame. An input frame is divided into rectangular blocks. For each block, the algorithm finds matching block – a block in the reference frame that matches the current block best. The match is estimated by maximizing the criterion of similarity between blocks or – equivalently – by minimizing the criterion of distortion between blocks [Kri97a, Dom98]. In other words, block matching algorithm finds a displacement value (motion vector) for each square block of pixels that minimizes prediction error of a block, as depicted in Fig. 2.2. In order to minimize the complexity of the algorithm, usually matching block is searched over a limited area of reference frame (search region in Fig. 2.2). Prediction with motion compensation is performed for each block of a video frame using estimated motion vector.

In early video codecs motion compensated prediction was applied for blocks of 8×8 , 8×16 or 16×16 luminance samples [Kog81, Nin82, CCI84, Eri85]. In 1987 the term of macroblock that consists of 16×16 luminance samples was introduced for the first time [CCI87]. To date, a macroblock is the basic fragment of the video frame used in most of the existing video coding algorithms. For every macroblock which is coded using motion compensated prediction at least one motion vector is sent [ISO93, ISO95, ITU05].

Application of motion-compensated prediction with variable-size blocks further improves the efficiency of video compression [Fli04]. Such an approach has been described more thoroughly in Section 3.2.1.1. The most advanced algorithms of video compression utilize complex motion model with blocks of variable size: from 16×16 to 4×4 luminance samples [ISO06] or from 16×16 to 8×8 luminance samples [SMP05].

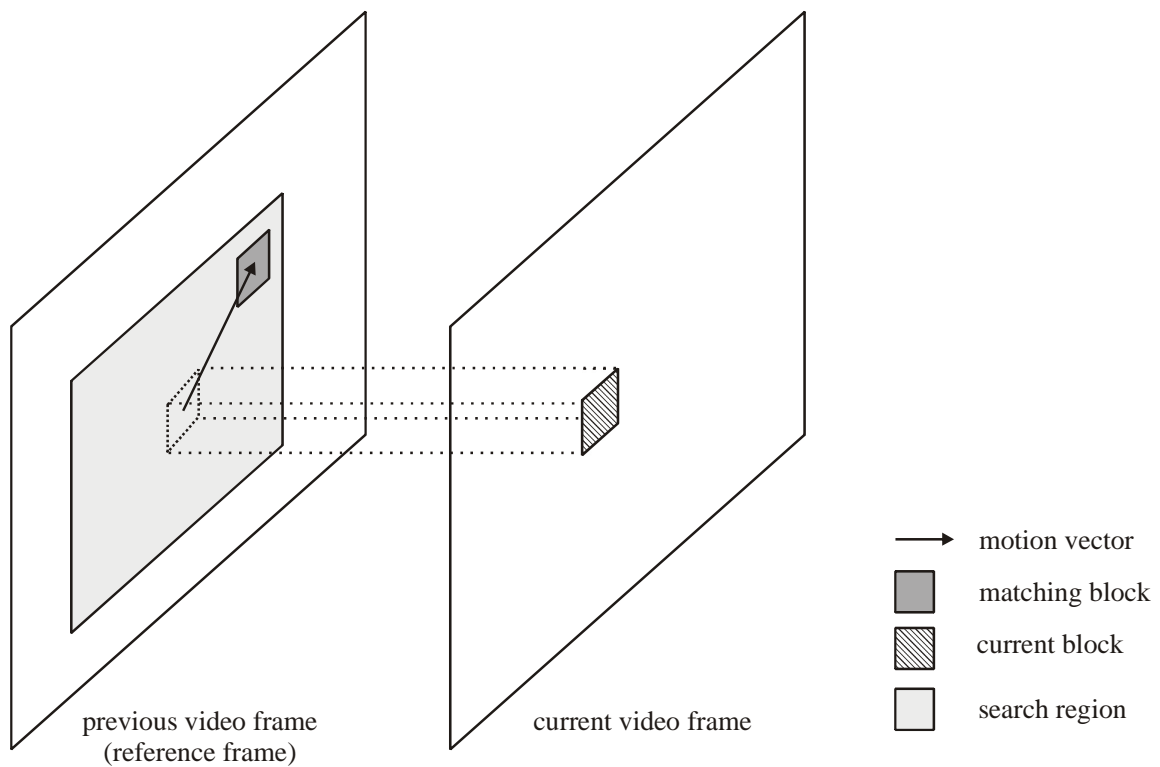


Fig. 2.2. Block-matching algorithm of motion estimation.

2.3.2. Pel-recursive algorithms of motion estimation

Pel-recursive algorithms of motion estimation assign one motion vector to each pixel in a video frame, thus the obtained motion field is very dense. The predicted frame is then reconstructed by interpolation of the reference frame at locations pointed by motion vectors [Tek95, Van03].

An example of pel-recursive motion estimation is the algorithm developed by Netravali and Robbins [Net79]. Netravali-Robbins algorithms usually lead to noisy fields of motion vectors that do not describe a “true” motion in a video sequence; however, they were used in early video compression algorithms [Net79, Wal87].

Other groups of pel-recursive techniques are differential methods of optical flow estimation, e.g. Horn and Schunck algorithm [Bar94]. The algorithm combines a gradient constraint equation with a global smoothness term [Hor81]. As a result, the

estimated motion field is dense and smooth. Optical flow based motion compensation was used for video compression by Lin and Shi [Lin97].

Currently, pel-recursive algorithms of motion estimation are not often used in video compression, mainly because of their computational complexity as well as obtained dense and rough motion field.

2.4. Scalable video coding with motion compensated-prediction

Many scalable video coding algorithms were proposed in order to represent multiresolution video sequences. Proposed techniques are divided into two major classes: wavelet-based video decomposition and modifications of hybrid video coding with DCT-like transform [Dom04].

Wavelet video coding techniques natively enable multiresolution representation of video [Woo02, Ohm02a]. In first approaches, wavelet-based techniques were used in order to achieve spatial scalability only, these techniques are called two-dimensional (2D) techniques. Further improvements introduced wavelet analysis also into temporal domain. These techniques are called three-dimensional (3D) techniques and enable temporal scalability as well. The techniques of scalable video coding using wavelets are presented in more detail in Section 2.4.1.

On the other hand, hybrid video coding with DCT-based transform was originally developed for classic, non-scalable video coding. Scalability was introduced into this class of video codecs later on. However, some powerful and very efficient scalable codecs that use motion-compensated prediction with DCT transform have been proposed over the years. One of these is presented in Section 2.4.2.

Most recent proposals incorporate a scheme of wavelet-based temporal analysis into classic hybrid-coding techniques. Thus, the meaning of term “3D techniques” has been extended also to hybrid scalable codecs.

2.4.1. Wavelet codecs

In a 2D scheme of wavelet decomposition, wavelet transform is used for spatial decorrelation of residual information instead of DCT-based transform in a coder with motion-compensated prediction [Bla98].

Regarding temporal analysis, the 3D video codecs are divided into two major groups [Abh03]:

- t+2D approach – spatial decomposition is performed after temporal analysis,
- 2D+t approach – temporal analysis follows spatial decomposition.

In t+2D techniques spatiotemporal decomposition comprises of temporal lifting step followed by a spatial decomposition of the video signal [Ohm93, Tau94]. The t+2D scheme provides an efficient scalable representation of a video sequence. Embedded bitstream provides an ability to decode a video with a variety of spatial and temporal resolutions and allows for fine granular bitstream decoding. Temporal subband coding [Ohm92, Pod95] produces good results in video compression but introduces blurring artifacts for video sequences with low frame rate [Con97]. On the other hand, the recursive decomposition in temporal domain introduces high decoding delay [Dom04] and makes it difficult to achieve random access to compressed video, which is important in many video coding applications [MP03-25, Zil05].

The 2D+t group of wavelet methods was inspired [Abh03] by the lifting scheme for wavelet [Swe95]. In 2D+t approach spatiotemporal decorrelation is performed using wavelet techniques of spatial subband coding with motion compensated prediction. In such an approach, motion compensation can be easily incorporated into temporal lifting step. Wavelet coders that exploits 2D+t scheme was proposed by Choi and Woods and Xu et al [Cho99, Xu02].

Wavelet-based scalable coding of video sequences has been proposed for over a decade. However, only recently presented video coders with motion compensated temporal filtering (MCTF) [Ohm02, Fli03], lifting schemes [Sch04] and 3-D wavelet decomposition [Ji04] with inter-layer encoding of motion vectors appear to be competitive with classic hybrid coding algorithms [Dom04].

2.4.2. Hybrid scalable codec with DCT-based transform

Various spatial and temporal resolutions of the same video sequence can be represented using classic hybrid compression algorithm with motion compensated prediction. Coarse-to-fine pyramid coding using hybrid video coders proved to be quite efficient approach for multiresolution, layered video representation [Nav94, Dom00, Mac03]. An example of a structure of scalable hybrid coder with layered approach is given in Fig. 2.3. This structure was used in order to develop advanced scalable hybrid coder at Poznań University of Technology [Bła04a, Bła04b, Bła06].

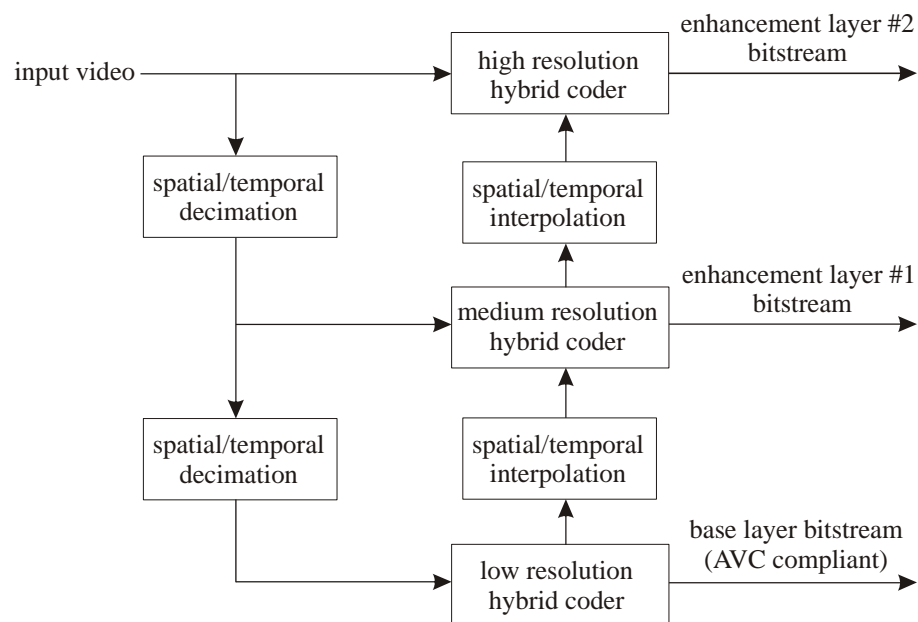


Fig. 2.3. Scalable hybrid coder with layered multiresolution video representation.

Video coder from Fig. 2.3 produces scalable bitstream that consists of three layers. The lowest resolution layer is called base layer [Dom00, Ohm01]. It enables to decode video sequence with a basic spatial and temporal resolution. Additional decoding of enhancement layer #1 allows for increasing the resolution of the decoded video sequence. When all three layers are decoded (base layer, enhancement layer #1 and enhancement layer #2), the video sequence is reconstructed with full resolution.

Scalable encoder from Fig. 2.3 consists of three hybrid sub-coders, each one operating on different level of spatial and temporal resolution [Mac02, Lan04]. Each sub-coder has its own, independent prediction loop. Encoded pictures from lower layers

are interpolated and used as additional reference frames for inter-layer prediction. In order to reduce high-frequency components of prediction residuals, advanced edge-preserving interpolation technique is involved [Dom03]. Temporal scalability is achieved by dropping of non-referenced B-frames, independently in each layer.

The codec discussed in this section use the most advanced techniques in order to achieve video compression [Bla06]. Base layer bitstream uses the same syntax and semantics as state-of-the-art AVC/H.264 video codec. Therefore, the base layer is properly decodable by all AVC/H.264 decoders, which is a great advantage. However, in enhancement layers there have been introduced modifications in order to improve the compression efficiency by exploiting data from the base layer [Dom99, Ros01, Mac02]. Some new prediction techniques have been added and existing prediction techniques have been modified [He01, Mac03, Bla06]. Available techniques of macroblocks coding in enhancement layer are presented in Tab. 2.1.

Tab. 2.1. Available prediction modes in enhancement layer of the scalable video coder from Fig. 2.3 [Bla03, Bla06].

frame type	macroblock prediction modes
Intra-coded (I)	<ol style="list-style-type: none"> 1. Spatial interpolation from base layer (16×16 block size). 2. All standard intra prediction modes.
coded with motion-compensated prediction (P)	<ol style="list-style-type: none"> 1. Prediction (forward) from the nearest reference frame. 2. Spatial interpolation from base layer (4×4 - 16×16 block size). 3. Average of two above (1, 2). 4. Temporal prediction modes from other reference frames, as defined in AVC specification. 5. All standard intra prediction modes.
coded with bidirectional motion-compensated prediction (B)	<ol style="list-style-type: none"> 1. Prediction (forward, backward and bidirectional) from the nearest reference frame. 2. Spatial interpolation from base layer (4×4 - 16×16 block size). 3. Average of two above (1, 2). 4. Temporal prediction modes from other reference frames, as defined in AVC specification. 5. All standard intra prediction modes.

As a result, the output scalable bitstream syntax is the same as that of AVC/H.264 codec with only small modifications of semantics of some syntax elements. Furthermore, as stated before, the base layer of the bitstream is fully AVC/H.264 compliant.

More detailed description of this scalable hybrid codec is given in [Bła04a, Bła04b].

2.5. Summary

In this chapter, a technique of video compression using motion-compensated prediction has been discussed. An architecture of hybrid video coder has been presented and main features of advanced video codecs have been pointed.

Techniques of motion estimation have been presented in Section 2.3, including block-matching algorithm and pel-recursive approach.

Scalable video codecs that exploit motion-compensated prediction are used in order to represent a video sequence with many spatial and temporal resolutions. Two major approaches to scalable video coding have been presented: wavelet-based approach and DCT-based approach. One of DCT-based approaches: advanced scalable video codec, developed at Poznań University of Technology has been presented more thoroughly in Section 2.4.2.

In the following chapter, motion model used during motion-compensated prediction in non-scalable video codec is considered in more detail. Many techniques used for reducing the energy of residual signal are presented, including variable block size, fractional accuracy of motion compensation or multi-hypothesis prediction.

Later in Chapter 3, several approaches to coding of estimated motion vectors are presented. Some practical examples of using these techniques are also given.

Chapter 3.

Motion model and its representation in non-scalable video coding

3.1. Introduction

In early video coding algorithms a motion model was quite simple [Kog81, Nin82, CCI84, Eri85]. It assumed only translational motion of fixed-size rectangles, usually being blocks of 16×16 luminance samples [Che93]. With motion-compensated prediction using 16×16 macroblocks only, motion estimation algorithm is relatively easy to implement and encoder control is not difficult. However, motion-compensated prediction using large blocks limits the ability of accurate prediction and introduces visible artifacts, known as blocking artifacts, especially for low bitrates [Dom98, Ska98, Ohm04].

Motion model was improved in next generations of video codecs [Dom05, Sul05, Lan06e]. Special prediction modes were added for interlaced video: in MPEG-2 algorithm, macroblocks are predicted using blocks of 16×16 or 16×8 luminance samples [ISO94, Tek95]. Multi-hypothesis motion-compensated prediction with multiple reference frames is employed in the newest video coding algorithms [Wie03, Dom03, ISO06]. On the other hand, variable block size was introduced in order to better match the shape of objects in video sequence [Sul91a, Ohm04]. Therefore, for complete representation of the motion data, the following information has to be encoded in most advanced video coders:

- motion vectors,
- reference frame indices,
- control data for a macroblock (partitioning, prediction mode).

In natural video sequences estimated motion vectors in a video frame are locally very similar in values. Therefore, in order to improve overall coding efficiency, motion field is compressed by reducing statistical redundancies.

The problem of motion modeling and motion vectors representation regards so called hybrid video coders, as well as wavelet video coders, in which techniques of motion compensation are involved. In both approaches, the same techniques of motion estimation and motion vectors encoding can be used.

In the following sections, some features of a motion model have been presented. The influence of motion model on motion vectors estimation and coding has been briefly discussed. Methods of motion vector compression are presented as well.

3.2. Motion compensation in advanced hybrid coders

3.2.1. Motion model

3.2.1.1. Variable block size

Motion-compensated prediction with blocks of variable size significantly improves the efficiency of video compression. In Variable Block Size Motion Compensation (VBSMC) technique, a predictive-coded image is decomposed into blocks of varying size and a motion vector is associated to each block [Cha90]. This approach allows for flexible motion segmentation: large blocks are chosen in areas with stationary motion field, while small blocks are chosen in areas with varying motion field. The decomposition structure of an image is coded as a side information and a motion vector is transmitted for each block. Thus, VBSMC algorithm adaptively decomposes an image into blocks with uniform motion, using rate-distortion optimization [Sul91a]. Examples of decomposition of video frame into partitions of variable size are depicted in Fig. 3.1.

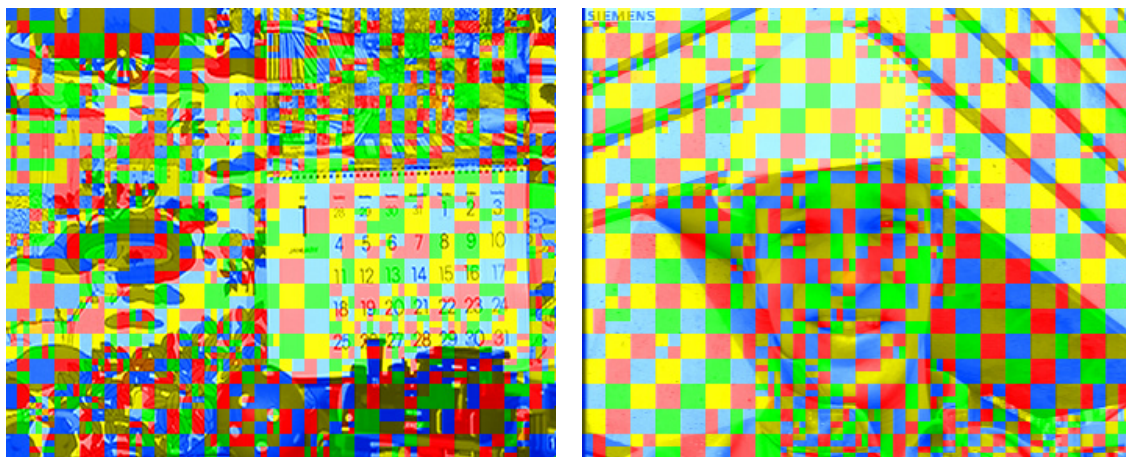


Fig. 3.1. Segmentation of a frame in video sequence using blocks with variable size (from 16×16 to 4×4 luminance sample). Frame 58 from *Mobile* sequence, frame 44 from *Foreman* sequence.

In most video coding algorithms, a basic partition with associated single motion vector is rectangular block of 16×16 luminance samples (macroblock). Macroblock can be usually decomposed further into smaller partitions. The size of the smallest partition is limited to 8×8 [ITU05, ISO98, SMP05, AVS06] or 4×4 [ISO06] luminance samples. As a result, encoder can flexibly choose the partitioning of each macroblock. For example, in AVC algorithm, 7 block sizes are available: 16×16 , 16×8 , 8×16 , 8×8 , 8×4 ,

4×8 and 4×4 [ISO06]. Each block can have one or two motion vector assigned. The maximum number of motion vectors per macroblock in AVC codec is 32, as showed in Tab. 3.1.

Tab. 3.1. Maximum number of motion vectors per macroblock for various partitioning in AVC codec (according to [ISO06]).

Partition size	Number of partitions in one macroblock	Number of motion vectors per macroblock (uni-directional prediction)	Number of motion vectors per macroblock (bi-directional prediction)
16×16	1	1	2
16×8	2	2	4
8×16	2	2	4
8×8	4	4	8
8×4	8	8	16
4×8	8	8	16
4×4	16	16	32

When motion-compensated prediction with variable size of blocks is used, compression of motion vector field becomes more complex and ambiguous. In such a case, motion field is non-uniformly sampled, thus obtaining a good predictor for the current motion vector is more difficult.

3.2.1.2. Overlapped Motion Compensation

Overlapped Motion Compensation (OMC) is a technique that eliminates blocking artifacts caused by block motion compensation [Nog92]. OMC is an example of a general concept of multihypothesis motion-compensated prediction [Fli04]. The algorithm uses more than one motion vector for prediction of each pixel, but still requires only one motion vector to be estimated for each block.

Originally [Nog92], the method employed two motion vectors for prediction of each block, as depicted on Fig. 3.2. First, one motion vector is estimated for each block in the current frame. In order to form a prediction signal, many motion vectors are used for

each block: a motion vector estimated for the current block and additional motion vectors estimated for neighboring blocks. These motion vectors point at blocks from a reference frame used for prediction. Prediction signal is obtained by summing up overlapped blocks weighted with window function.

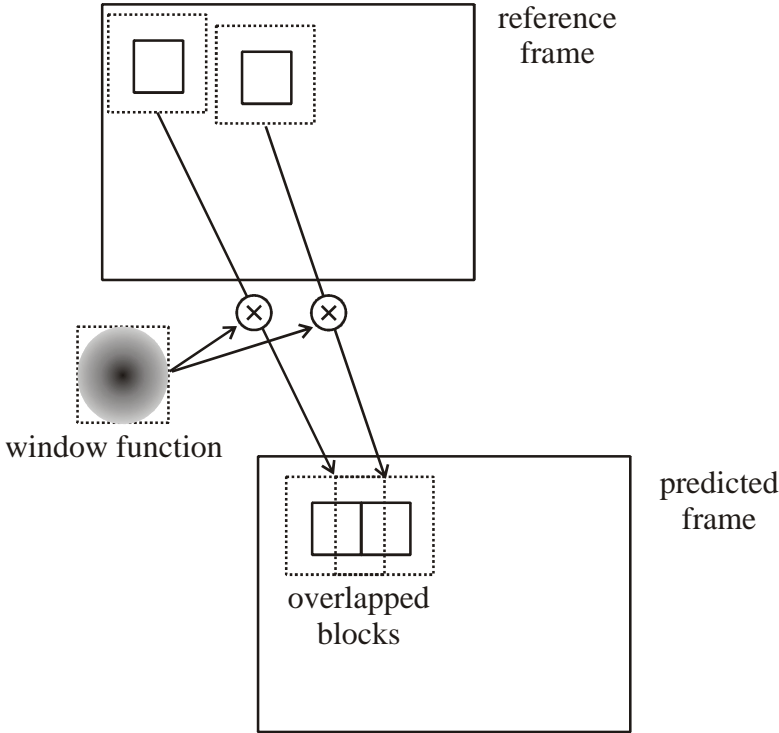


Fig. 3.2. Overlapped Motion Compensation scheme, as proposed by Nogaki and Ohta [Nog92].

Overlapped Motion Compensation scheme has been adopted in video coding algorithm described by H.263 standard [ITU05]. Five motion vectors are used in order to form prediction signal for each 8×8 block. Beside the “main” motion vector of the current block (mv_0), motion vectors from adjacent blocks are used (mv_1, mv_2, mv_3, mv_4) as showed in Fig. 3.3.

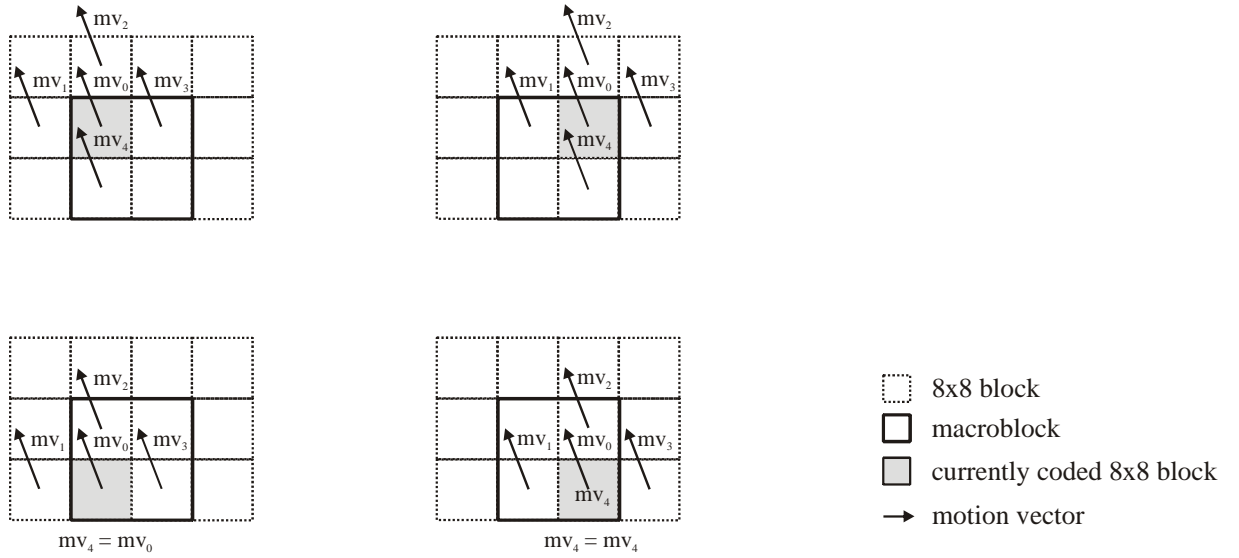


Fig. 3.3. Motion vectors used for overlapped motion compensation in H.263 codec [ITU05]. Four cases are shown, for each of the 8×8 block of the macroblock.

Blocks of samples from a reference frame, pointed by motion vectors $mv_0 \dots mv_4$, are weighted with windowing arrays H_0, H_1, H_2, H_3 and H_4 and form prediction blocks $p_0 \dots p_4$:

$$p_k(x+i, y+j) = H_k(i, j) \cdot \hat{I}_M(x+mv_{k,x}+i, y+mv_{k,y}+j), \quad (3.1)$$

where $0 \leq i, j < 8, 0 \leq k < 5$ and:

- p_k – partial prediction signal,
- H_k – weighting array given by (3.2),
- $\hat{I}_M(x, y)$ – a sample from the reference frame M , taken from location (x, y) ,
- $mv_{k,x}, mv_{k,y}$ – the horizontal and vertical components of k -th motion vector ($0 \leq k < 5$).

Windowing arrays H_0, H_1, H_2, H_3 and H_4 used during calculation of partial prediction signal are given by the following matrices:

$$\begin{aligned}
H_0 &= \begin{bmatrix} 4 & 5 & 5 & 5 & 5 & 5 & 5 & 4 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 6 & 6 & 6 & 6 & 5 & 5 \\ 5 & 5 & 6 & 6 & 6 & 6 & 5 & 5 \\ 5 & 5 & 6 & 6 & 6 & 6 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ 4 & 5 & 5 & 5 & 5 & 5 & 5 & 4 \end{bmatrix} \\
H_1 &= \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\
H_3 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \end{bmatrix} \\
H_2 &= \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
H_4 &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}
\end{aligned} \tag{3.2}$$

The final prediction \tilde{I} is obtained by summing the products according to equation:

$$\tilde{I}(x, y) = p_0(x, y) + p_1(x, y) + p_2(x, y) + p_3(x, y) + p_4(x, y). \quad (3.3)$$

Overlapped motion compensation along with variable block size makes the coding algorithm more complex, as more hypotheses have to be tested in order to choose the best prediction mode for the macroblock.

3.2.1.3. Fractional motion vectors

An important feature of the motion model is the accuracy of the estimated motion vectors. The most simple algorithms estimate integer motion vectors only. It means that the minimal allowed displacement of a motion-compensated block is one luminance sample. If motion in video sequence is, for example, slower than 1 sample per frame, it cannot be properly compensated; such a situation is depicted in Fig. 3.4.

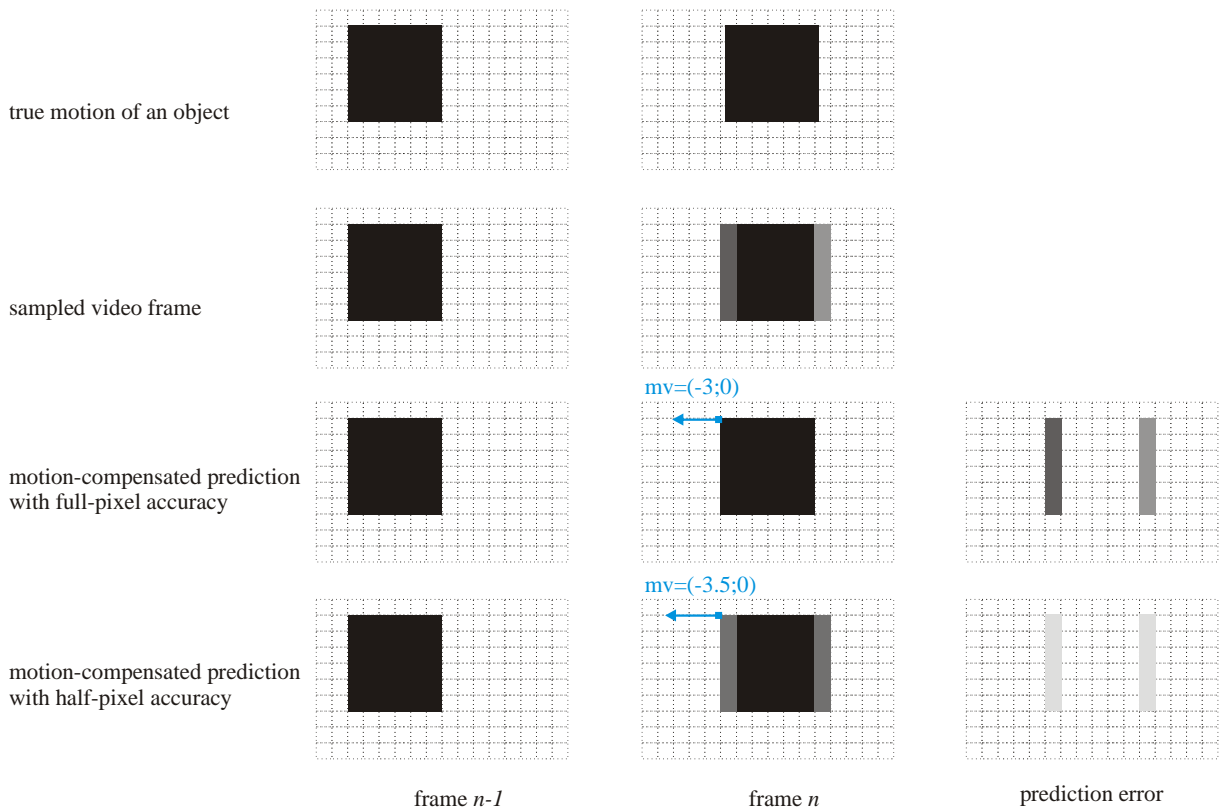


Fig. 3.4. An example of motion compensated prediction using integer and fractional motion vectors. Lighter signal has a less energy – it is more efficiently coded.

It has been reported [Gir87] that with integer-pixel accuracy of the motion vector, a gain introduced by motion-compensated prediction over optimum intra-frame encoding of the signal is limited to ~ 0.8 bit/sample in moving areas. An improvement of prediction for displacements with non-integer velocities can be achieved using fractional motion estimation. In order to form a prediction signal at non-integer samples locations, spatial interpolation is performed using zero-phase FIR filters. With this technique, motion vectors are estimated with the accuracy of $\frac{1}{2}$, $\frac{1}{4}$ or even $\frac{1}{8}$ of a distance between samples. The scheme allows for aliasing cancellation and more accurate modeling of a motion in order to predict better current values of samples. Better prediction minimizes the prediction error, as depicted in Fig. 3.4.

Motion model using motion vectors with accuracy of $\frac{1}{2}$ -pel has been used in MPEG-2 video coding algorithm. Simple averaging filter has been adopted for estimation of samples at half-pel locations. More sophisticated interpolation scheme is used in the most advanced video coding algorithms. For example, during developing of AVC/H.264, $\frac{1}{3}$ - and $\frac{1}{8}$ -sample accurate motion-compensated prediction was proposed, but finally the idea was dropped due to complexity reasons [Wie03]. After all, motion vectors in AVC/H.264 video codec are estimated using $\frac{1}{4}$ -pel accuracy.

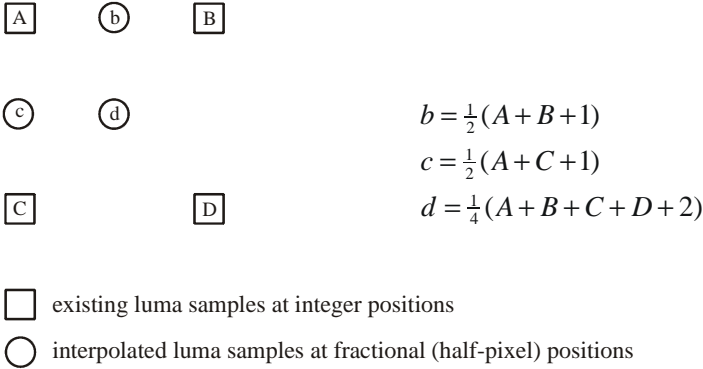
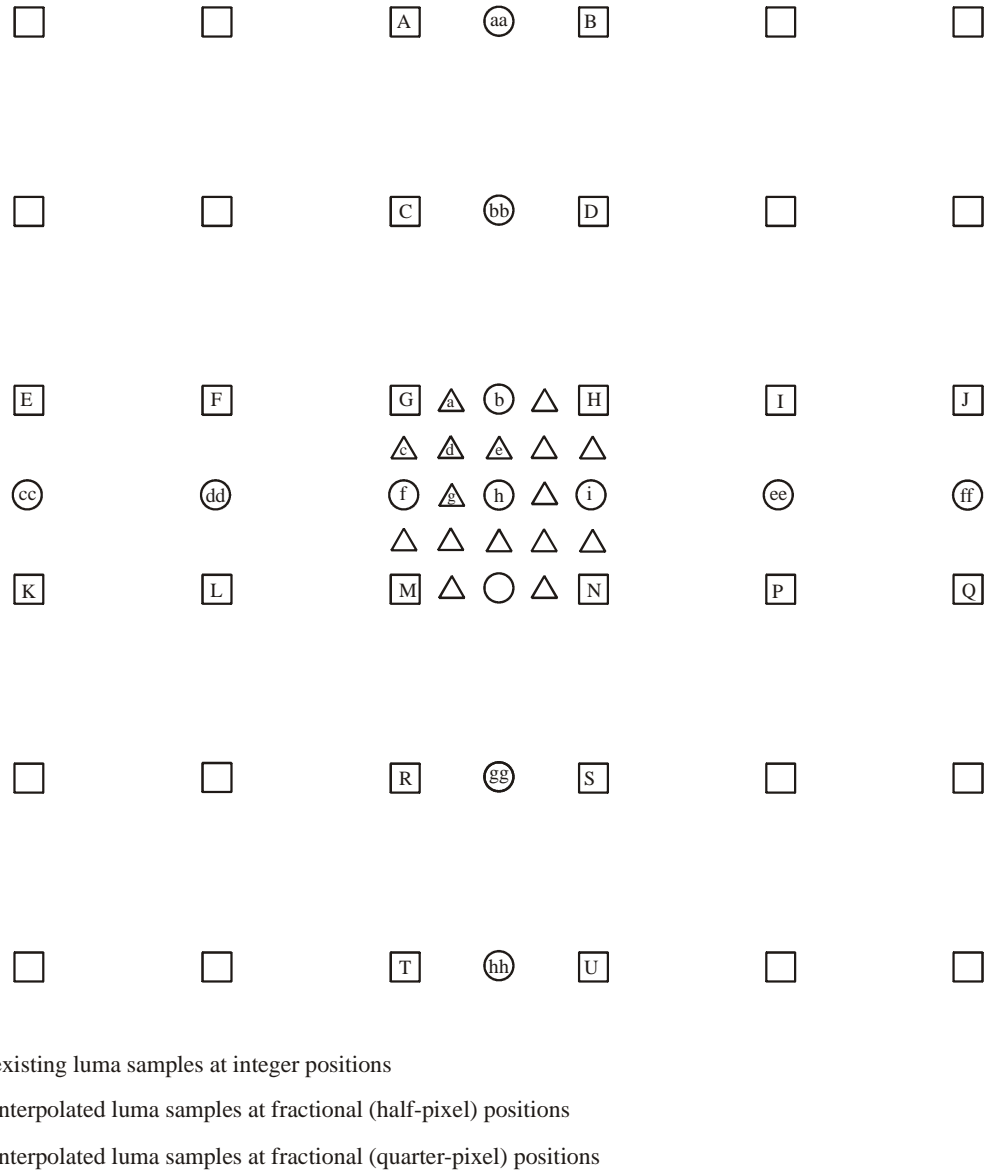


Fig. 3.5. Interpolation of fractional luma samples in MPEG-2 and H.263 video coding algorithms. Prediction values at half-sample positions are obtained by averaging samples at integer-sample positions [ISO94, ITU05].

Interpolation schemes for fractional locations of luminance samples in various video codecs are depicted in Fig. 3.5 and Fig. 3.6. Chrominance samples are estimated using

the same [ISO94, ITU05] or similar (usually simpler) [ISO06] interpolation schemes as luminance samples.



$$b = \frac{1}{32}(E - 5 \cdot F + 20 \cdot G + 20 \cdot H - 5 \cdot I + J + 16)$$

$$f = \frac{1}{32}(A - 5 \cdot C + 20 \cdot G + 20 \cdot M - 5 \cdot R + T + 16)$$

$$h = \frac{1}{32}(cc - 5 \cdot dd + 20 \cdot f + 20 \cdot i - 5 \cdot ee + ff + 16)$$

$$a = \frac{1}{2}(G + b + 1)$$

$$c = \frac{1}{2}(G + f + 1)$$

$$e = \frac{1}{2}(b + h + 1)$$

$$g = \frac{1}{2}(f + h + 1)$$

$$d = \frac{1}{2}(b + f + 1)$$

Fig. 3.6. Interpolation of fractional luma samples in AVC/H.264 video coding algorithm.

The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter. Prediction values at quarter-sample positions are obtained by averaging samples at integer- and half-sample positions [ISO06].

Improving accuracy of motion vectors introduces more complexity into motion estimation block of a video encoder: estimation of fractional motion vectors require samples interpolation which is usually very slow, because of huge number of samples to process. Estimation of motion vectors with sub-pixel accuracy allows for significant improvement of the efficiency of video compression, however encoding process is more complex and lasts longer. Usually, fractional motion estimation is performed only for sub-pixel locations around the neighborhood of estimated integer-valued motion vector. It allows for reducing time complexity.

3.2.1.4. Motion vectors over picture boundaries

Recently developed video coding algorithms allow motion vectors to point outside the regular spatial area of a picture. In order to form a prediction signal, the “virtual” image area is extrapolated by expanding the regular image area and repeating the edge samples. The gain in compression efficiency is achieved by exploiting of predictive coding of motion vectors – in some cases encoder do not need to sent a residual of a motion vector. The signal of prediction is formed from the “virtual” area of a reference picture.

On the other hand, the technique of motion vectors over picture boundaries makes motion estimation at the side of encoder simpler and more uniform. Encoder estimates motion in the same way for all blocks of a video frame and do not need check the “edge conditions”. “Edge conditions” provide that motion vectors point inside the reference picture.

This technique is simple and improves coding performance in some cases, thus it has been used in many video coding algorithms [ITU05, ISO06].

3.2.1.5. Global Motion Compensation

Global Motion Compensation (GMC) is an advanced tool used in video processing and coding [Per02]. GMC utilizes the phenomenon that a significant part of a visible motion within the video sequence is caused by camera motion [Smo04]. For example, translational movement of camera causes global, uniform motion in a video sequence.

Global motion is also caused by rotation of a camera, as well as scene zooming and panning.

The proposed approach is to model this global motion by a set of parameters which can be sent to decoder in order to form a predicted video frame with global motion compensation. It allows for reducing the bitrate and achieving better compression efficiency.

Global Motion Compensation together with technique of *sprites* encoding were researched to be used in object-based video coding algorithms [Smo99].

Sprite technique allows for representing of video scene as a set of patches [Ohm04]. A patch can be defined as an image area that contains a visual object. Each patch represents an object from given video scene and can be encoded independently. The final image is assembled from these patches, using a description of the scene.

The most widely known object-based video coding technique is described by MPEG-4 standard [ISO98]. MPEG-4 distinguishes entities in video frames that can be accessed and manipulated independently [Sad02], for example content-based encoding can be performed separately for text, static background and moving object. These entities in MPEG-4 are called Video Object Planes (VOP).

In Advanced Simple Profile (ASP) of MPEG-4 Visual, a picture that is coded using prediction based on global motion compensation is called S(GMC)-VOP. Global spatial transformations, including image warping, are used in order to improve the efficiency of the prediction of sample values [ISO98]. The perspective model of global motion in MPEG-4 is defined with the following equations [Che05]:

$$\begin{aligned}
 p(x, y) &= r(x', y'), \\
 x' &= \frac{m_0x + m_1y + m_2}{o_1x + o_2y + 1}, \\
 y' &= \frac{n_0y + n_1x + n_2}{o_1y + o_2x + 1},
 \end{aligned} \tag{3.4}$$

where:

- $p(x,y)$ – motion-compensated prediction signal at location (x,y) ,
- $r(x',y')$ – sample from reference frame, taken from location (x',y') ,
- m_0, n_0 – scaling factors,

- m_1, n_1 – rotation factors,
- m_2, n_2 – translation factors,
- o_1, o_2 – tilt factors.

A technique of Global Motion Compensation also has been proposed for AVC/H.264 video codec [Smo04]. Authors reported even 20% bitrate saving at the same visual quality for some classes of video sequences.

3.2.1.6. Motion compensation using control-grid interpolation

In control-grid interpolation (*CGI*) method of motion compensation, a video frame is divided into small regions by regular grid (*control grid*) [Ska98, Ohm04], as depicted in Fig. 3.7. Motion is estimated for nodes of control grid using any algorithm (either block matching or pel-recursive). Samples for motion-compensated prediction are obtained by interpolation of the samples between nodes from the reference frame. Thus, control-grid interpolation allows for spatial image transformations and motion-compensated prediction [Sul91].

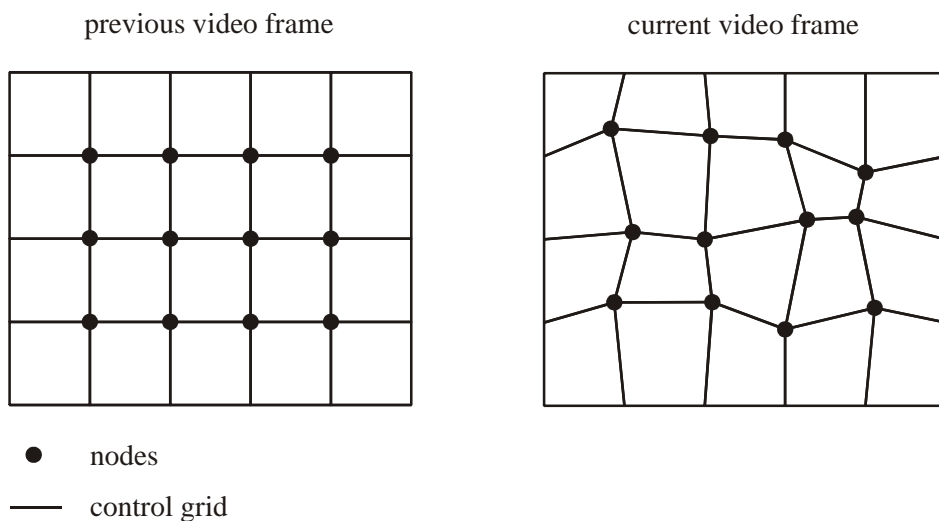


Fig. 3.7. Control-grid interpolation scheme.

In Fig. 3.7 the control grid is square; however, a triangular grid can also be used. In such a particular case, the method is called Triangle Motion Compensation (*TMC*). A

video frame in TMC is tessellated using triangular patches; motion in the frame is described using transformation of these triangular patches [Nak91].

Block Matching Algorithm described in Section 2.3.1 may be considered a special case of CGI technique: nodes of control grid are situated in the centers of blocks and nearest-neighbor scheme of interpolation is used [Sul91]. All samples from given block share the same motion vector. A simple interpolation scheme used in BMA method (nearest-neighbor) causes visual artifacts that appear in the borders of blocks, they are called blocking artifacts.

Control-grid interpolation technique has been proposed over the years in applications for video compression [Sul91, Chu94, Ish97, Cho05]. However, mainly because of the complexity reason, block-based methods are used most often.

3.2.2. Temporal prediction modes

In its original version, motion-compensated prediction was performed using frames from the past only [Roc69, Mou69]. This technique provided quite efficient video representation. For each predicted block, a single motion vector is estimated that refers to the reference frame from the past. The scheme that uses previous video frame for motion-compensated prediction is called forward prediction [Tek95, Dom98, Ohm04]. However, some problems with prediction from previous frames may occur regarding, for example, covered and uncovered background that does not exist in a video frame from the past [Ohm04]. On the other hand, good prediction from the past frames is not possible for the content that appears just in the current frame.

In order to improve coding efficiency, backward prediction that can exploit similarity of a current frame to the frame from the future was proposed [Yam89]. In such a case, motion-compensated prediction is performed using frame from the future thus estimated motion vector refers to the frame from the future.

Combination of the two modes of motion-compensated prediction: forward and backward is called bidirectional prediction [Hid89, Lig89, Son89, Ric02]. Two motion vectors are estimated in bidirectional prediction: one that refers to a frame from the past and one that refers to a frame from the future. The final prediction signal is obtained by averaging samples from the past reference frame with samples from the future reference frame. Forward, backward and bidirectional predictions are depicted in Fig. 3.8. Frame

naming convention used in Fig 3.8 (I-frame, P-frame and B-frame) have been discussed in Section 2.1.

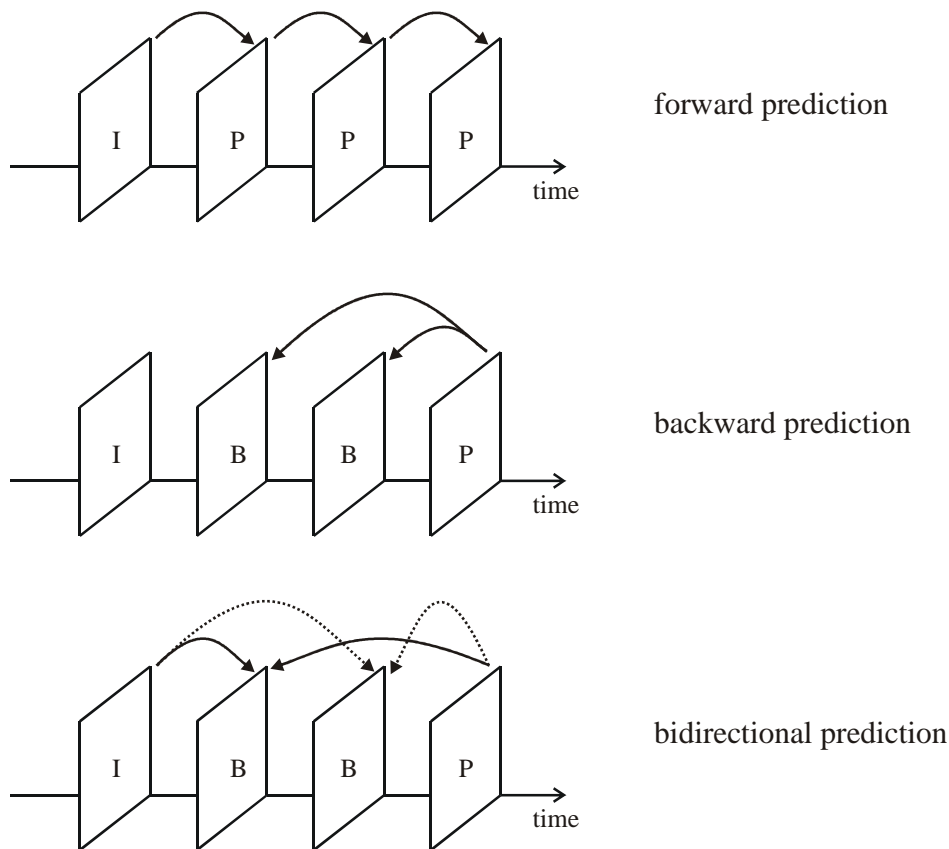


Fig. 3.8. Forward, backward and bidirectional motion-compensated prediction.

Decision about temporal prediction mode can be taken independently for each macroblock, which means that each macroblock can be predicted using different references: from the past, from the future or both. It allows for very efficient prediction of the content of video frame. Utilized prediction modes have to be sent by the encoder as a side information.

3.2.3. Multiframe motion compensation – multi-hypothesis prediction

In classic video coding, only the nearest frame (either from the past or from the future) is used as a reference frame for motion-compensated prediction. However, multiframe techniques, also called multi-hypothesis predictions [Fli04], have been proposed [Muk85, Got93, Wie99a]. Together with long-term memory, prediction using multiple reference frames is useful in the case of uncovered background: uncovered

fragment of the current frame is predicted from the frame on which it was visible. Long-term memory implemented in video codec allows for storing and referencing the video frame that appeared before the most recent frames. An example of motion-compensated prediction using long-term memory has been depicted in Fig. 3.9.

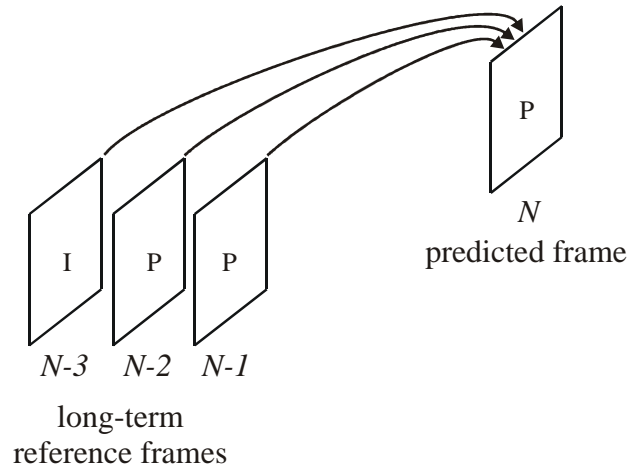


Fig. 3.9. Motion-compensated prediction using long-term memory.

The technique of multiframe motion compensation is also used in order to increase error resilience of a codec. In the case of transmission errors, the reference frame is erroneously reconstructed and errors propagate further. However, when multiple reference frames are used, encoder can encode the current frame regarding the frame which was reconstructed properly.

Multiframe motion compensation can be used for forward, backward and bidirectional prediction. Extra information for each motion vector has to be sent about reference frame that should be used for prediction (index of the reference frame). Therefore, trade-off between better prediction and extra motion information has to be achieved.

3.2.4. Inference of motion information

Motion vectors estimated for natural video sequences are spatially and temporally correlated, thus, in some cases, motion information can be completely inferred from the temporal or spatial neighborhood without extra information being sent [Won95, Tou01, Sun01, Tou05]. This technique provides very efficient video

representation, because neither motion vectors nor macroblock prediction mode have to be encoded in a bitstream [Sul03, Tou05].

Motion vector field is stationary when motion vectors from neighboring locations are highly correlated with each other [Sun01]. In the case of high spatial correlation, motion field is spatially stationary. In the case of high temporal correlation, motion field is temporally stationary.

Motion information can be inferred very accurately in the case of stationary motion vector field [Lai02, Tou02]. The most advanced tools for motion derivation are incorporated into AVC/H.264 video coding algorithm in macroblocks coded using *SKIP* and *DIRECT* modes [ISO06]. There are defined two kinds of special motion vector predictions that exploit spatial or temporal motion vector similarity in pictures coded using bidirectional prediction.

In temporal mode, motion vectors for the current block are derived by scaling of a co-located motion vector from the reference frame; bidirectional motion-compensated prediction is then performed [Wan95, Tou01]. Co-located motion vector is the motion vector, which was used for motion-compensated prediction of a block that appear at the same spatial coordinates as the current block as depicted in Fig. 3.10.

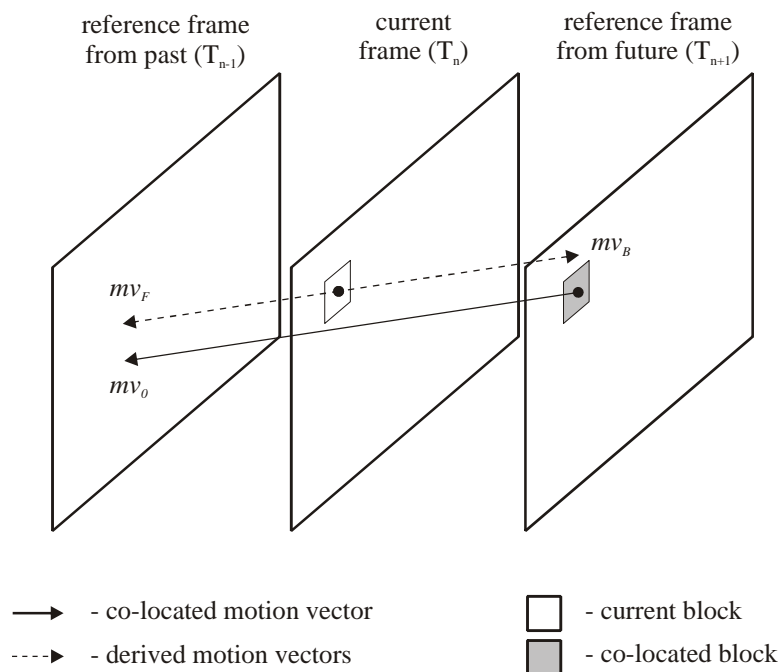


Fig. 3.10. Derivation of motion vectors in temporal motion inference mode in AVC/H.264 [ISO06].

Motion vectors mv_F and mv_B from Fig. 3.10 that are used respectively for forward and backward prediction are derived according to equations:

$$mv_F = \frac{T_n - T_{n-1}}{T_{n+1} - T_{n-1}} \cdot mv_0,$$

$$mv_B = mv_0 - mv_F, \quad (3.5)$$

where:

- mv_0 – co-located motion vector,
- mv_F, mv_B – forward and backward motion vectors,
- T_{n-1}, T_n, T_{n+1} – sampling time of the previous, current and subsequent frame respectively.

The final prediction is obtained by averaging samples coming from forward and backward motion-compensated prediction.

When spatial mode of motion information inference is used in AVC/H.264 video codec, aside from motion vectors, also prediction type is derived from the neighboring blocks [Lan05, Tou05]. Forward, backward and bidirectional prediction can be inferred depending on motion-compensated prediction modes that were used during reconstruction of adjacent blocks. When prediction type and reference frame indices are fixed, motion vector prediction is performed as described in Section 4.2.2.

3.3. Motion vectors representation

3.3.1. Motion vector field

In order to reduce temporal redundancies in a video sequence, a hybrid coder performs motion-compensated prediction using reference frames from the past or reference frames from the future. Therefore, motion vectors are estimated for each video frame that is predictively coded. The locations, for which motion vectors are estimated, depend on motion model that is used in video coding algorithm. Estimated motion vectors form motion vector field. They have to be transmitted to decoder in order to

reconstruct compressed video sequence. Examples of visualization of motion vector field are given in Fig. 3.11.

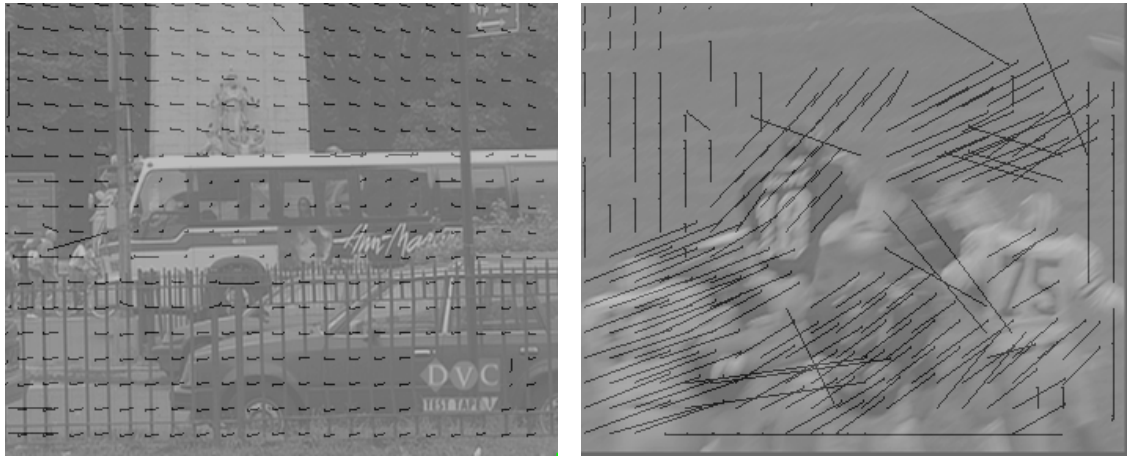


Fig. 3.11. Visualizations of motion vector fields estimated with block matching algorithm. Frame 38 from *Bus* sequence, frame 74 from *Football* sequence, 352×288 (CIF).

Motion vector field estimated in a hybrid video coder is spatially correlated – adjacent motion vectors are often similar to each others. Thus, it is natural to use techniques of residual coding of motion vectors. The most often used technique of motion vectors coding is prediction using previously encoded motion vectors (Fig. 3.12) that are spatially nearby to the currently coded motion vector.

The signal of prediction mv_P is formed in predictor P , and then it is subtracted from the current value of motion vector mv_C . The residual value Δmv is put into the input of entropy coder, which is matched with the statistical characteristics of the differential signal.

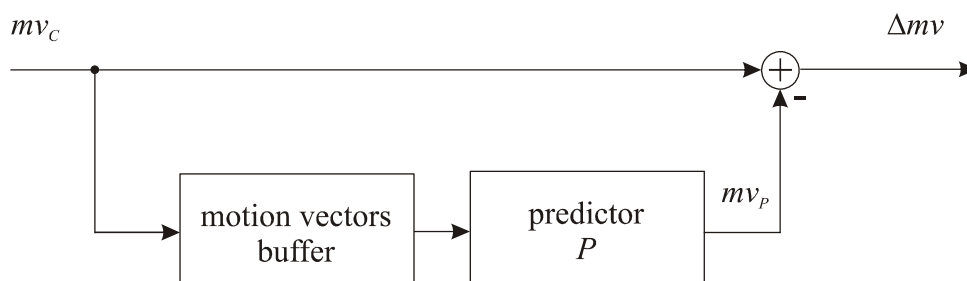


Fig. 3.12. Predictive coding of motion vectors.

Other proposed techniques of representation of motion vectors involve vector quantization and adaptively switched predictors. In the following sections these techniques are discussed briefly.

3.3.2. First-order prediction

In early video compression algorithms, the components of motion vectors were coded using simple DPCM scheme with respect to the last encoded motion vector (Fig. 3.13). For example, in MPEG-2 standard [ISO95] data of each slice of macroblocks is preceded with the unique synchronization codeword and the motion vector components of the first macroblock in slice are encoded independently of any other data. Such an approach resets the predictor of the motion vector for each slice and allows for reliable bitstream error recovery. Moreover, the predictors of the motion vector components are reset whenever an intra coded macroblock is coded which has no concealment motion vector or when macroblock is not coded (skipped) [ISO95].

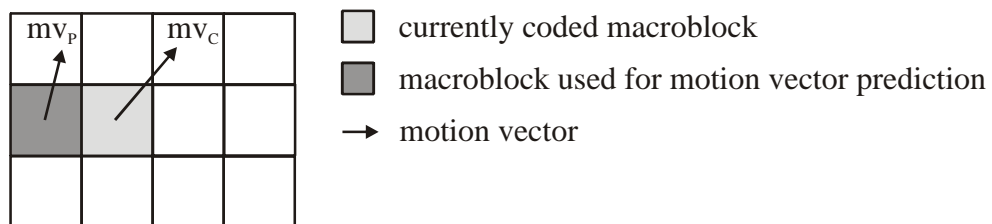


Fig. 3.13. Macroblock used in DPCM motion vector prediction.

In MPEG-2, residual components of the motion vectors are entropy coded using variable length coding. Firstly, scaling factor f_code is sent for all motion vectors in the current picture. For each component of motion vector, UVLC-coded $motion_code$ is sent, as well as fixed-length codeword $motion_residual$ for a motion vector refinement.

With the following 3 variables: f_code , $motion_code$ and $motion_residual$, each component of the motion vector prediction error ($delta$) is derived as follows [ISO95]:

$$\begin{aligned}
 r_size &= f_code - 1 \\
 f &= 1 \ll r_size
 \end{aligned}
 \tag{3.6}$$

```

if ( ( f == 1 ) || ( motion_code == 0 ) )
    delta = motion_code ;
else {
    delta = ( ( Abs(motion_code) - 1 ) * f ) + motion_residual + 1 ;
    if ( motion_code < 0 )
        delta = - delta ;
}

```

where:

- r_size* – auxiliary variable used for derivation of motion vector component,
- f* – bitmask used for derivation of the range of motion vector component.

The final value of motion vector component is then derived according to the predicted value and current scale factor *f_code*:

$$\begin{aligned}
 r_size &= f_code - 1 \\
 f &= 1 \ll r_size \\
 high &= (16 * f) - 1 ; \\
 low &= ((-16) * f) ; \\
 range &= (32 * f) ;
 \end{aligned}
 \tag{3.7}$$

```

vector = prediction + delta ;
if ( vector < low )
    vector = vector + range ;
if ( vector > high )
    vector = vector - range ;

```

where:

- high* – maximum value of the motion vector component,
- low* – minimum value of the motion vector component,
- range* – range of possible values of the motion vector component,
- prediction* – predicted value of the motion vector component,
- delta* – a value of motion vector residual,
- vector* – final value of decoded motion vector component.

Variable f_code is in set $\langle 1, 2, \dots, 9 \rangle$ and limits the range of a motion vector residual according to Tab. 3.2.

Tab. 3.2. The impact of f_code in MPEG-2 on extreme values of motion vector component residual in full-pel units [ISO94].

f_code	minimum value of motion vector residual	maximum value of motion vector residual
1	-8	7.5
2	-16	15.5
3	-32	31.5
4	-64	63.5
5	-128	127.5
6	-256	255.5
7	-512	511.5
8	-1024	1023.5
9	-2048	2047.5

3.3.3. Median prediction

The technique of simple first-order prediction of motion vector components was further replaced by a median prediction [ISO98, ITU05]. In median prediction, components of the current motion vector are predicted from the set of previously encoded motion vector. In order to form a prediction signal, the component-wise median filtering is performed using motion vector from the neighboring blocks depicted in Fig. 3.14.

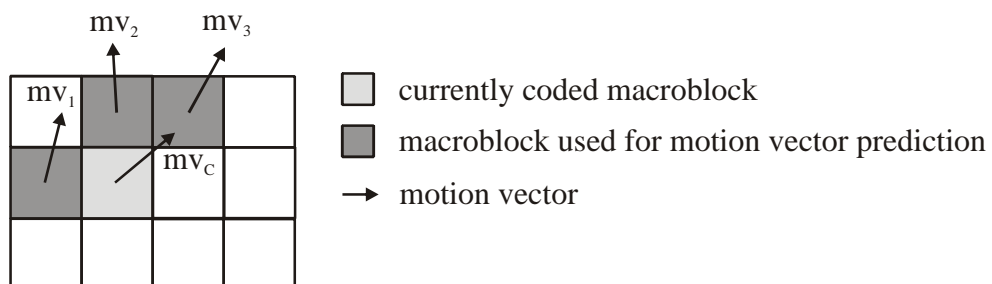


Fig. 3.14. Macroblocks used in median motion vector prediction.

Components of a motion vector are predicted according to equations:

$$\begin{aligned}
 mv_{P,x} &= med(mv_{1,x}, mv_{2,x}, mv_{3,x}), \\
 mv_{P,y} &= med(mv_{1,y}, mv_{2,y}, mv_{3,y}),
 \end{aligned}
 \tag{3.8}$$

where:

- $mv_{P,x}, mv_{P,y}$ – predicted components of motion vector (horizontal and vertical respectively),
- $med()$ – median operator,
- $mv_{1,x}..mv_{3,x}$ – horizontal components of motion vectors used for prediction (Fig. 3.12),
- $mv_{1,y}..mv_{3,y}$ – vertical components of motion vectors used for prediction (Fig. 3.12).

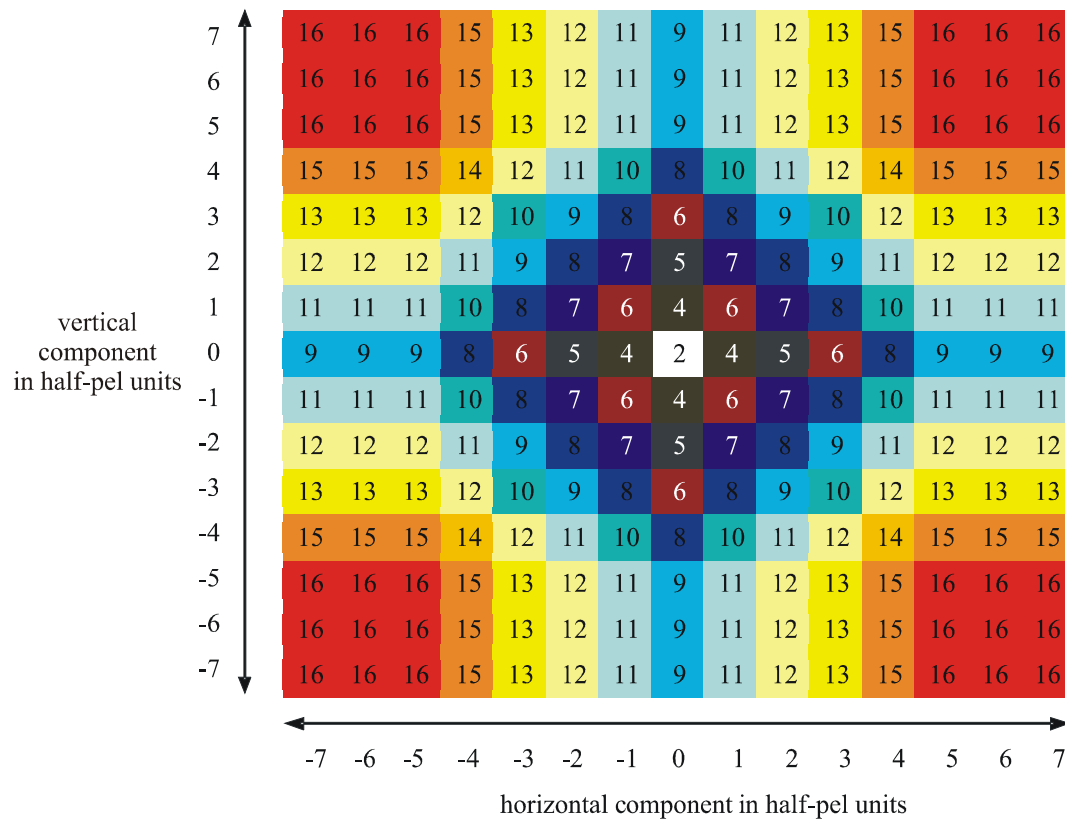


Fig. 3.15. Bits required for encoding motion vector residual in H.263 [Tou99].

Median-based approach to motion vector prediction was used in H.263 video coding algorithm [ITU05]. In the original version of H.263, motion vectors used in motion compensated prediction were restricted to the range $\langle -16, 15.5 \rangle$. Number of bits required for encoding a motion vector residual depending on motion vector residual value in H.263 standard is given in Fig. 3.15.

In order to represent the motion vector residuals in a bitstream efficiently, residual components of a motion vector are encoded in the bitstream using Variable Length Codes (VLC), defined in H.263 standard.

The median-based prediction scheme proved to be extremely efficient and is widely used in most of advanced video coding algorithms. It provides the best coding efficiency for smooth motion vector fields.

3.3.4. Motion vector coding using vector quantization

Vector quantization techniques for compact representation of motion vectors in video coders were proposed by Lee and Woods. They interpreted block-matching motion estimation algorithm as a special type of vector quantization (VQ) and they called it Motion Vector Quantization (MVQ) [Lee95a]. In the algorithm, search region is represented as a codebook and motion vectors are represented as code vectors.

Let codebook $C = \{mv_1, mv_2, \dots, mv_N\}$ represents the motion vector search region, containing N possible locations of blocks used for motion-compensated prediction in the previous frame, as depicted in Fig. 3.16. Motion vector $mv_O = (mv_{O_x}, mv_{O_y})^T$ is the estimated motion vector from the codebook C that is best in the sense of motion-compensated prediction. In order to represent this motion vector, only the index i of the codeword from C has to be transmitted.

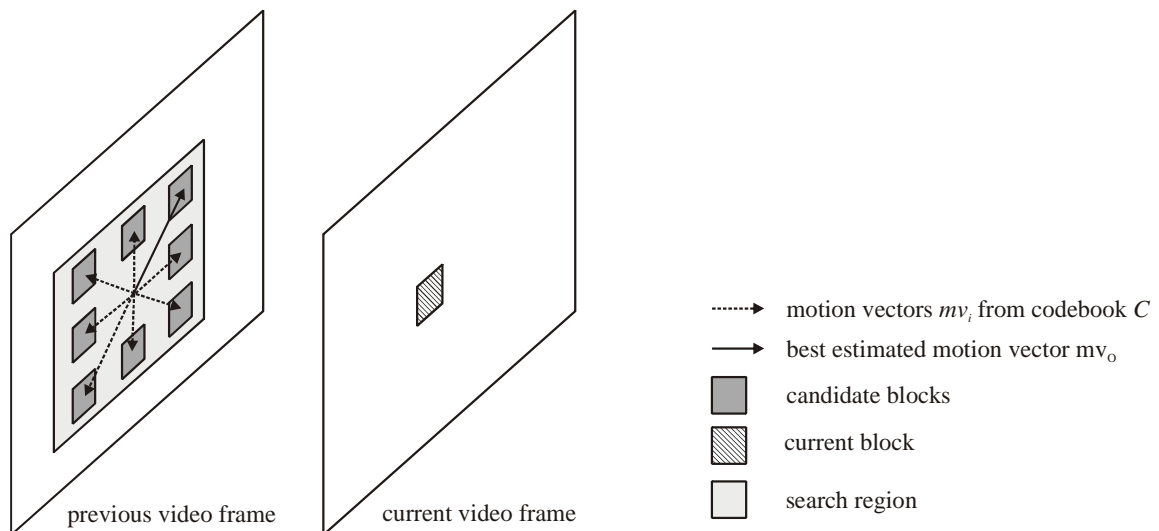


Fig. 3.16. Motion vector representation using vector quantization.

The codebook C has N motion vectors that represent N possible locations of blocks for prediction. Thus, the required bitrate to transmit integer-pel accuracy motion vectors for $M \times M$ block size when fixed-length codes are used is [Lee95a]:

$$r = (\log_2 N \cdot W \cdot H \cdot n) / M^2 \quad (3.9)$$

where:

- r – bitrate required to transmit motion vectors using MVQ (bits per second),
- W – horizontal dimension of a video frame,
- H – vertical dimension of a video frame,
- n – number of frames per second.

Lee and Woods further improved the algorithm of MVQ by developing a “macro motion vector quantizer” that performs coarse vector quantization on a group of motion vectors [Lee95b, Kri97a] and further refinement of obtained results.

Similar approach to motion vectors representation with vector quantization of motion vector field was proposed by Regunathan and Rose [Reg97]. They proposed an iterative algorithm for design of the codebook, which optimizes the compression performance. The estimated, dense motion vector field is represented using a set of “super motion vectors” (SMV) and quantized values of motion vectors of all blocks in a macroblock.

However, the presented algorithms of motion vectors coding using vector quantization have not been employed in advanced video codecs so far. The main reason for this is that, when motion vectors are allowed to be chosen from a limited set only, the efficiency of motion-compensated prediction is worse.

3.3.5. Motion vector coding based on Minimum Bitrate Prediction

A technique of adaptively switched predictors of a motion vector has been proposed by Kim and Ra [Kim99]. The proposal was named Minimum Bitrate Prediction (MBP). In this method, motion vector can be predicted using first-order or median prediction, dependent on values of neighboring motion vectors. The best prediction of current motion vector is chosen from among motion vectors belonging to blocks A, B and C depicted in Fig. 3.17. The best prediction is a vector which enables to produce the minimum bits for representation of the current motion vector (Minimum Bitrate Prediction criterion).

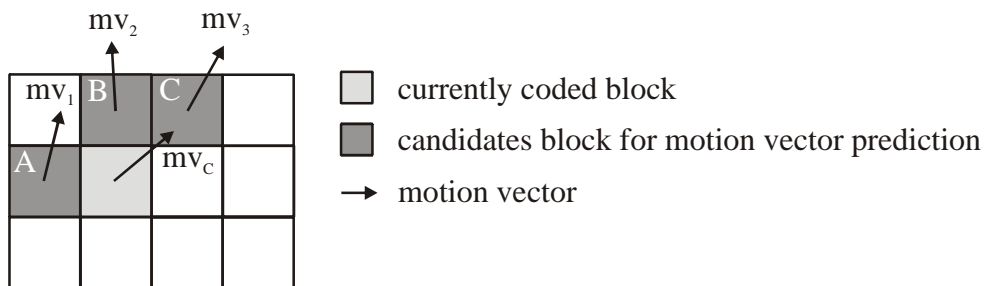


Fig. 3.17. Blocks and motion vectors used in Minimum Bitrate Prediction method of motion vector coding.

Differential values of motion vector components Δmv_x and Δmv_y are calculated with reference to the best prediction and then entropy coded using variable length coding.

Usually, additional information that would determine which neighboring motion vector was used as prediction should be transmitted. Because there are three possibilities, 2 bits are required in order to determine the chosen vector. However, in MBP method, the proper predictor is chosen adaptively based on actual values of neighboring motion vectors mv_1 , mv_2 , mv_3 and transmitted residual values Δmv_x and Δmv_y . Therefore, in some cases, decoder can determine the proper motion vector used for prediction without transmitting additional information. Decoder utilizes the

knowledge that vector was chosen at the encoder side on the basis of Minimum Bitrate Prediction. As a result, 0, 1 or 2 bits are needed for each component of motion vector in order to represent the prediction type. In special cases, median component-wise prediction can be performed as well. The following example shows when no extra bits are required in order to chose the best prediction of current component of motion vector.

Assume the following data at the decoder’s side:

- values of components of neighboring motion vectors: $mv_{1x}=2$, $mv_{2x}=6$, $mv_{3x}=11$,
- the value of received residual $\Delta mv_x=3$.

Therefore, possible values of components of current motion vector are 5, 9 and 14 respectively. However, because the Minimum Bitrate Prediction criterion was used, the only valid and non-ambiguous value is 14, and the best determined prediction component of current vector is mv_{3x} . In all other cases, the chosen predictor would not fulfill the Minimum Bitrate Prediction criterion. This is illustrated in Fig. 3.18.

In order to signal the chosen prediction type in cases when the context of the current motion vector does not allow for determining the prediction value, one-bit or two-bit codeword is transmitted. Therefore, up to four prediction types can be chosen: prediction from neighboring block A, prediction from neighboring block B, prediction from neighboring block C and median prediction using vectors from blocks A, B and C.

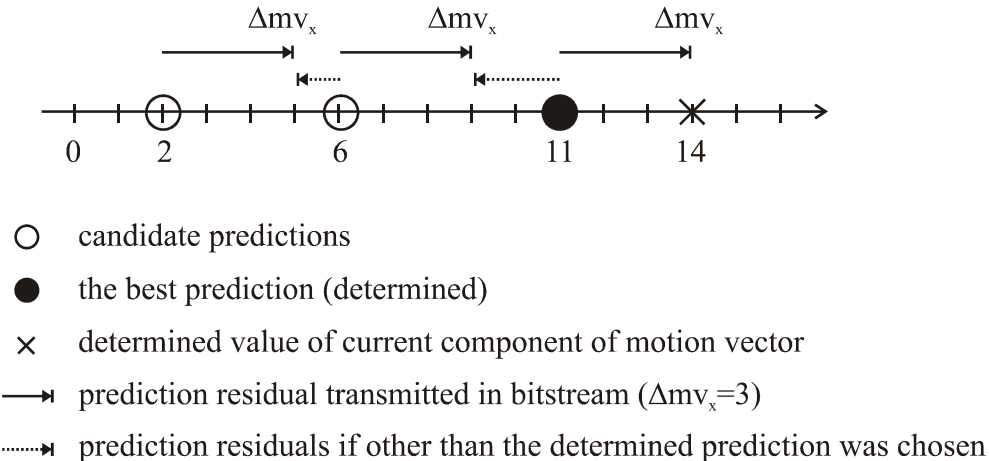


Fig. 3.18. Determination of the value of component of current motion vector in MBP method.

Minimum Bitrate Prediction method was applied into MPEG-4 Visual video codec. The authors reported up to 10% saving in bitrate of motion vectors when compared against standard method of motion vectors encoding used in MPEG-4 Visual video codec. The best results were obtained in the video sequences with fast and complex motion.

The MBP algorithm was further extended by Moon et al [Moo06]. They introduced an idea of joint coding of components of each motion vector using single codeword. A decision of whether to use MBP coding of motion vector or whether to code the components of motion vector together with single codeword is taken based on the local contents of a video frame. In areas with fast and complex motion MBP method is used, while in areas with slow and homogeneous motion the latter method is used.

The algorithm by Moon et al was implemented also in MPEG-4 Visual video codec. Tests were performed for relatively low values of quantization parameter Q_P ($Q_P=8$ and $Q_P=10$). In all cases it outperforms the original MBP method. Bit savings in motion vectors bitstream were up to 34.4 % (video sequence *Container*) as compared to the original method of motion vectors coding used in MPEG-4 Visual codec. Unfortunately, authors did not report the overall bitrate and the value of PSNR achieved. Results for higher values of Q_P were not reported as well.

3.4. Conclusions and summary

Advanced techniques of compression of video require very accurate algorithms of motion-compensated prediction in order to minimize prediction residuals. Since the beginnings of video compression, the motion model has become more sophisticated and more complex. More accurate motion model allows for more efficient representation of a video signal. On the other hand, it also requires more computational power and more sophisticated ways of the representation of the estimated motion parameters, such as motion vectors or reference pictures indices.

In most advanced video coding algorithms, like VC-1 or AVC/H.264, the following techniques are used in order to perform motion-compensated prediction:

- variable block size,
- fractional motion vectors,
- motion vectors over pictures boundaries,

- multi-hypothesis prediction,
- advanced inference of motion information.

These techniques significantly improve the compression efficiency of the modern video codecs comparing to their predecessors.

However, in order to represent this extensive motion model, new techniques of encoding of motion data had to be developed. Among others, median prediction of motion vectors has been most widely utilized and is used in both: AVC/H.264 and VC-1 video codecs.

In the following chapter, a detailed description of the techniques of motion vectors encoding in state-of-the-art video codecs is given. The efficiency of applied solutions is experimentally researched. New techniques of vector median prediction are proposed by the author and they are compared against the existing component-wise solutions.

Chapter 4.

Advanced coding of motion vectors

4.1. Introduction

In the most advanced video coders, the motion model is very complex. In order to perform motion-compensated prediction, each block is described by a set of the following parameters:

- accuracy of estimated motion vector,
- size of block,
- temporal prediction mode,
- motion vector components,
- index of the reference frame used for prediction.

Furthermore, frames encoded using motion-compensated prediction may contain macroblocks that are intra-coded, hence motion vector field becomes heterogeneous. Therefore, advanced techniques have to be used in order to represent efficiently the motion information.

Usually, the accuracy of motion vector is determined by compression algorithm. However, TML-2 video codec, which was developed by VCEG, utilized Adaptive Motion Accuracy (AMA) technique, in which the accuracy of motion vector was adapted for each block [Wie03]. Other proposals assume adaptation of motion vector accuracy at frame level [Rib99], for example in VC-1, $\frac{1}{2}$ -pixel and $\frac{1}{4}$ -pixel accuracy can be chosen independently for each frame [SMP05]. In AVC/H.264 motion vectors are always represented with $\frac{1}{4}$ -pixel accuracy.

Tab. 4.1. Inter-frame coded macroblock types in P-pictures in AVC/H.264 [ISO06].

name of macroblock type	number of partitions	partitions width (in luminance samples)	partitions height (in luminance samples)
P_L0_16x16	1	16	16
P_L0_L0_16x8	2	16	8
P_L0_L0_8x16	2	8	16
P_8x8	4	8	8
P_8x8ref0	4	8	8
P_Skip	1	16	16

Partitioning of the macroblock and macroblock prediction mode are determined by macroblock type. For example, six inter-frame macroblock types are defined in P-pictures in AVC/H.264 as showed in Tab. 4.1. Additionally, in macroblock types named P_8x8 and P_8x8ref0, each partition of 8×8 luminance samples can be further divided into smaller blocks: 8×4, 4×8 and 4×4 luminance samples. As a result, macroblock can be divided in 259 ways. The idea of dividing the macroblock into partitions is given in Fig. 4.1.

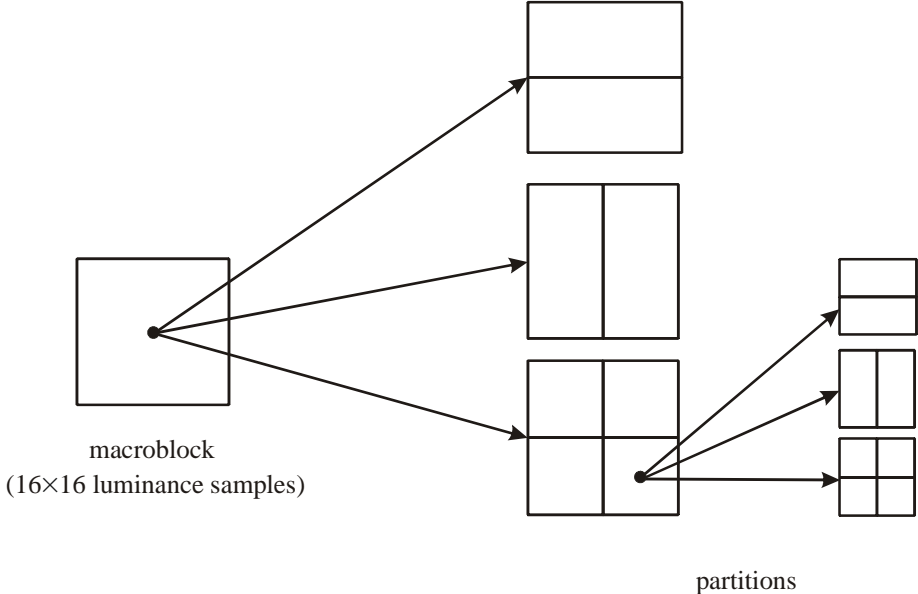


Fig. 4.1. Partitioning of macroblock in AVC/H.264 video coder.

Motion vectors and reference frame indices for all blocks can be inferred using adjacent motion vectors or explicitly transmitted in a bitstream.

In order to efficiently represent motion vectors in advanced hybrid video coders, some assumptions have to be made regarding the nature of motion field. When the minimum size of predicted block is 4×4 luma samples, it is assumed that in pictures coded with motion-compensated prediction, motion vector and index of reference frame are assigned to each 4×4 luma block, regardless the block prediction mode and partition size. Furthermore, in pictures coded using bidirectional prediction, two motion vectors and two indices of reference frames are assigned to each 4×4 luma block. It allows for uniform referencing to the motion information for all blocks, regardless of adjacent blocks size and prediction modes. An example of motion vector assignment is given in Fig 4.2.

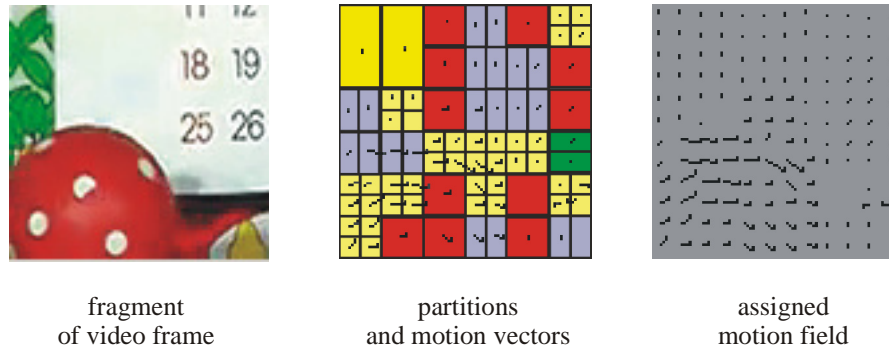


Fig. 4.2. Motion vector assignment in a fragment of frame 32 from sequence *Mobile* 176×144 in AVC/H.264 video codec.

When motion vector field is defined over the whole picture area (a motion vector is assigned to each block), motion vectors can be efficiently represented using mechanism of spatial prediction and differential coding.

In the most advanced video codecs, component-wise prediction of motion vectors is performed. In the following section, motion vector prediction scheme that is used in AVC/H.264 video codec is described. Another state-of-the-art video codec VC-1 uses similar, median-based methods of motion vector prediction as well.

In filtering of vector-valued signals, usually better properties are achieved when vector median filtering is performed [Ast90, Arg91]. Therefore, alternative, original scheme of motion vectors prediction has been proposed and tested in Section 4.3. In the proposed method, vector median has been used instead of component-wise median. The prediction efficiency of the original proposal against the existing technique has been experimentally compared. The experiments should show whether vector-based approach to prediction of motion vector is better than component-wise approach. The results are presented in Section 4.3.2.

In Section 4.4, entropy coding of motion vector prediction residuals in AVC/H.264 codec is described. Two alternative methods are presented. An comparison of efficiency of adaptive arithmetic coder CABAC against variable length coding using Exp-Golomb codes is performed in Section 4.4.3.

The experiments described in this section should give the answer for the question how efficient are the existing techniques of motion vectors prediction and motion vectors coding. The proposals of improving the techniques of motion vectors coding in

the case of multiresolution representation of video sequence, which are presented in the following chapters, base on the results and conclusions obtained in this chapter.

4.2. Component-wise prediction

Very complex and sophisticated motion model is used in AVC/H.264 [ISO06], VC-1 [SMP05] and AVS [AVS06] video coding algorithms,. However, in AVC/H.264 video codec the most advanced motion model has been incorporated. Therefore, a representation of motion vectors in AVC/H.264 has been described precisely in this dissertation. In both: VC-1 and AVS codecs prediction of motion vectors is somewhat simpler.

Two motion vector prediction schemes are defined in AVC/H.264: directional prediction and median prediction. Directional prediction is utilized when macroblock is divided into two rectangular partitions of 16×8 or 8×16 luma samples. Otherwise median prediction scheme is used.

4.2.1. Median prediction of motion vectors

The scheme of median prediction of motion vectors is used for all sizes of partitions in AVC/H.264 except the rectangular partitions of 16×8 and 8×16 luminance samples, as described in preceding section. The scheme is also partially utilized during inferring motion information in special modes of macroblock prediction, when no other motion data is sent (SKIP and DIRECT modes).

Median prediction is performed in four steps (see Fig. 4.3. for marking of blocks):

- when block C is not available, it is replaced by block D for further operations,
- when any of the neighboring blocks A, B or C is intra coded or does not use motion compensated prediction from the same temporal direction (backward or forward) as current block, it is assumed that it uses abstract, not-existing, reference picture for prediction,
- when the only available neighboring block is block A, its motion vector is directly taken as a prediction and the prediction is finished,

- otherwise, when there is one and only block (among A, B and C) that uses the same reference frame as current partition, its motion vector is directly taken as a prediction and the prediction is finished,
- otherwise, scalar median filtering of the motion vector components of blocks A, B and C is performed and the result is taken as the predicted motion vector.

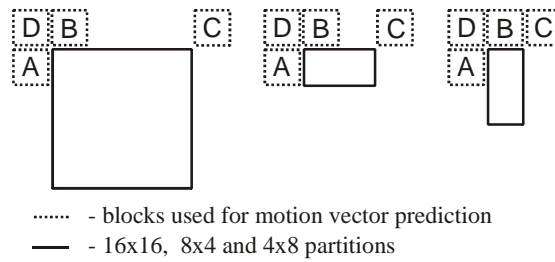


Fig. 4.3. Blocks used for motion vector prediction when median motion vector prediction is performed for various sizes of partition.

Tab. 4.2. The result of motion vector prediction, depending on availability of a adjacent macroblock and reference frame used in inter-frame prediction.

adjacent block				predicted motion vector
A	B	C	D	
a	b,x	b,x	b,x	A
b,x	a	b,x	b,x	B
b,x	b,x	a		C
b,x	b,x	x	a	D
a	a	x	a	med(A,B,D)
otherwise				med(A,B,C)

a – available (the same reference frame),
 b – available (different reference frame or intra coded),
 x – unavailable,
 when block is unavailable or intra coded, both motion vector components are equal to 0.

In Tab. 4.2 possible predictors for the current motion vectors are shown, depending on availability and reference picture, which is used in inter-frame prediction.

After prediction, residual values of motion vectors component are entropy coded in the bitstream.

4.2.2. Directional predictions of motion vectors

Motion-compensated prediction using rectangular blocks of 16×8 and 8×16 is designed in AVC/H.264 exclusively for efficient representation of motion in the case of directional displacements within a macroblock. First-order, directional motion vector prediction is used in this case. Blocks used during prediction are shown in Fig. 4.4.

In the 16×8 mode (horizontal motion within macroblock) motion vector for top partition is predicted from top neighbor (B in Fig. 4.4). Motion vector for bottom partition is predicted from left neighbor (A on Fig. 4.4); that is because the bottom neighbor of current macroblock is not available for prediction yet, as it has not been decoded so far.

In 8×16 mode (vertical motion within macroblock) motion vector for left partition is predicted from left direction, using its left neighboring block (C in Fig. 4.4). Motion vector for right partition is predicted from its up-right neighbor (D in Fig. 4.4). Right neighbor of the right partition has not been decoded so far, thus it can not be used for prediction.

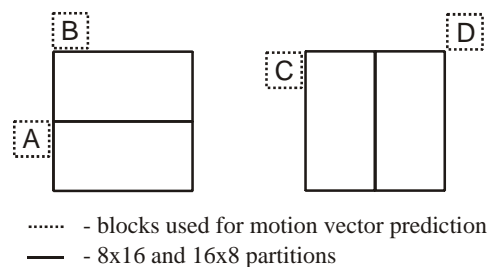


Fig. 4.4. Directional prediction of motion vectors in 16×8 and 8×16 macroblock partitioning modes.

The directional methods of prediction in 16×8 and 8×16 partitions are used only when adjacent macroblocks are available and use the same reference frame for motion compensated prediction. A macroblock is available when it exists in the same slice of macroblocks and is not outside the boundary of coded frame.

When any of the neighboring blocks used normally for prediction is not available or uses different reference frame, median prediction is performed, which is described in the following section.

4.2.3. Component-wise motion vector prediction – experimental results

In order to examine the actual efficiency of motion vectors prediction in AVC/H.264 video codec, a series of experiments have been carried out by the author of this dissertation. The goal of experiments was to research the actual efficiency of motion vectors encoding in AVC/H.264 video codec in order to develop improvements in the advanced encoding scheme. The efficiency of prediction of motion vectors have been tested for several test sequences.

Average residual component of motion vector have been measured and histograms of residual values have been constructed in order to estimate the performance of prediction scheme. Tests were performed for various video sequences and various spatial resolutions. The following sequences were tested: *Ice*, *City*, *Foreman*, *Football*, *Mobile* and *Bus*. Two resolutions have been taken into consideration: 4CIF (704×576), and CIF (352×288).

In the next sections, the following symbols have been used in the tables and in the figures:

$\overline{ \Delta mv }$	– average module of residual of motion vector component, given in units of ¼-samples,
γ	– percentage of the given value of motion vector residual in overall bitstream,
B	– bitrate.

4.2.3.1. Average absolute values of motion vector prediction residuals in 4CIF video sequences

In Tab. 4.3 average absolute values of motion vector prediction residuals in 4CIF test sequences *Ice* and *City* for 3 different bitrate are shown. In Fig. 4.5, a histogram with residuals of motion vectors components is presented for *City* sequence. Experiments were performed using AVC/H.264 reference software version 7.3 [ISO06a]. The following parameters have been set in the configuration file of the encoder for 4CIF sequences:

- period between I frames: 100,
- group of pictures: I-P-P-P,

- number of reference frames: 2,
- entropy coding: CABAC,
- range of motion estimation +/- 64 samples (full-pel units),
- range of bitrate: 700 kbps – 3000 kbps.

Tab. 4.3. Average prediction error module for motion vector components in *Ice* and *City* sequences (704×576, IPPP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

	bitrate (kbps)	PSNR (dB)	$ \overline{\Delta mv} $	motion vectors with $ \overline{\Delta mv} > 10$ (%)
CITY	758.5	34.54	1.644	4.1
	1537.8	38.18	2.594	6.5
	3164.5	41.48	4.017	10.5
ICE	761.8	42.39	3.392	8.3
	1562.0	45.64	4.849	11.8
	3082.0	48.4	6.001	14.6

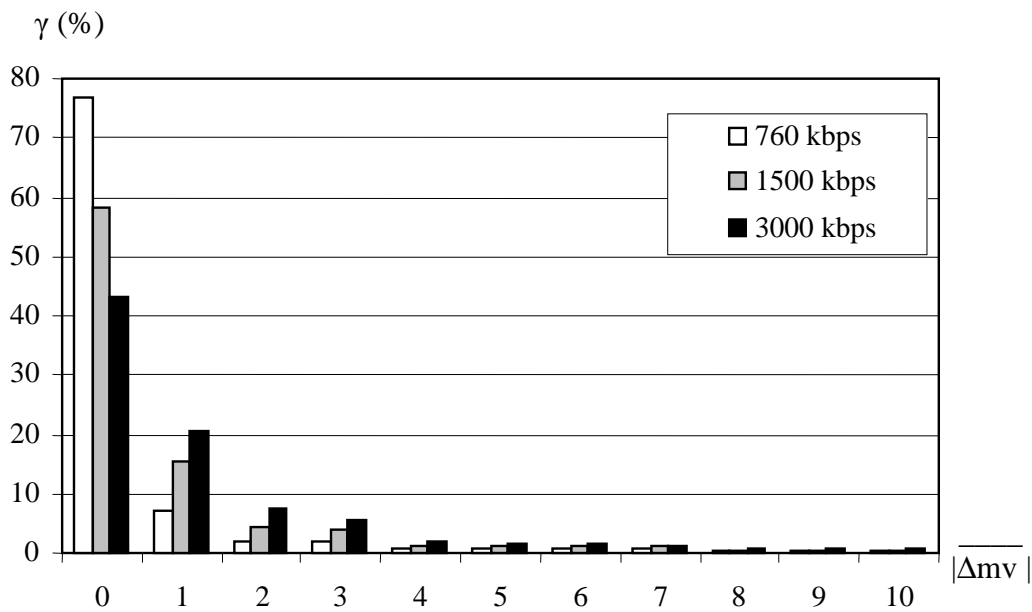


Fig. 4.5. Histogram of motion vector prediction residuals for various bitrate in *City* sequence (704×576, IPPP), AVC/H.264 video codec.

The residuals are given in the units of 1/4-sample.

Experiments performed for 4CIF video sequences *Ice* and *City* revealed high efficiency of motion vector prediction used in AVC/H.264 video codec. Especially for low bitrates (Tab. 4.3), the percentage of low values of motion vector prediction residuals is significant. As an example, histogram in Fig. 4.5 depicts the impact of target bitrate on the motion vector prediction residuals in *City* sequence.

4.2.3.2. Average absolute values of motion vector prediction residuals in CIF video sequences (P-frames)

In Fig. 4.6-4.9 histograms of motion vector prediction residuals for P- frames have been presented. The experiments have been performed by the author for CIF sequences *Mobile*, *Bus*, *Football* and *Foreman*. Experiments were performed using SVC reference software version 4.0 [ISO06b]. The codec was setup to produce non-scalable, AVC/H.264-compliant bitstream. The following parameters have been set in the configuration file of the encoder for CIF sequences:

- period between I frames: 100,
- group of pictures: I-B-P-B,
- number of reference frames: 2,
- entropy coding: CABAC,
- range of motion estimation +/- 64 samples (full-pel units),
- range of quantization parameter Q_p : 31-39.

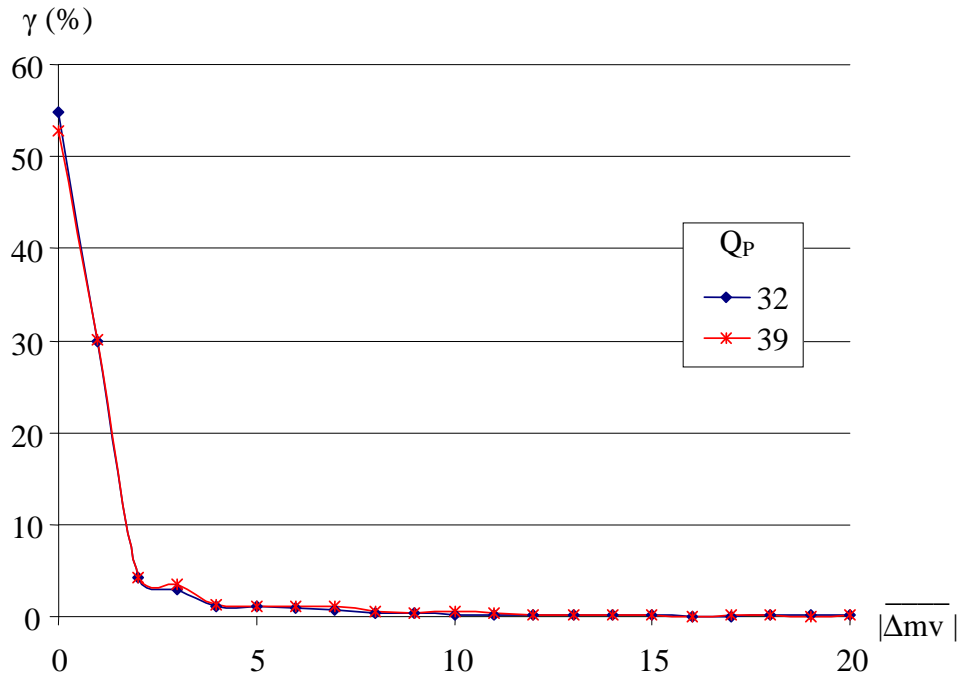


Fig. 4.6. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

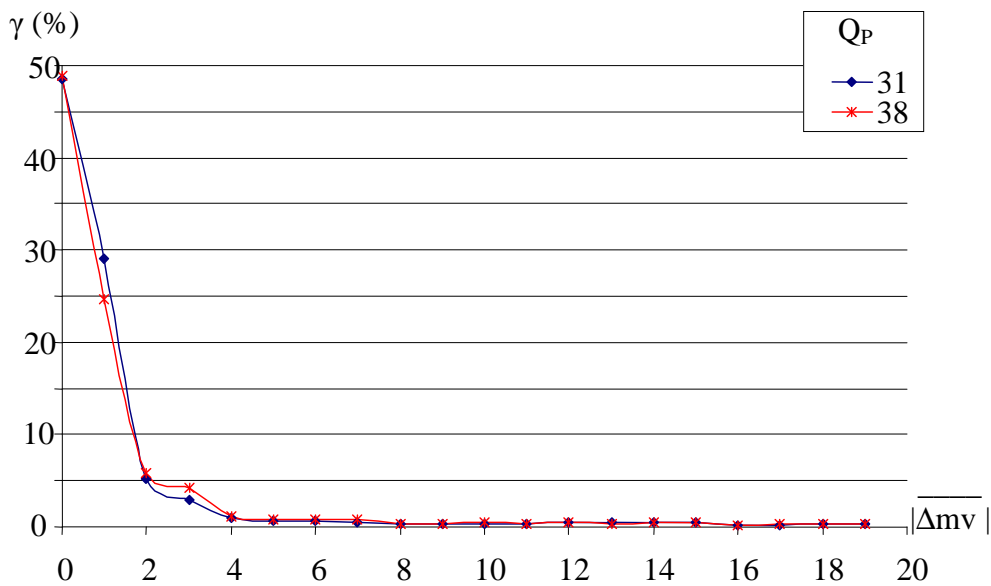


Fig. 4.7. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

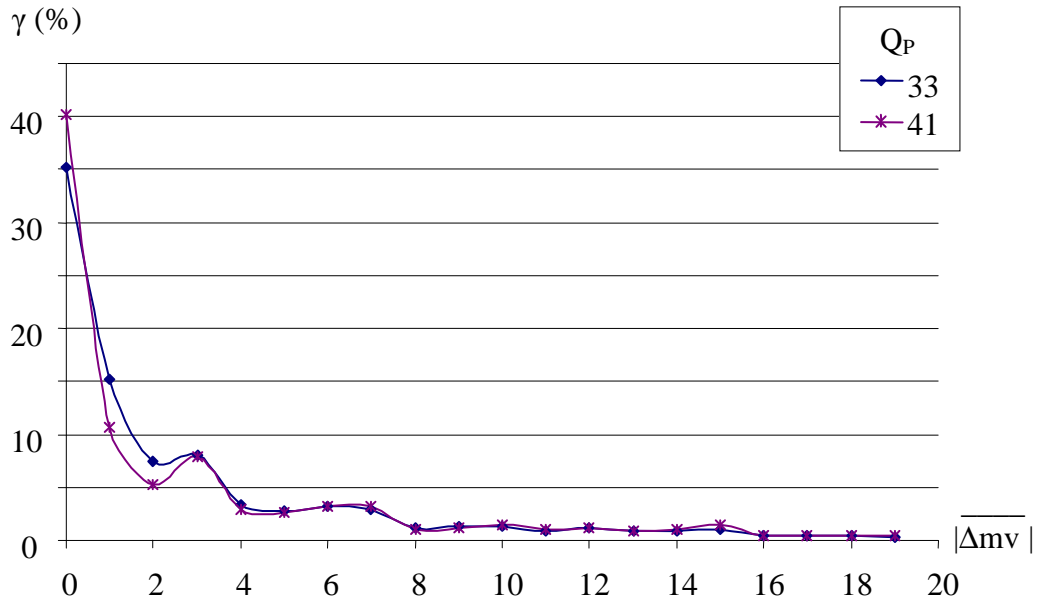


Fig. 4.8. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

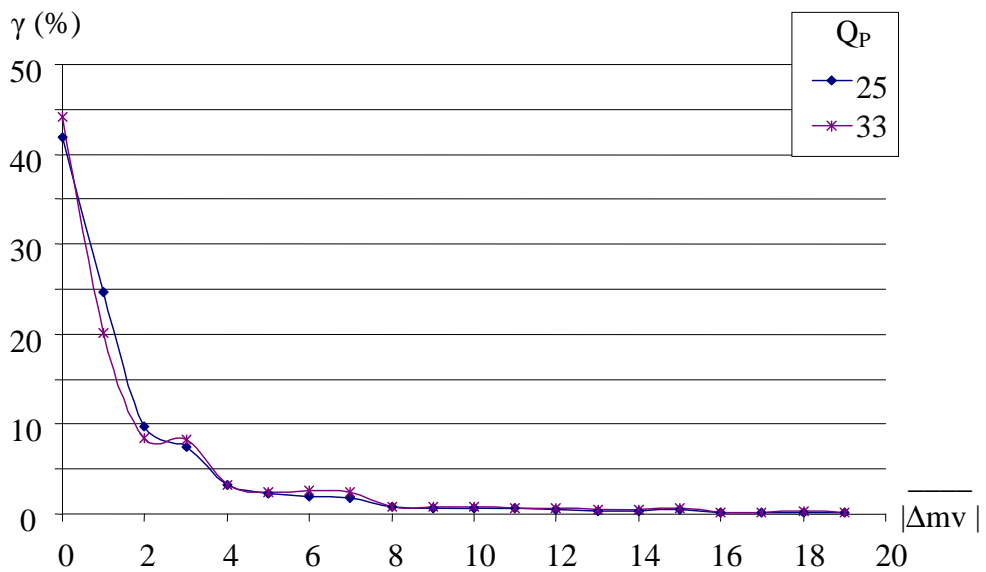


Fig. 4.9. Histogram of motion vector prediction residuals in P-frames for various values of quantization parameter in *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

Experiments performed for CIF video sequences *Mobile*, *Bus*, *Football* and *Foreman* confirm high efficiency of motion vector prediction scheme used in AVC/H.264 video

codec. However, comparing with the results obtained for 4CIF sequences, there have been observed lower differences between the values of motion vector residuals for low and high bitrates (high and low values of quantization parameter Q_P , respectively).

4.2.3.3. Average absolute values of motion vector prediction residuals in CIF video sequences (B-frames)

In Fig. 4.10-4.13 histograms of motion vector prediction residuals for B- frames have been presented. Experiments have been performed for CIF sequences *Mobile*, *Bus*, *Football* and *Foreman*. The experiments were performed using SVC reference software version 4.0 [ISO06b]. The codec was setup to produce non-scalable, AVC/H.264-compliant bitstream. The following parameters have been set in the configuration file of the encoder for CIF sequences:

- period between I frames: 100,
- group of pictures: I-B-P-B,
- number of reference frames: 2,
- entropy coding: CABAC,
- range of motion estimation +/- 64 samples (full-pel units),
- range of quantization parameter Q_P : 31-39.

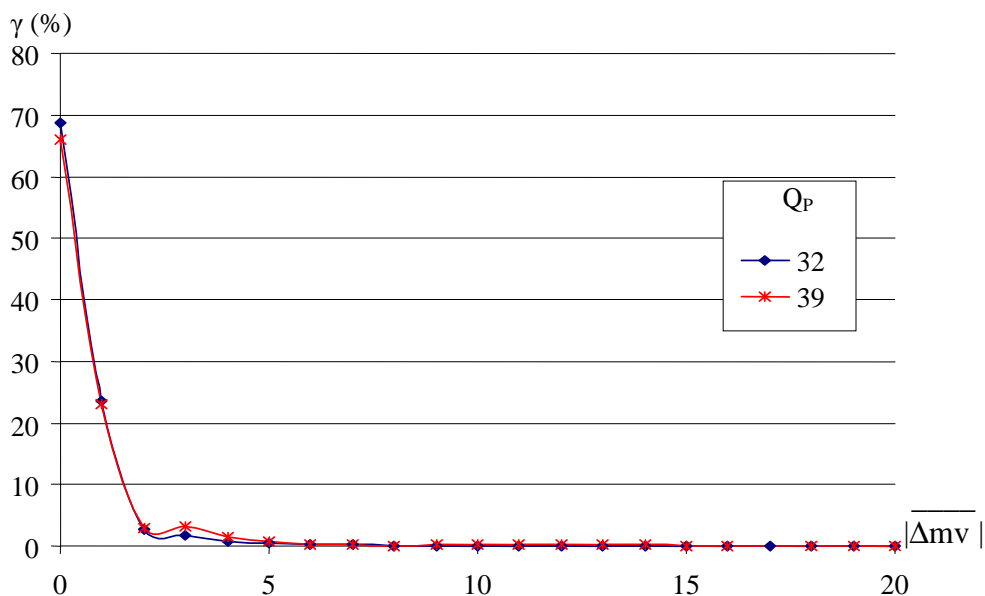


Fig. 4.10. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

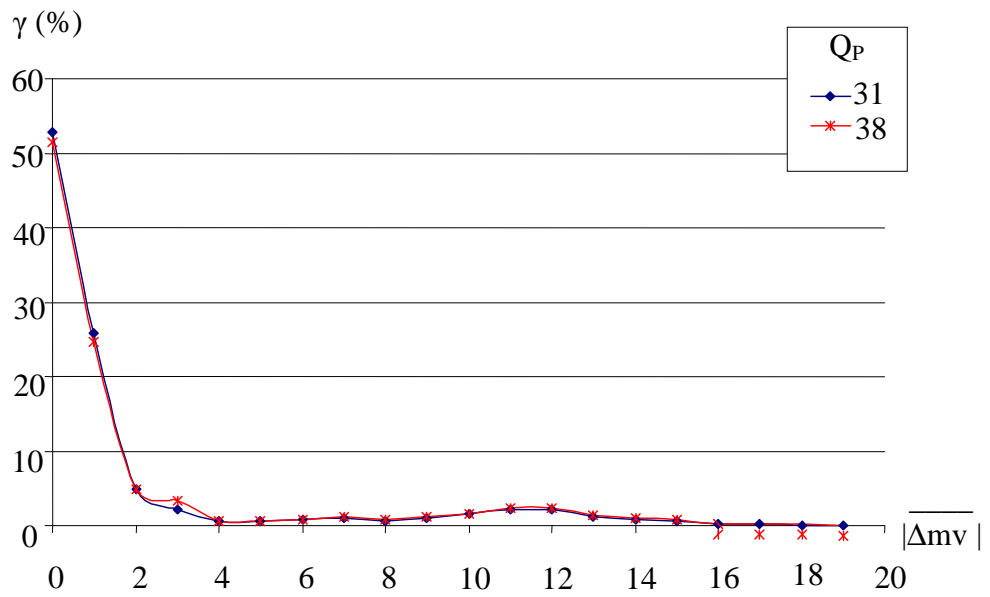


Fig. 4.11. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

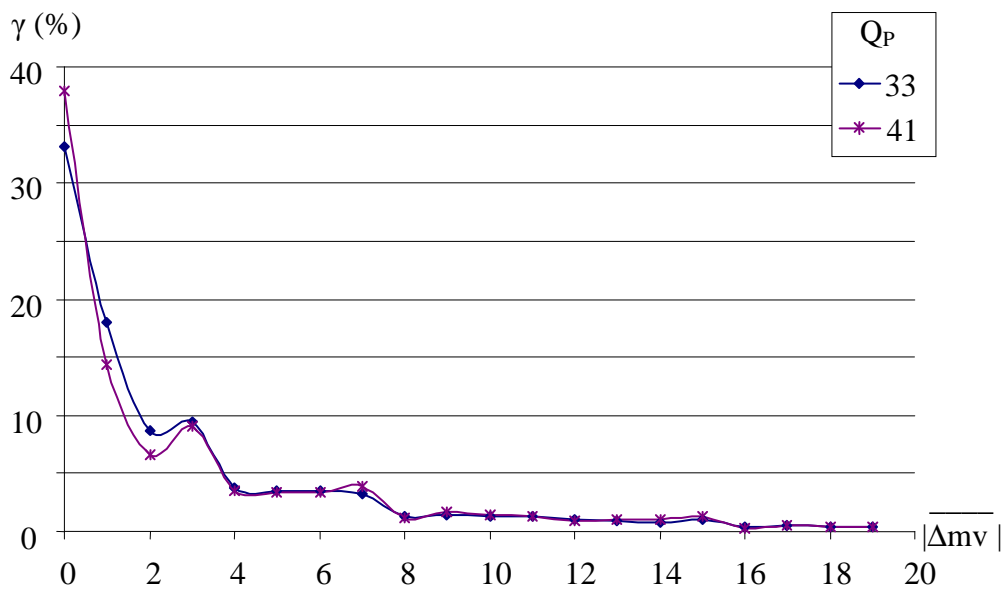


Fig. 4.12. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

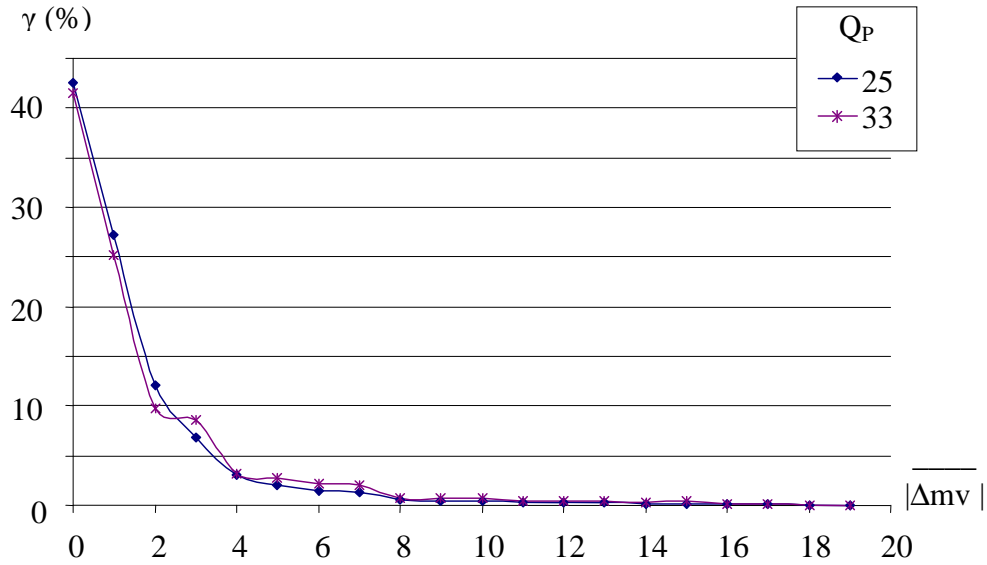


Fig. 4.13. Histogram of motion vector prediction residuals in B-frames for various values of quantization parameter in *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

The values of motion vector prediction residual in B-frames are quite similar to the values obtained in P-frames (Section 4.2.3.2). In sequences with smooth motion (*Mobile* and *Bus*) median prediction performs slightly better in B-frames: there are more motion vector residuals with lower values in B-frames than in P-frames (compare Fig. 4.6 vs. Fig 4.10 and Fig. 4.7 vs. Fig.4.11).

4.2.4. Component-wise motion vector prediction – conclusions

The adaptive scheme of motion vector prediction defined in AVC/H.264 works very well. This means that the residual motion vectors information in most cases is very close to zero. The average component-wise prediction errors are concentrated near value 0, thus the residual values are very efficiently represented in the bitstream.

For example, in 4CIF *City* sequence encoded with low bitrate (~760 kbit/s), about 75% of the motion residual in P-frames have the value of zero. In the case of low bitrate, the value of average motion vector residual component is also very small: 1.644 for *City* sequence and 3.392 for *Ice* sequence. With the increase of bitrate, average residual motion information also increases – average motion vector residual component increases

up to 4.017 and 6.001 respectively. That is because the rate-distortions criterion of video encoding allows for more bits to be allocated for motion vectors.

In the case of video sequences with CIF resolution, the percent of zero motion vector prediction errors in P-frames vary from 38% (in *Football*) up to 55% (in *Mobile*). Moreover, in B-frames, the percent of zero motion vector prediction errors reaches 70% (in *Mobile*). It means that the motion vector field in B-frames is smoother than that estimated in P-frames, especially for video sequences that contain smooth motion. Nevertheless, both in P- and B-frames the percent of residual values of motion vector equal to 0 is relatively high.

Median prediction of motion vectors performs best in video sequences with slow and smooth motion like *Mobile* and *City*. The percent of zero-valued components of motion vectors is the highest in such sequences (55%-75%). On the other hand, in video sequences with fast or rough motion, like *Football* or *Foreman*, the percentage of zero-valued motion vectors is slightly lower (32%-45%). However, the average prediction efficiency is still very high, even when fast motion occurs in a sequence.

Since the low values of motion vector residuals are represented very efficiently in the bitstream (as described in the following section), the obtained results of motion vector prediction significantly reduces the total number of bits needed in order to encode motion vectors.

On the other hand, the results presented in this chapter prove that there is only small room for improvements of existing methods of motion vectors prediction. In order to improve the efficiency of encoding of motion vectors, new methods of motion vector prediction should even more minimize the prediction residual. The minimization of motion vector residual can be obtained by better prediction of motion vectors for the cases in which the existing methods produce high values of motion vector residual.

When multiresolution representation of video sequence is considered, the refinement of motion vectors prediction can be achieved by exploiting the motion field from the low-resolution video sequence. Such an approach is presented later, in Chapters 6, 7 and 8.

4.3. Vector median prediction

Using the standard, component-wise motion vector prediction, the obtained values of prediction residuals are very low. However, usually better filtering properties for vector-valued signals has vector-based processing. Further considerations, presented in this section, regard the author's proposal of vector-based median prediction of the motion vector in non-scalable advanced video codec AVC/H.264.

Modified methods of motion vector prediction have been proposed and implemented. The methods exploit vector median prediction instead of component-wise median prediction. Such an approach, although seems quite obvious, has not been yet proposed nor described in literature. The proposals exploit norms l_1 and l_2 in order to estimate vector median.

Experimental comparison of the existing component-wise approach has been performed against proposed vector-based approaches. The experimental results of using vector median on prediction error in motion vectors coding are presented in Section 4.3.2.

4.3.1. Median filtering of vector-valued signals

For vector-valued signals, median filtering that is applied separately for each component of the vector produces unwanted distortions [Ast90]. A result of the component-wise median prediction can be a vector that is not in the set of input vectors. A vector median that uses l_1 and l_2 norms is widely used for eliminating problems with component-wise median filtering.

In image processing median filters perform a nonlinear data smoothing while preserving edges unblurred. Median filters have desirable properties for denoising signals when the noise characteristic is unknown.

The median of N scalars $x_i, i=1,2,\dots,N$ is defined as the value x_{med} such as:

$$\bigwedge_y \sum_{i=1}^N |x_{med} - x_i| \leq \sum_{i=1}^N |y - x_i| \quad (4.1)$$

As the result, x_{med} is always one of the set x_i .

The simplest approach to perform median filtering on vector-valued signals is to process the individual components independently of each other. However this simple method has some drawbacks, for example edge jitter. On the other hand, a result of component-wise vector median can be a vector that does not exist in the input data set.

Another approach for extending the median operation onto vector-valued signals is to use vector norm l_1 or l_2 instead of absolute values operator used in (4.1). Vector median of N vectors $x_i, i=1,2,\dots,N$ is then defined as the value x_{med} such as [Ast90]:

$$\bigwedge_{y=1,\dots,N} \sum_{i=1}^N \|x_{med} - x_i\|_{l_x} \leq \sum_{i=1}^N \|x_j - x_i\|_{l_x} \quad (4.2)$$

where l_x denotes either norm l_1 or l_2 .

The above vector median filtering has proven to be very efficient in processing of vector-valued signals, for example in application of removal of noise [Zhe93, Bar97] or deblurring [Arg91] of colour images. Herein, the author's proposal is to use vector median as an alternative approach to component-wise median for motion vectors prediction.

4.3.2. Vector median motion vector prediction - experimental results

The goal of the following experiments was to compare efficiency of proposed motion vector prediction with vector median against the mostly used component-wise median prediction.

Vector medians with the norms l_1 and l_2 defined in (4.2) have been implemented and used in the spatial motion vectors prediction in AVC/H.264 codec instead of standard component-wise median. Three cases were tested:

- scalar median as defined in AVC/H.264 video coding standard (referred as **SCALAR** in the tables and on the figures),
- vector median calculated using norm l_1 (referred as **MED-L1**),
- vector median calculated using norm l_2 (referred as **MED-L2**).

The experiments have been performed for the CIF sequences *Bus*, *Foreman*, *Football* and *Mobile*. They were performed using SVC reference software version 4.0 [ISO06b]. The codec was setup to produce non-scalable, AVC/H.264-compliant bitstream. The following parameters have been set in the configuration file of the encoder:

- period between I frames: 64,
- group of pictures: I-B-P-B-P,
- number of reference frames: 2,
- entropy coding: CABAC,
- range of bitrate: 200 kbps – 1200 kbps.

For each sequence, the average motion vector prediction error has been measured and the average number of bits per motion vector component has been calculated.

The percentage of the given value of motion vector residual in overall bitstream (parameter γ) is not presented for these experiments, because applied prediction schemes had only little impact on its value, thus the differences in diagrams are unnoticeable.

4.3.2.1. Bitrate and distortion

In Tables 4.4-4.7, bitrate and PSNR are shown for motion vectors prediction using various median filters for sequences *Bus*, *Football*, *Foreman* and *Mobile* respectively.

Tab. 4.4. Bitrate (kbps) and PSNR (dB) in *Bus* (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.

(kbps)/(dB)	QP=31	QP=33	QP=35	QP=37	QP=38
SCALAR	764.4/33.81	578.6/32.34	441.5/31.00	327.9/29.56	291.5/29.00
MED-L1	762.6/33.81	578.6/32.36	441.9/31.01	328.1/29.57	292.5/29.02
MED-L2	762.5/33.80	578.2/32.35	442.3/31.01	327.4/29.58	291.9/29.01

Tab. 4.5. Bitrate (kbps) and PSNR (dB) in *Football* (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.

(kbps)/(dB)	QP=33	QP=35	QP=37	QP=39	QP=41
SCALAR	603.7/33.86	475.9/32.64	361.7/31.36	275.5/30.21	213.3/29.14
MED-L1	604.2/33.86	475.5/32.64	362.5/31.38	276.8/30.21	213.6/29.15
MED-L2	604.9/33.85	476.2/32.64	362.5/31.38	276.2/30.21	213.7/29.16

Tab. 4.6. Bitrate (kbps) and PSNR (dB) in *Foreman* (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.

(kbps)/(dB)	QP=25	QP=27	QP=29	QP=31	QP=33
SCALAR	614.3/39.74	441.5/38.53	330.4/37.39	245.4/36.09	187.2/34.88
MED-L1	614.8/39.74	442.2/38.54	330.5/37.39	245.6/36.09	185.9/34.88
MED-L2	615.6/39.74	441.3/38.53	330.8/37.38	244.8/36.08	185.5/34.88

Tab. 4.7. Bitrate (kbps) and PSNR (dB) in *Mobile* (352×288, IPPP) sequence using various median predictions of motion vector in AVC/H.264 video codec.

(kbps)/(dB)	QP=32	QP=33	QP=35	QP=37	QP=39
SCALAR	961.6/32.15	790.6/31.26	559.3/29.82	373.5/28.27	261.8/26.85
MED-L1	963.1/32.15	790.8/31.26	560.2/29.82	373.9/28.27	262.4/26.85
MED-L2	962.7/32.15	792.3/31.26	559.2/29.82	373.4/28.27	261.8/26.84

The obtained results of overall coding efficiency including achieved bitrate and the value of PSNR parameter are very similar for all researched methods of motion vector prediction. However, in the following sections, more detailed results of motion vectors prediction and motion vectors representation are depicted.

4.3.2.2. Average absolute values of components of motion vector residual

The influence of various median predictions on average absolute values of components of motion vector residual is given in Fig. 4.14 – Fig. 4.21. In Fig. 4.14-4.17 the average motion vector component residual is presented for P-frames. In Fig. 4.18-4.21 the average motion vector component residual is presented for B- frames.

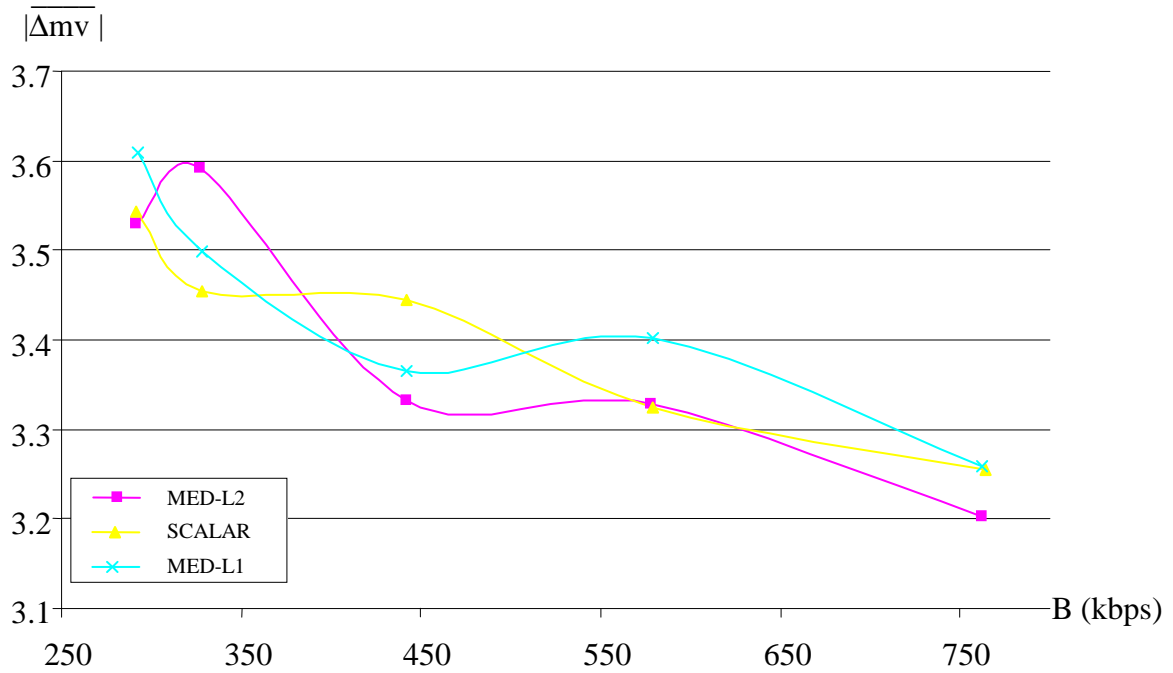


Fig. 4.14. Average absolute values of components of motion vector residual for various median predictions in P-frames, *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

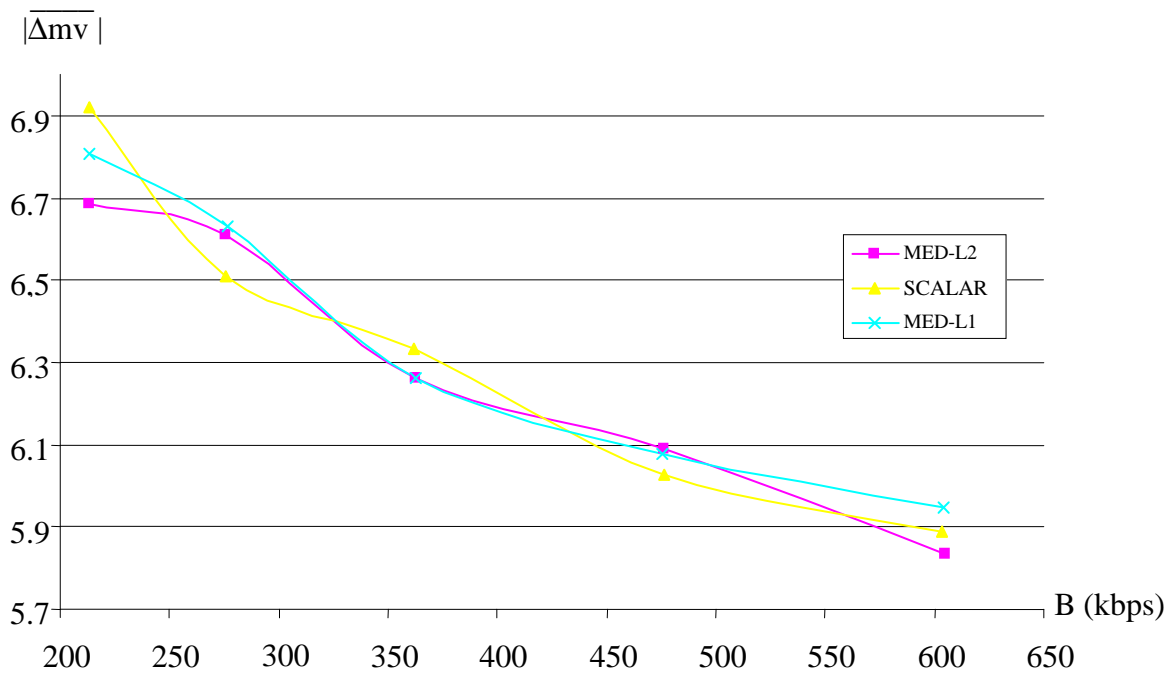


Fig. 4.15. Average absolute values of components of motion vector residual for various median predictions in P-frames, *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

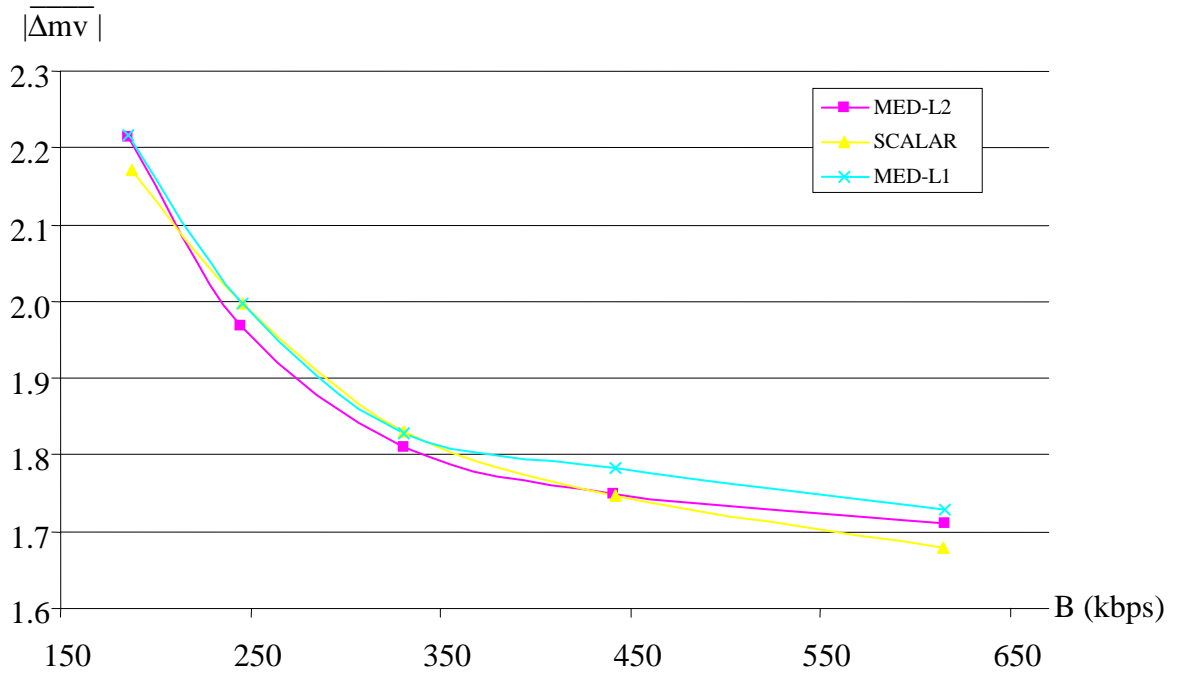


Fig. 4.16. Average absolute values of components of motion vector residual for various median predictions in P-frames, *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

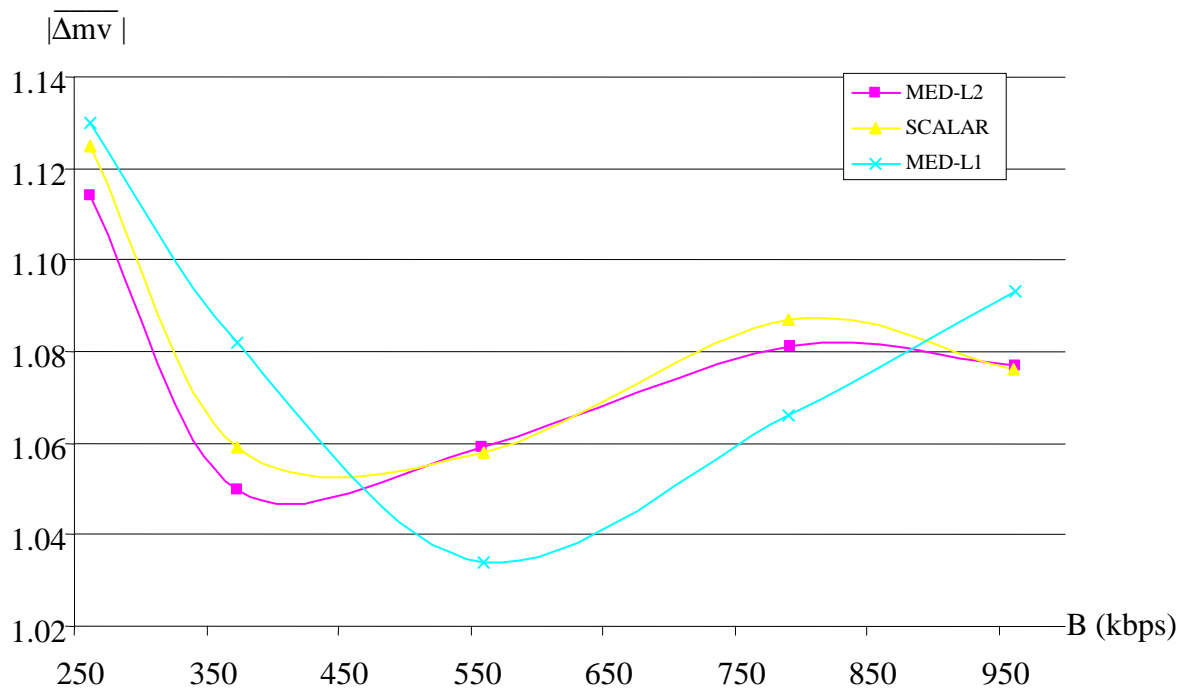


Fig. 4.17. Average absolute values of components of motion vector residual for various median predictions in P-frames, *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

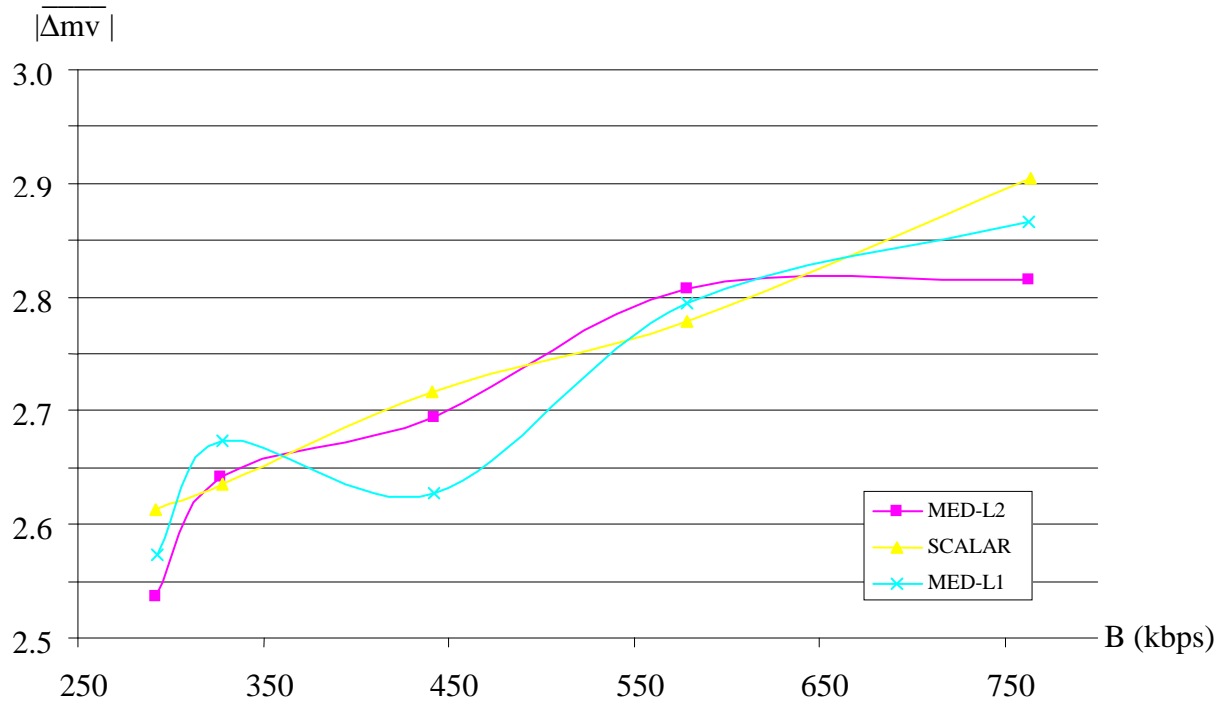


Fig. 4.18. Average absolute values of components of motion vector residual for various median predictions in B-frames, *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

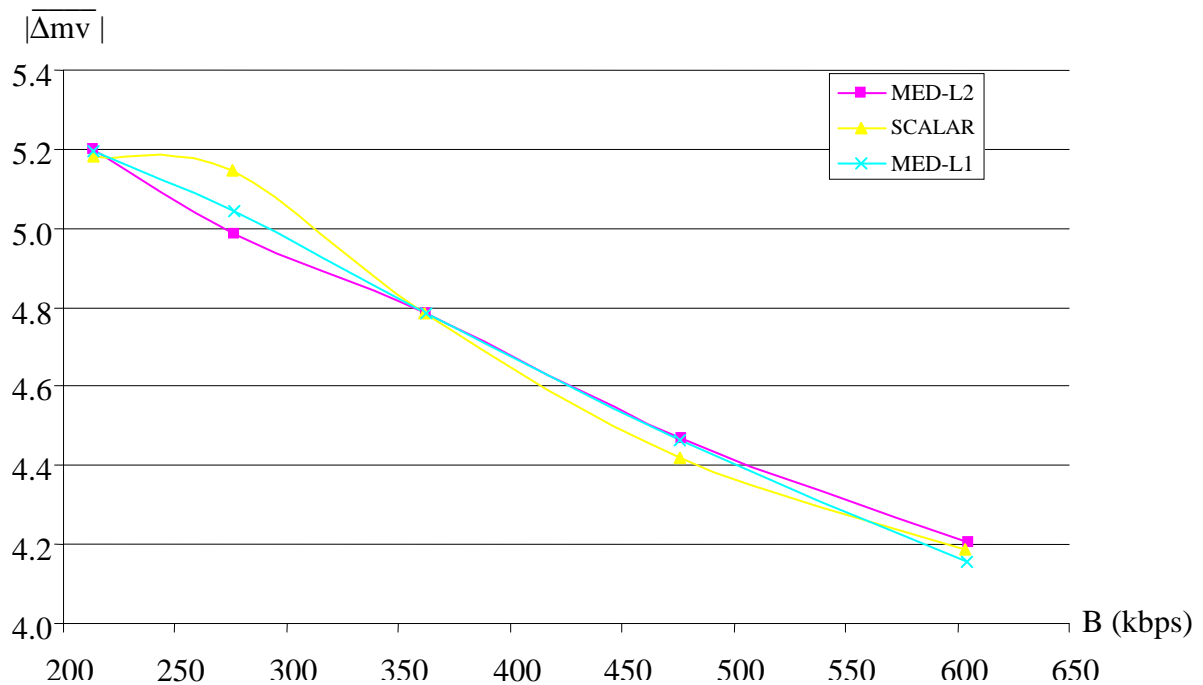


Fig. 4.19. Average absolute values of components of motion vector residual for various median predictions in B-frames, *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

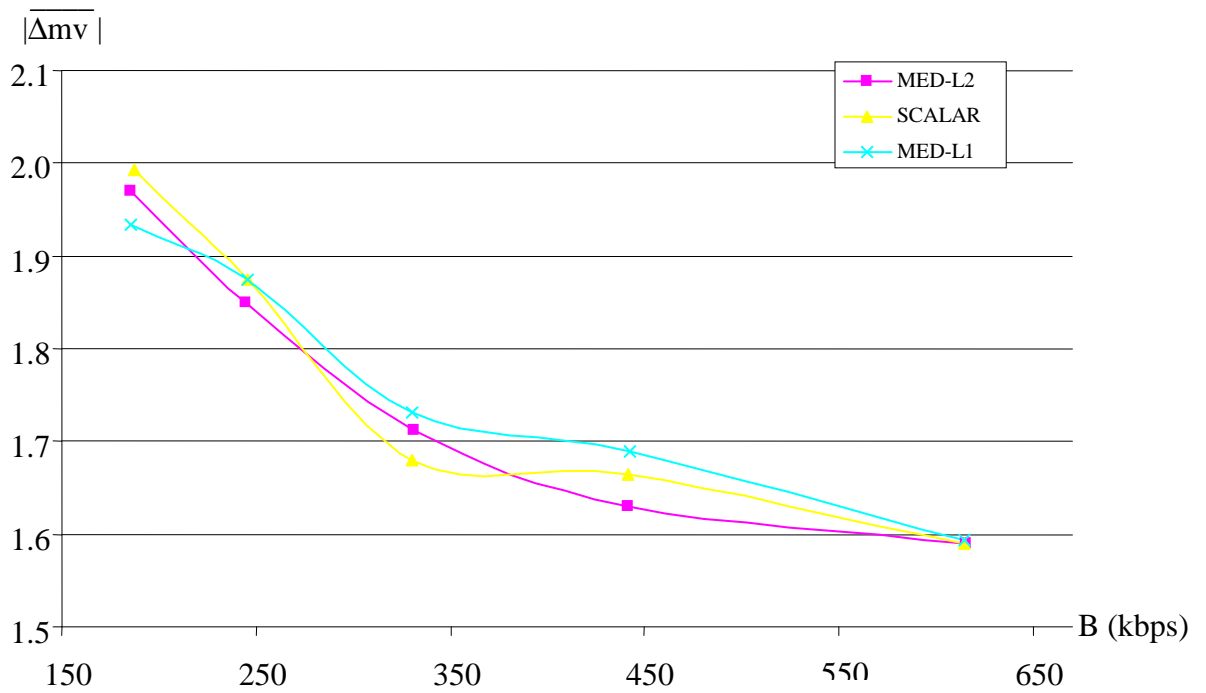


Fig. 4.20. Average absolute values of components of motion vector residual for various median predictions in B-frames, *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

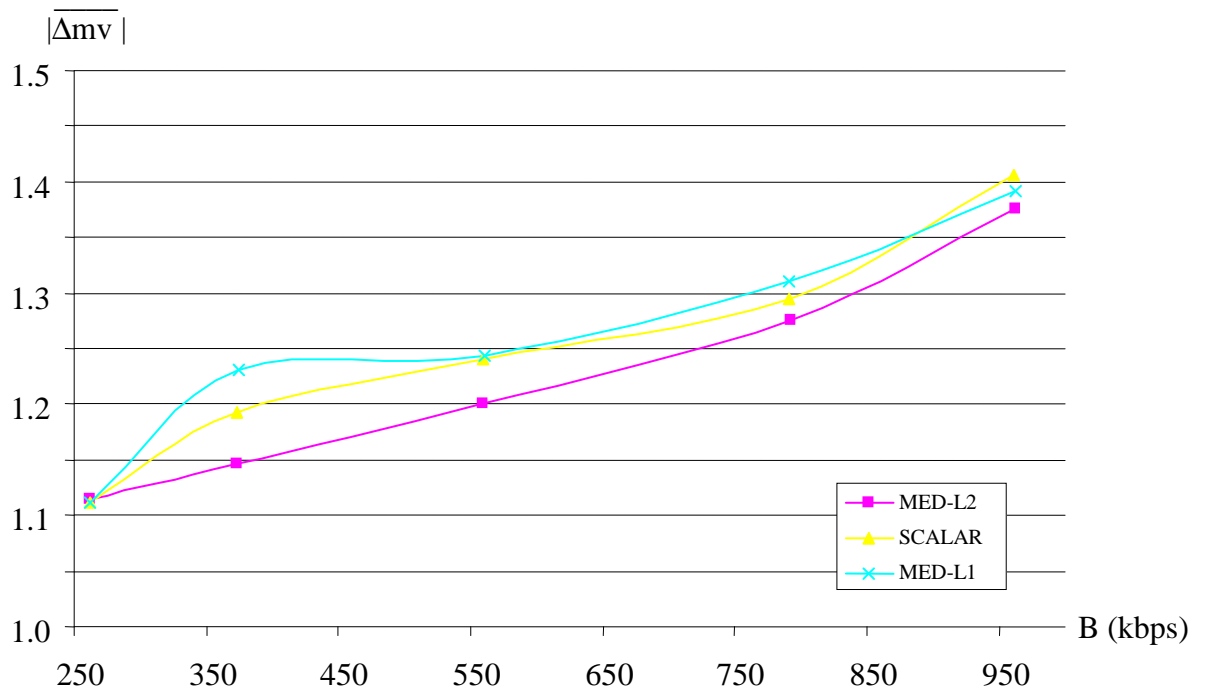


Fig. 4.21. Average absolute values of components of motion vector residual for various median predictions in B-frames, *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

Regardless of the applied method of motion vector prediction, achieved average absolute values of components of motion vector residual are very similar both in P-frames and in B-frames. The difference in obtained results between component-wise prediction scheme and vector-based prediction scheme is very small. Furthermore, the results depend on the video sequence and bitrate. In video sequences with fast and rough motion (*Football*, *Foreman*) the performance of each researched method is almost the same (compare Fig. 4.15, Fig. 4.16, Fig. 4.19 and Fig. 4.20). In such sequences, motion vector prediction residual depends more on local roughness of motion field than on the motion vector prediction scheme.

4.3.2.3. Average number of bits per motion vector component

The diagrams with average number of bits per motion vector component for vector medians and component-wise median prediction has been presented in Fig. 4.22-4.25. The number of bits was estimated by accumulating the number of bits written into a bitstream after encoding of each component of motion vector for all motion vectors that were encoded in the sequence. The obtained value was then divided by the total number of encoded motion vector components.

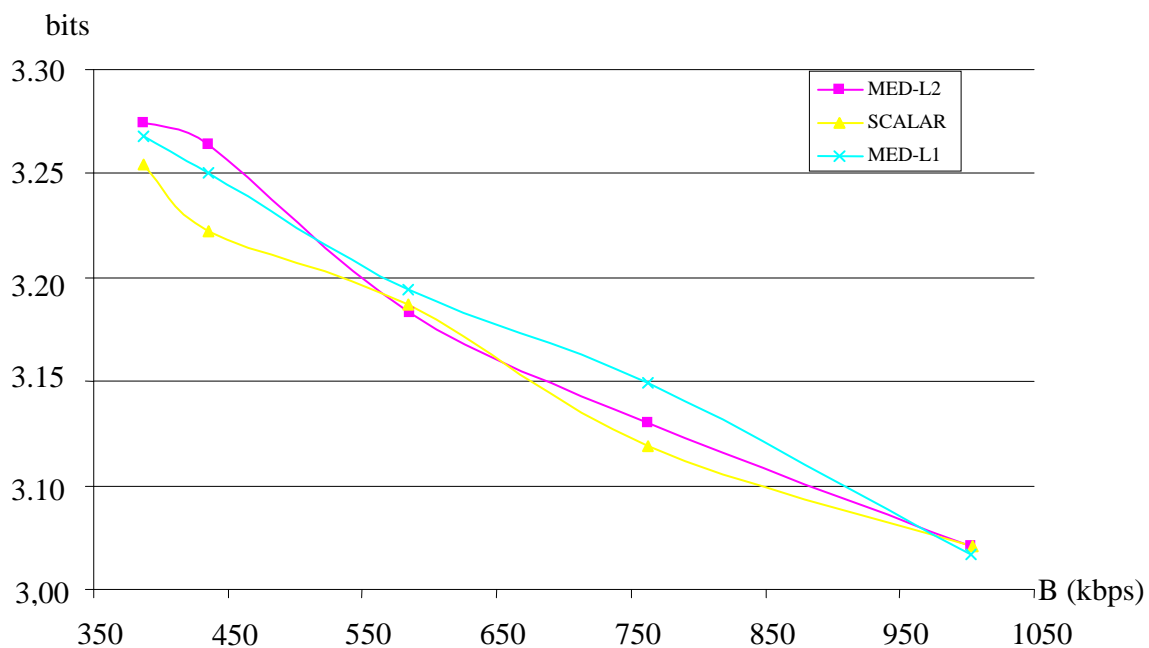


Fig. 4.22. Average number of bits per motion vector component for various median predictions, *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

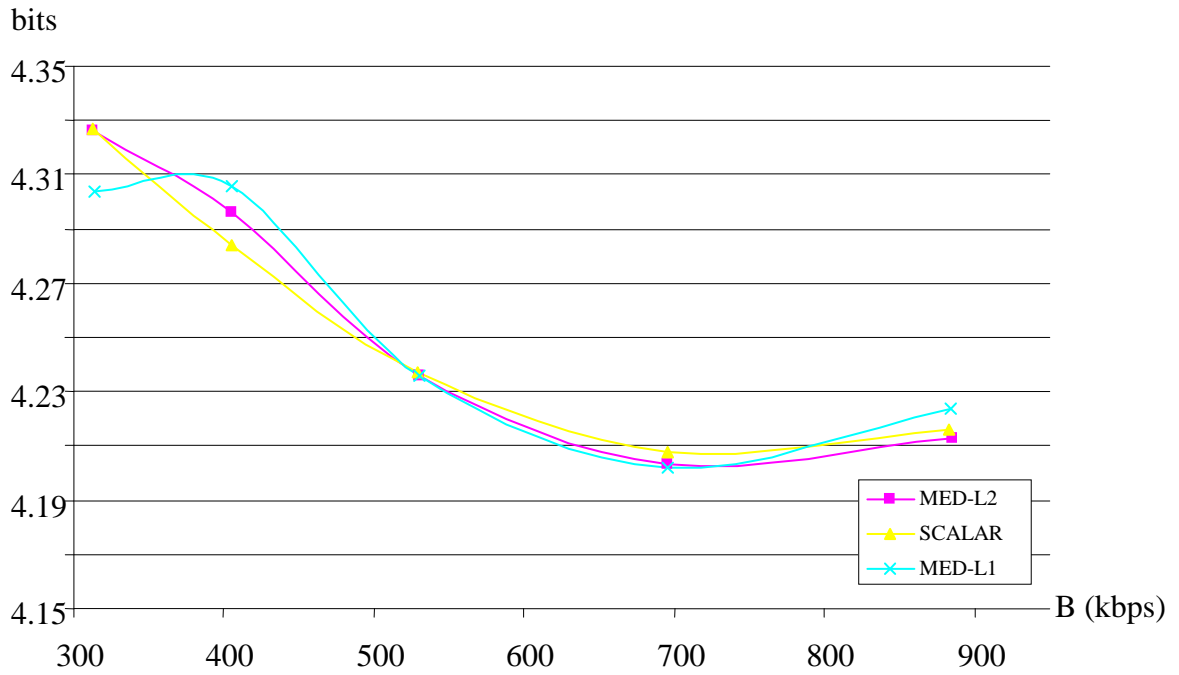


Fig. 4.23. Average number of bits per motion vector component for various median predictions, *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of ¼-sample.

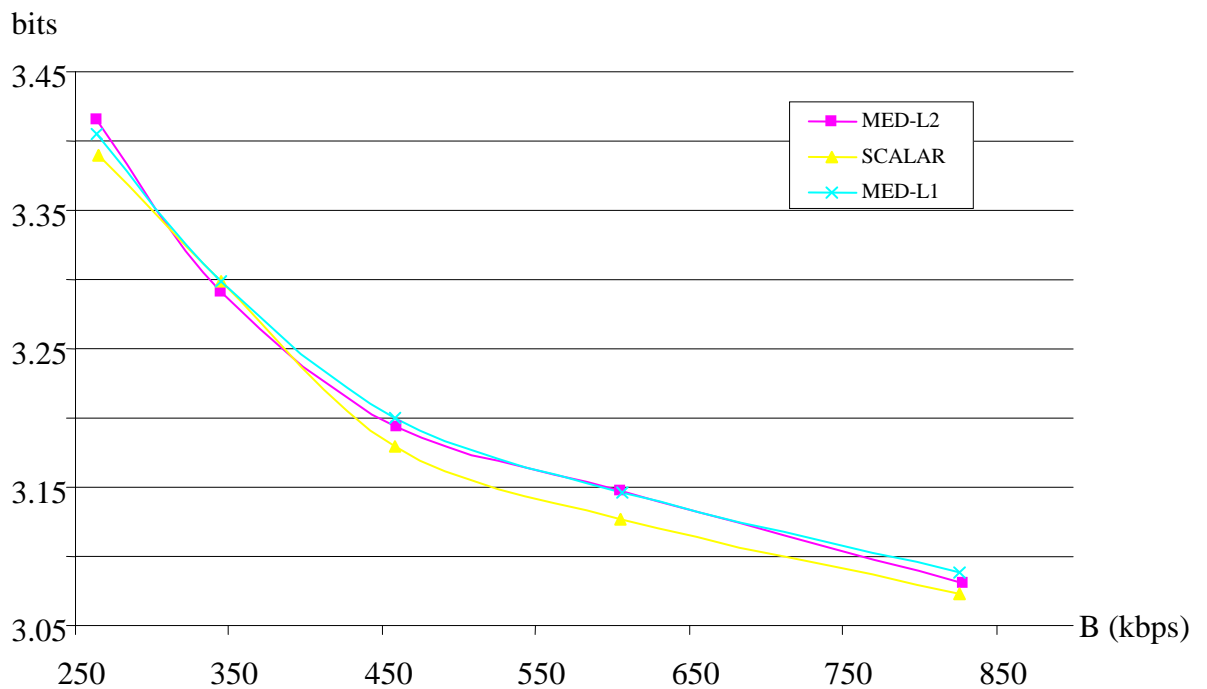


Fig. 4.24. Average number of bits per motion vector component for various median predictions, *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of ¼-sample.

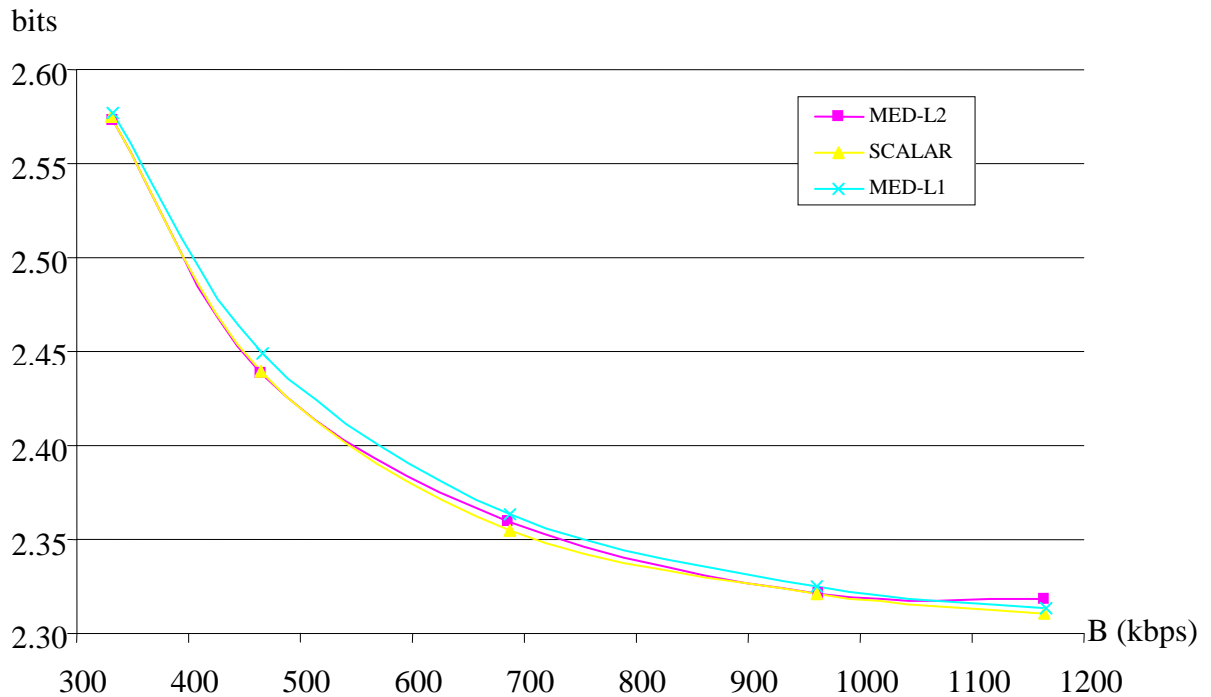


Fig. 4.25. Average number of bits per motion vector component for various median predictions, *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

Again, the difference in average number of bits per motion vector component between component-wise prediction scheme and vector-based prediction scheme is negligible. The obtained values of average number of bits are very similar for all video sequences and for all bitrates.

4.3.3. Vector median motion vector prediction - conclusions

The experimental results prove that all variants of median motion vectors prediction lead to quite similar results expressed by respective rate-distortion curves and average residuals of motion vectors. The average number of bits per motion vector component is also quite similar for all kind of median prediction, as depicted in Fig. 4.22 – 4.25 and depends on the achieved bitrate. It is impossible to determine the conditions (for example the content of a video sequence), for which the vector median prediction outperforms the component-wise median prediction of motion vectors. In the tables 4.4-4.7, the lowest bitrate has been achieved alternately: when component-wise median was

used and vector median with norm l_1 was used. However, the maximum values of PSNR were achieved for vector median with norm l_2 , so the experimental results are ambiguous.

On the other hand, calculations of vector medians are somewhat more complicated than the component-wise median. In Tab. 4.8, the number of operations for component-wise median and vector medians is showed.

Tab. 4.8. Number of operations depending on median type for two-dimensional vector.

	component-wise median	vector median, using l_1 norm	vector median, using l_2 norm
comparison	2 or 3	2 or 3	2 or 3
adding	0	6	6
multiplying	0	0	6
total operations	2 or 3	8 or 9	14 or 15

Replacing the component-wise median by vector median in the module of motion vectors prediction in AVC/H.264 codec has not brought expected improvements. Using the vector median for prediction of motion vectors in hybrid video coders introduces additional operations and does not result in increasing prediction nor compression efficiency.

4.4. Entropy coding of motion vectors

After prediction, motion vector residuals are entropy coded in order to minimize the number of bits needed to represent residual values. Entropy coding exploits the fact that some values are more probable than others, thus can be represented using fewer number of bits [Sal98, Say00, Ohm04].

Usually, variable length coding is used in order to represent motion vectors. Since lower values of motion vectors happen more often, statistical properties of residual data are used during preparation of codebooks. The examples of entropy coding of motion vectors have been given in Sections 3.3.2 and 3.3.3 in the previous chapter.

In AVC/H.264 video coding, there are two alternative methods of entropy coding: coding that uses Universal Variable-Length Codes (UVLC) and Context-Adaptive

Binary Arithmetic Coding (CABAC). In UVLC mode Exp-Golomb (Exponential Golomb) codes are used in order to represent motion vector residuals. In the latter one, more sophisticated mode CABAC, adaptive binary arithmetic coder is used in order to represent motion vector residuals.

Exp-Golomb codes are also used in order to represent motion vector residuals in AVS video coding [AVS06]. They are used in exactly the same manner as in UVLC entropy coding mode of AVC/H.264 coding algorithm [Di03, Fan04].

On the other hand, in another advanced video coding algorithm VC-1, both components of motion vector, as well as the flag indicating the presence of transform coefficients are coded together using single syntax element (MVDATA or BLKMVDATA). They are represented in a bitstream by a variable length codeword followed by two fixed length codewords [SMP05]. The value of variable size code determines the length of the following fixed length codewords. Tables with variable size codewords were determined empirically [Rib03]. The codewords are specified using 4 tables. Given table used in order to decode motion vector is chosen independently for each video frame. As a result, entropy coding in VC-1 codec is quite similar to that used in MPEG-2 standard, which has been shortly discussed in Section 3.3.2.

In the following sections entropy coding of motion vectors in AVC/H.264 is described in more detail.

4.4.1. Coding of motion vectors using Exp-Golomb codes

Exp-Golomb codes [Teu78] are variable length codes. The number of codewords, thus the ability to represent the coded values, grows exponentially with the length of the code [Wen97, Di03]. The Exp-Golomb codes are constructed regularly, in the following way:

$$\underbrace{0 \dots 0}_M 1 \underbrace{x_{M-1} \dots x_0}_M \quad (4.3)$$

The leading zeros and the first ‘1’ can be regarded as “prefix” of the codeword. The following M bits encode the actual value and can be regarded as “suffix” of the codeword, thus the entire codeword consist of $2M+1$ symbols. A special case is the first

codeword of the codebook that has no leading prefix and no trailing suffix, it consist of just a single ‘1’ symbol.

Exp-Golomb codes are decoded from a bitstream by the detection of a sequence of M zeros followed by symbol “1”. $M+1$ bits of the prefix are discarded and M bits of the suffix are combined to form the binary value $info$. Final value val is then calculated in the following manner [Ric02a]:

$$val = 2^M + info - 1 \quad (4.4)$$

Motion vector residuals are represented using Exp-Golomb codes with signed-mapping scheme [Di03, ISO06]. Decoded value of the given codeword (val) is mapped in order to form the signed value of residual (res) according to the following rule:

$$res = \begin{cases} \left\lceil \frac{1}{2}(val + 1) \right\rceil & \text{for odd values of } val, \\ -\left\lceil \frac{1}{2}(val + 1) \right\rceil & \text{for even values of } val. \end{cases} \quad (4.5)$$

The examples of Exp-Golomb codes and their mapping onto signed values are given in Tab. 4.9.

Tab. 4.9. Examples of Exp-Golomb codewords and their mapping onto signed values.
First 9 codewords are given.

codeword	value val	signed-mapped value res
1	0	0
010	1	1
011	2	-1
00100	3	2
00101	4	-2
00110	5	3
00111	6	-3
0001000	7	4
0001001	8	-4
...

The length n of the Exp-Golomb codeword increases logarithmically with the value res of motion vector residual. The following formula for n has been derived by the author [Lan06d]:

$$n = \begin{cases} 1 & \text{for } res = 0, \\ 2 \lceil \log_2(2|res|) \rceil + 1 & \text{for } res \neq 0, \end{cases} \quad (4.6)$$

where:

- n – the length of the Exp-Golomb codeword,
- res – residual value of the motion vector component.

4.4.2. Coding of motion vectors using CABAC

Context-Adaptive Binary Arithmetic Coding [Mar01, Mar01a] is the most advanced entropy coding technique used in the video coding. It has been developed exclusively for AVC/H.264 codec [Wie03]. CABAC uses binary arithmetic coding engine [Wit87, Hel96, Say00] together with a scheme of adaptation to the local values of coded syntax elements [Sal98]. The following mechanisms are used in order to compress the input values:

- binarization of the input syntax elements,
- context modeling,
- binary arithmetic coding.

The generic block diagram in Fig. 4.26 shows the scheme of encoding residual values of motion vectors using CABAC [Mar03].

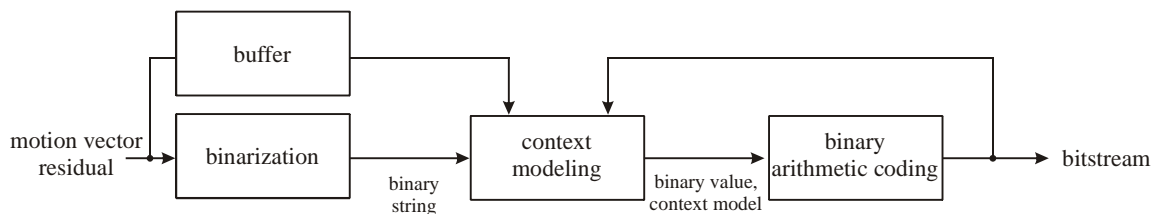


Fig. 4.26. Block diagram of motion vector residual encoding using CABAC entropy coder in AVC/H.264.

In the binarization block, residuals of motion vectors components are converted into strings of binary symbols using unary k-th order Exp-Golomb (UEGk) binarization scheme [Mar01, ISO06]. The probability of occurrence of given binary symbol in the string is estimated in context modeler, based on previously encoded values of motion vector residuals. The results of modeling of the context are input to the binary arithmetic coder and they are represented in the bitstream. The output of arithmetic encoder updates the context modeler in order to match the local probability of occurrence of given value of motion vector residuals.

CABAC allows for significant improvement of coding efficiency in AVC/H.264. It has been reported that for typical video material used in broadcast application, average bitrate savings for overall bitstream are 9%-14% as compared with entropy coding using UVLC codes [Mar03].

4.4.3. Comparison of Exp-Golomb coding against CABAC coding of motion vectors - experimental results

In CABAC it is impossible to estimate a priori the number of bits needed to encode given values of motion vector residuals due to the arithmetic coding with adaptation mechanism. Experimental tests have been performed by the author in order to compare Exp-Golomb coding against CABAC coding of motion vector residuals.

The number of bits written into the bitstream has been measured in AVC/H.264 video coder after coding of each motion vector component residual with the given value. The final result was obtained by averaging the number of bits for specific value of residual component over all motion vectors component written in the bitstream.

The tests have been performed for the CIF sequences *Bus*, *Foreman*, *Football* and *Mobile*. Experiments were performed using SVC reference software version 4.0 [ISO06b]. The codec was setup to produce non-scalable, AVC/H.264-compliant bitstream. The following parameters have been set in the configuration file of the encoder:

- period between I frames: 64,
- group of pictures: I-B-P-B-P,
- number of reference frames: 3,
- entropy coding: CABAC,

- range of quantization parameter Q_p : 25-41.

In Fig. 4.27 – 4.30 the graphs show the average number of bits used for encoding the given values of motion vector residuals in sequences *Bus*, *Foreman*, *Mobile* and *Football* respectively.

The number of bits used for representation of the motion vector residuals has been also measured separately for each frame types: P-frames and B-frames. The example results for *Football* sequence are showed in Fig. 4.31 (for P-frames) and in Fig. 4.32 (for B-frames).

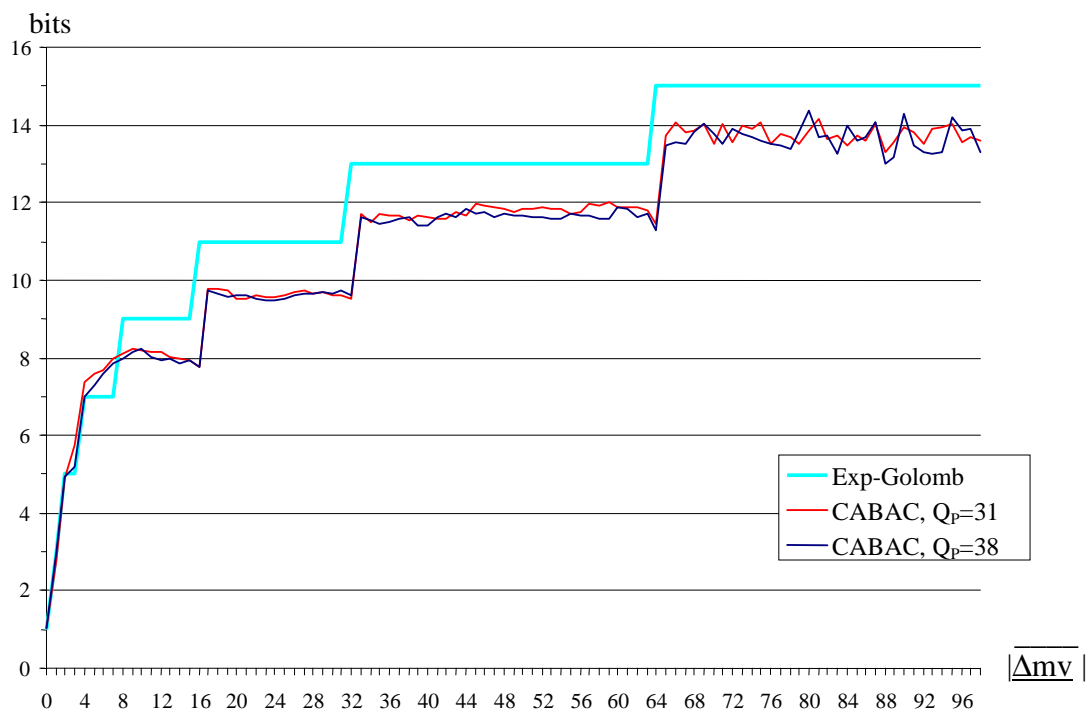


Fig. 4.27. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in *Bus* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

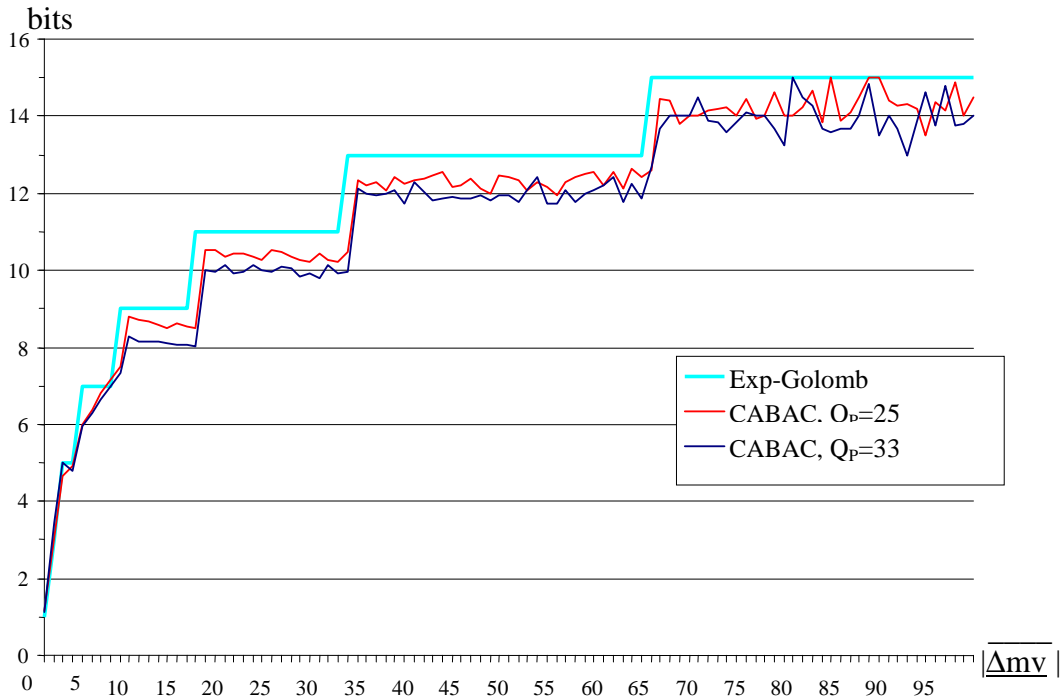


Fig. 4.28. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in *Foreman* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

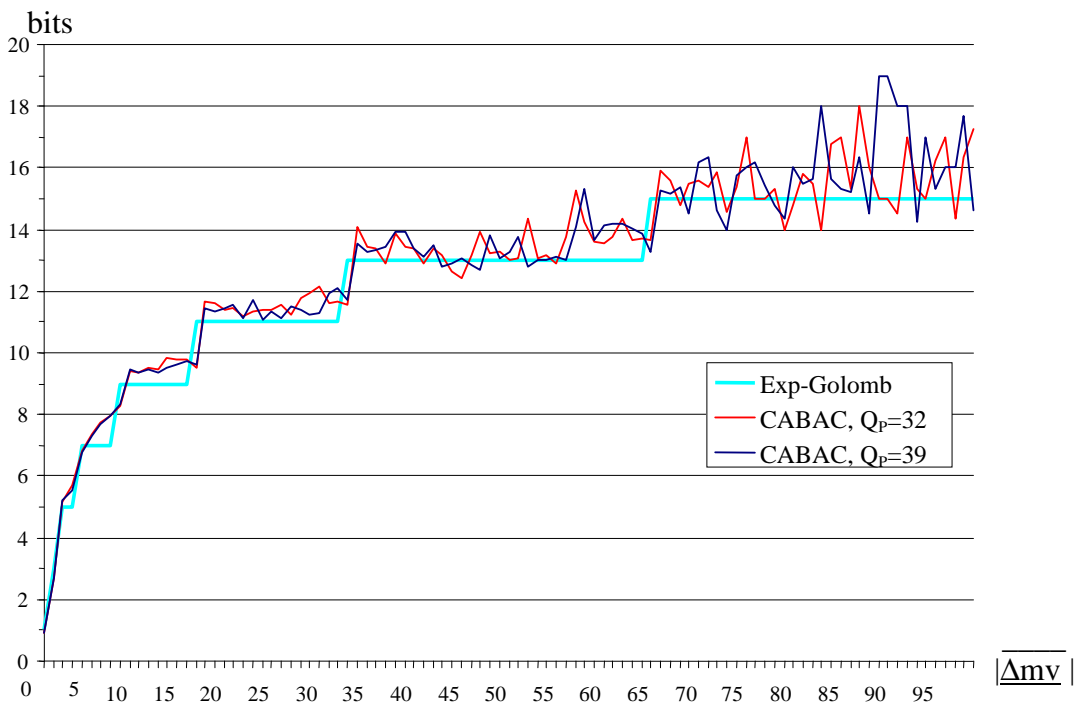


Fig. 4.29. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in *Mobile* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

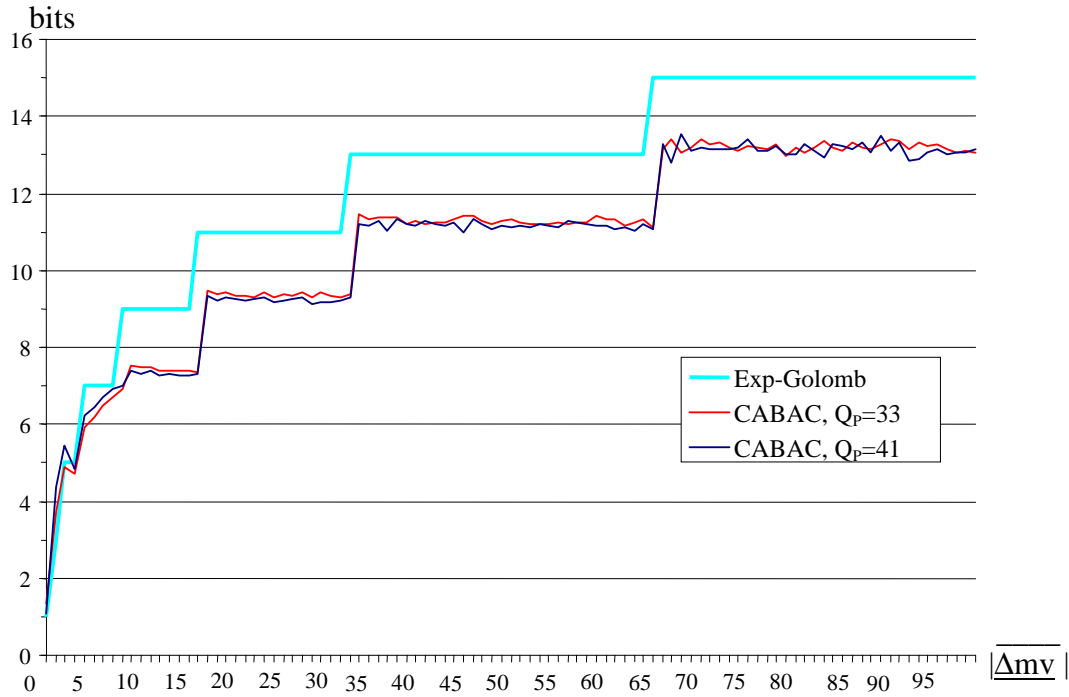


Fig. 4.30. Number of bits for residual motion vector component. Comparison of Exp-Golomb codes against CABAC entropy coding in *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

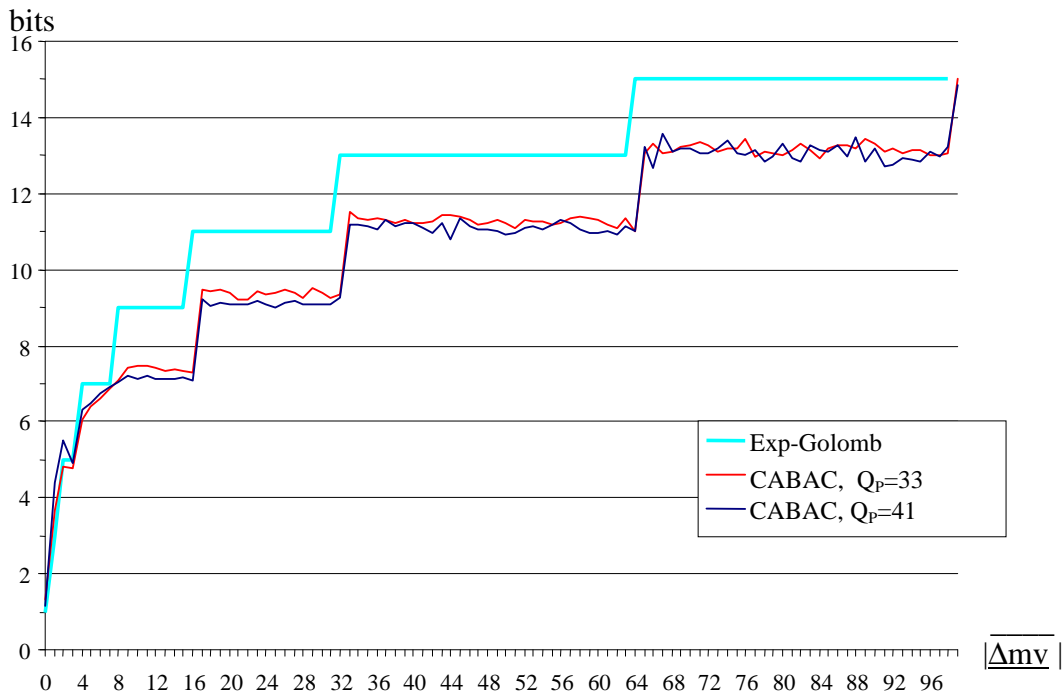


Fig. 4.31. Number of bits for residual motion vector component in P-frames. Comparison of Exp-Golomb codes against CABAC entropy coding in *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of 1/4-sample.

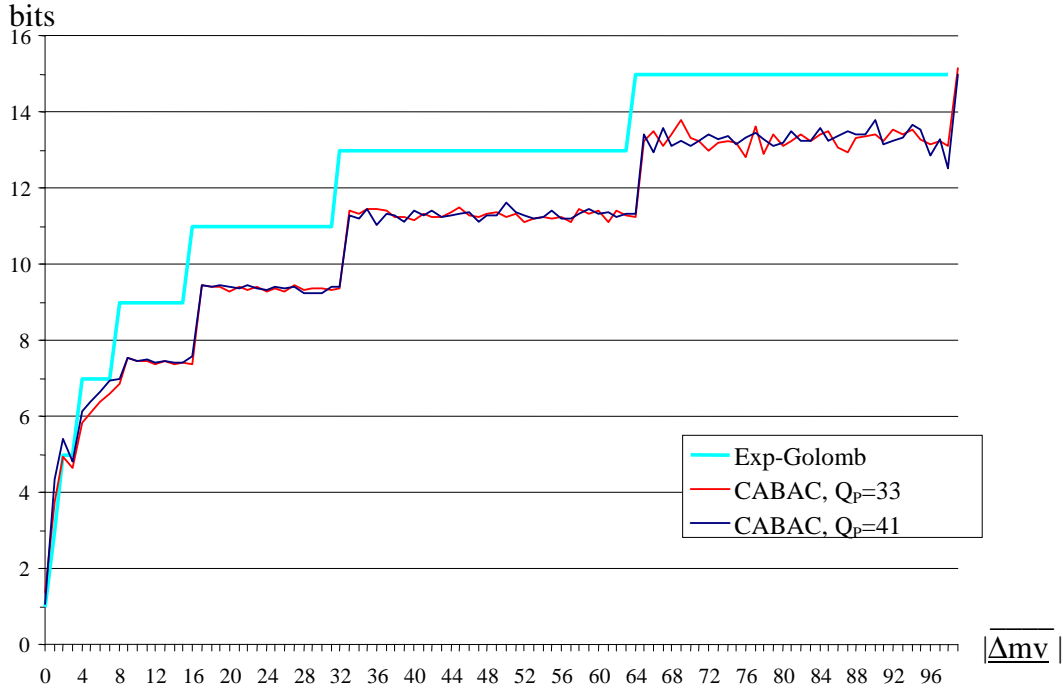


Fig. 4.32. Number of bits for residual motion vector component in B-frames. Comparison of Exp-Golomb codes against CABAC entropy coding in *Football* sequence (352×288, IBPBP), AVC/H.264 video codec. The residuals are given in the units of ¼-sample.

In Tab 4.10 the average differences between the length of the codeword using Exp-Golomb coding and the length of the codeword using CABAC for corresponding values of motion vector residuals are shown. The differences are calculated according to the following equation:

$$\Delta_{\text{CABAC}} = \frac{1}{2R} \sum_{i=-R}^R (l_{\text{CABAC},i} - l_{\text{Exp-Golomb},i}), \quad (4.7)$$

where:

- R – the range of considered motion vector residuals,
- Δ_{CABAC} – the average difference between lengths of codewords,
- $l_{\text{CABAC},i}$ – the length of codeword using CABAC entropy coding, for residual value of motion vector equal i . The length is averaged for 5 different bitrates,
- $l_{\text{Exp-Golomb},i}$ – the length of Exp-Golomb codeword for residual value of motion vector equal i .

Positive values of Δ_{CABAC} mean shorter length of codeword for Exp-Golomb codes, while negative values mean shorter length of codeword for CABAC. Values of motion vector residuals were limited to the range $\langle -99, 99 \rangle$ in the units of $\frac{1}{4}$ -sample ($R=99$).

Tab. 4.10. Average difference between length of the codeword using CABAC and length of the codeword using Exp-Golomb coding for corresponding values of motion vector residuals.

Video sequence	Δ_{CABAC}
<i>Bus</i>	-1.166
<i>Football</i>	-1.689
<i>Foreman</i>	-0.850
<i>Mobile</i>	0.507

4.4.4. Comparison of Exp-Golomb coding against CABAC coding of motion vectors - conclusions

The efficiency of coding of motion vectors using CABAC in AVC/H.264 has been tested. In video sequences *Foreman*, *Football* and *Bus* Context-Adaptive Arithmetic Coding outperforms Exp-Golomb coding. However, in *Mobile* video sequence, Exp-Golomb coding produces shorter codewords on average for motion vector residuals.

The biggest gain of CABAC has been achieved in *Football* video sequence – the average length of codewords for motion vector residuals is almost two bit shorter on average when CABAC was used. On the other hand, in *Mobile* sequence, the Exp-Golomb codes outperform CABAC – the average length of codewords for motion vector residuals is about 0.5 bit longer when using CABAC.

Usually CABAC coding produces shorter bit sequence than Exp-Golomb codes for larger values of motion vector component residuals (greater than 1.75 in full-pel units in *Bus* sequence, greater than 0.5 in full-pel units in *Football* and *Foreman* sequence). The maximum gain from using CABAC is achieved in the video sequences with relatively fast motion (*Football* and *Bus*). However, *Mobile* sequence, where the motion is rather

smooth and slow, the Exp-Golomb coding of motion vectors seems to be more efficient than CABAC.

There is no noticeable difference in the efficiency of CABAC in P- and B-frames. In all the test sequences, the average length of codeword in P- and B-frames is quite similar as depicted in Fig. 4.31 and Fig. 4.32.

The experimental results prove that in video sequences with fast motion and rough motion field like *Football*, it is profitable to use adaptive arithmetic coding. In the sequence with very slow motion and smooth motion field, Exp-Golomb coding gives shorter length of the codeword on average. On the other hand, in such sequences, motion vector data is the minor part of the overall bitrate [Lan03a] and transform coefficients form the majority of the overall bitstream. Additionally, median prediction described in Section 4.2, performs more efficiently in the sequences with slow motion, therefore the residual values of motion vectors are lower.

Exp-Golomb codes are matched with the motion vector residuals with exponential distribution. This exponential distribution better approximates the distribution of motion vector residuals in the video sequence with smooth motion vector field [Lan03, Lan03a]. Therefore, a possible reason for better performance of CABAC in the sequences with rough motion field is poor matching of standard Exp-Golomb codewords with the distribution of motion vector residuals.

On the other hand, CABAC has the ability to adapt to the local distribution of motion vector field. It allows for more efficient encoding of rough motion vector residuals. However, in the case when there is few motion data in a bitstream (video sequences with smooth motion), CABAC engine better matches transform coefficients distribution than motion vector residuals distribution, thus its efficiency of motion vectors encoding decreases.

In three on four tested video sequences, CABAC has definitely outperformed Exp-Golomb codes in coding of motion vectors. Although CABAC coding is more complex and requires additional computational power, it is undoubtedly a very efficient tool for entropy coding of motion vector residuals.

4.5. Summary

In this chapter the techniques of motion vectors coding used in advanced video codecs have been described with a special emphasis on the techniques utilized in AVC/H.264 video coding algorithm.

The methods of motion vectors prediction have been described in Section 4.2. The efficiency of motion vector prediction in non-scalable codec for various resolutions and various content of video sequence has been experimentally verified. Median prediction scheme is very efficient and produces very low prediction residuals (30%-75% of residual motion vector components encoded in a bitstream have value of 0). The efficiency of median prediction is extremely good, especially in sequences with slow and smooth motion.

However, motion vectors prediction using vector median, which was proposed by the author in Section 4.3, has not improved the efficiency of motion vectors coding.

The methods of entropy coding used in AVC/H.264 have been described in Section 4.4. The experimental comparison of coding performance using Exp-Golomb codes against context-adaptive arithmetic coding (CABAC) has been presented. CABAC proved to be definitely more efficient in a sequence with fast and rough motion (the codeword for single motion vector component is almost 1.7 bit shorter on average when CABAC was used in *Football* video sequence). In sequences with slow motion, Exp-Golomb coding of motion vectors seems to be a little more efficient than CABAC (CABAC codeword for single motion vector component is about 0.5 bit longer on average in *Mobile* video sequence).

The experiments presented in this chapter prove that the overall performance of motion vectors coding in advanced video codec is very efficient and there is just small room for further improvements of the existing techniques.

However, in the following chapter, motion vector fields in multiresolution representation of video sequence are considered. An analysis of these multiresolution motion vector fields should answer the question about possible similarities of motion vectors estimated for various resolutions of the video sequence. These correlations could be used in order to improve motion vectors coding in scalable video codec.

Chapter 5.

Multiresolution motion fields

5.1. A problem of multiresolution motion representation

In order to produce scalable bitstream that represents video sequence with many spatial resolutions, encoder performs motion estimation in each layer for the given spatial resolution of the input video sequence. Obtained motion fields may differ at each stage of spatial decomposition because of block-based motion estimation in each layer and the given rate-distortion criterion of macroblocks encoding.

Scalable hybrid video encoder produces motion vector fields for each spatial resolution of a sequence, as depicted in Fig. 5.1. Motion vectors for different resolutions have to be encoded in the bitstream.

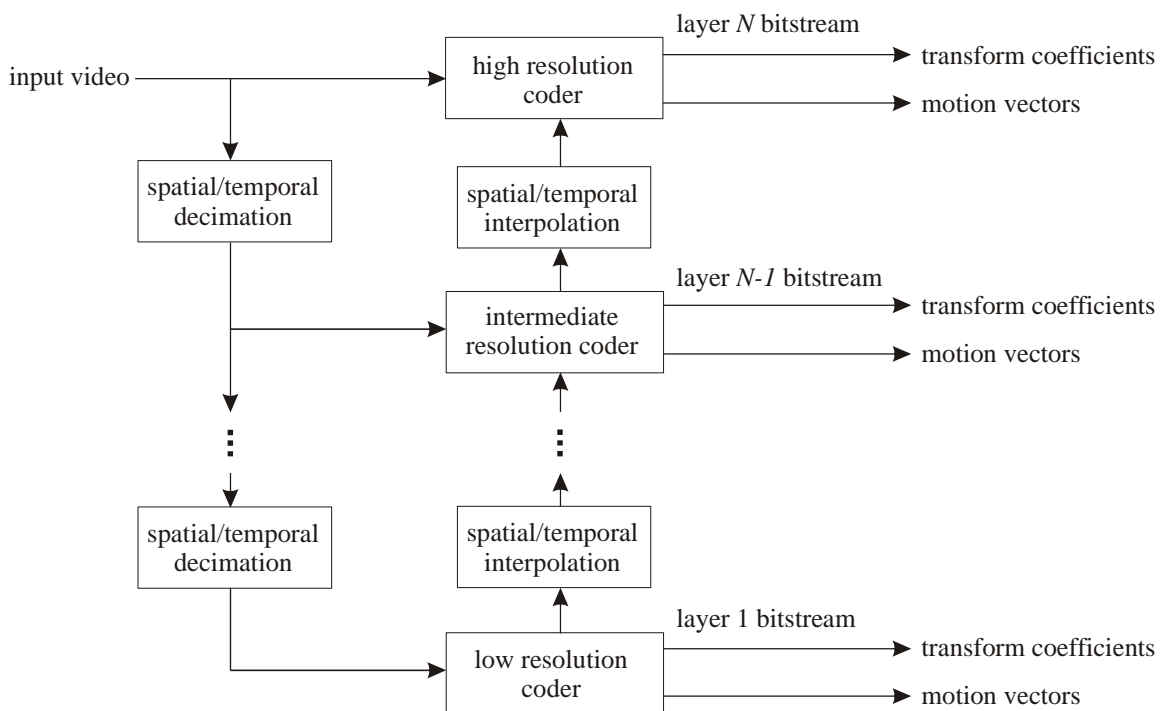


Fig. 5.1. Multiresolution video coding using pyramid of video coders.

The problem of multiresolution motion vector representation includes motion estimation and motion vectors encoding. Motion vectors calculated in each layer of scalable video coder may be estimated and represented as follows:

- independently estimated and independently represented,
- independently estimated and jointly represented,
- jointly estimated and independently represented,

- jointly estimated and jointly represented.

On the one hand, the best efficiency assures independent motion estimation for each resolution of the video sequence. However, existing correlations between motion vectors estimated for various resolutions, encourage to involve some techniques of joint encoding of motion vectors. Exploiting a motion vector from low-resolution layer during coding of motion vectors from high-resolution layer can possibly improve the efficiency of overall video compression.

The problem of estimation and representation of motion vectors regards hybrid DCT-based video coders as well as wavelet-based video coders. In both techniques of video coding, motion-compensated prediction requires motion field to be estimated and represented in a quite similar manner.

5.2. Multiresolution motion estimation in scalable video coding

When simulcast coding is used, motion vectors for each resolution of the video sequence are estimated and encoded independently [Gu99]. On the other hand, the technique of joint motion estimation for many layers of scalable video codec has been proposed [Con97]. Moreover, motion vectors estimated simultaneously for various spatial resolutions of video sequence can be encoded jointly in the bitstream [Lan03, Bar04a].

In pyramid coding scheme (Fig. 5.1) with joint motion estimation, motion can be estimated using coarse-to-fine or fine-to-coarse strategy. In coarse-to-fine approach, motion is first estimated at the coarsest level of resolution. The result of coarse-level estimation is then used as the initial estimate for the motion at a higher level. This scheme repeats until motion field for the highest resolution is obtained [Kar04]. Refinements of motion vectors are coded and transmitted for each level of the spatial resolution of a video sequence [Zaf93, Bar04a]. In coarse-to-fine approach good prediction is obtained at the coarsest resolution, but suboptimal motion estimation is performed for finer resolutions.

In fine-to-coarse strategy, in the first step motion is estimated for the highest spatial resolution. Obtained motion vectors are then scaled to coarser resolutions [Che01]. Therefore, accurate motion field is obtained for the high resolution video, while suboptimal motion vectors are estimated for lower resolutions.

Another technique of motion estimation in multiresolution video coding is independent estimation of motion vectors at each spatial resolution using rate-distortions criteria [Nav95, Con97, Bła04a]. Separate estimation of the motion vectors for each spatial resolution does not guarantee inter-resolution correlations. However, implicit correlations between multiresolution motion vector fields still exist and can be exploited in order to improve coding efficiency [Lan06].

In 2004, MPEG organization carried out a subjective comparison of scalable video codecs, including wavelet based codecs and scalable hybrid codecs. The best coding efficiency proved video encoders with rate-distortion optimization and independent motion estimation at each spatial resolution [Bar04, Sch04, Bła04b, Wie04]. Very good coding efficiency achieved a codec described in Section 2.4.2, with independent motion estimation and joint motion representation in enhancement layers [Bła04b].

5.3. Correlation of multiresolution motion vectors

In order to represent video sequence with various spatial resolutions using video coding with motion-compensated prediction, motion shall be estimated on each stage of spatial decimation. In Section 5.1 some observations have been made regarding different approaches to multiresolution motion estimation. However, in this chapter, we focus on independently estimated fields of motion vectors for different spatial resolutions of the same video sequence.

The goal of the experiments is to measure implicit similarities and correlation between motion vectors estimated for different video resolutions.

In order to produce multiresolution motion vector fields, a scalable video codec has been used [Bła04a, Bła04b]. The architecture of the encoder is depicted in Fig. 5.2.

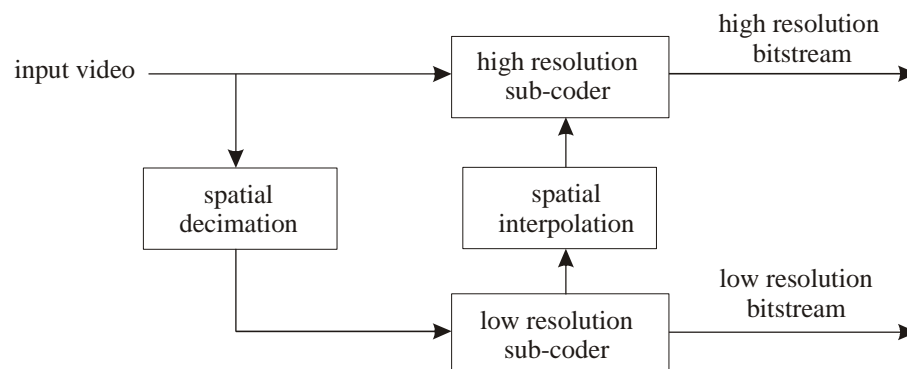


Fig. 5.2. Two-layer scalable video encoder, which has been used in experiments.

The encoder consists of two sub-coders with two loops of motion-compensated prediction. Motion is independently estimated in each sub-coder. The encoder produces a bitstream which represents video sequence with two different spatial resolutions. Temporal scalability in each layer is achieved by dropping bi-directionally coded, non-referenced frames.

As stated before, in the following experiments, motion vectors are independently estimated for each spatial resolution. In order to measure the correlation between estimated motion vectors from low resolution video sequence and high resolution video sequence, the differential motion vector field has been calculated as follows:

$$\Delta mv_{HL}(x, y, n) = mv_H(x, y, n) - m\tilde{v}_L(x, y, n), \text{ for } 0 \leq x < W, 0 \leq y < H \quad (5.1)$$

where:

- $\Delta mv_{HL}(x, y, n)$ – differential motion vector at location (x, y) in n -th video frame,
- $mv_H(x, y, n)$ – high-resolution motion vector at location (x, y) in n -th video frame,
- $m\tilde{v}_L(x, y, n)$ – interpolated low-resolution motion vector at location (x, y) in n -th video frame,
- W, H – width and height of the video sequence in 4×4 block units, horizontal and vertical dimensions respectively.

The average value of differential motion vector length is then calculated for each frame:

$$|\Delta mv_{HL}(n)| = \frac{1}{H \cdot W} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} \|\Delta mv_{HL}(i, j, n)\|_2, \wedge_{i,j} ref_H(i, j, n) = ref_L(i, j, n) \quad (5.2)$$

where:

- $|\Delta mv_{HL}(n)|$ – average value of differential motion vector length in n -th video frame,
- $ref_H(i, j, n)$ – reference frame used for motion-compensated prediction in high-resolution video sequence at location (i, j) in n -th video frame,
- $ref_L(i, j, n)$ – reference frame used for motion-compensated prediction in low-resolution video sequence at location (i, j) in n -th video frame.

In order to find the smoothness of differential motion field, standard deviation is estimated, separately for each video frame, using equation:

$$\sigma_{HL}(n) = \sqrt{\frac{1}{H \cdot W} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (\|\Delta mv_{HL}(i, j, n)\|_2 - |\Delta mv_{HL}(n)|)^2}, \quad (5.3)$$

$$\bigwedge_{i,j} ref_H(i, j, n) = ref_L(i, j, n)$$

Additionally, average length of differential motion vector and standard deviation of differential motion vector field is calculated for video sequence consisting of N frames:

$$\overline{\Delta mv_{HL}} = \frac{1}{N} \sum_{k=0}^{N-1} |\Delta mv_{HL}(k)| \quad (5.4)$$

$$\overline{\sigma}_{HL} = \frac{1}{N} \sum_{k=0}^{N-1} \sigma(k) \quad (5.5)$$

Another originally proposed parameter for characterizing the similarities of multiresolution motion vector fields is mutual matching parameter η_{HL} . It has been defined as follows:

$$\eta_{HL}(n) = \frac{1}{H \cdot W} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} 1, \quad \bigwedge_{i,j} ref_H(i, j, n) = ref_L(i, j, n) \quad (5.6)$$

Parameter η_{HL} is in range $\langle 0;1 \rangle$; it describes the percent of mutual matching of the motion vectors from high resolution video by motion vectors from low resolution video. Motion vectors match mutually, when there exist co-located motion vectors in low-resolution video and high-resolution video and when they both use the same reference frame for motion-compensated prediction. The average mutual matching is calculated for the entire video sequence consisting of N frames:

$$\overline{\eta}_{HL} = \frac{1}{N} \sum_{k=0}^{N-1} \eta_{HL}(k) \quad (5.7)$$

5.4. Multiresolution estimation of motion vectors – experimental results

Statistical properties of the motion vectors estimated for different spatial resolutions have been researched. It has been assumed that the measures of similarity between motion vector fields are the following parameters:

- average length of the difference between corresponding motion vectors for low and high spatial resolutions Δmv_{HL} ,
- standard deviation of the differential motion vector length σ_{HL} ,
- originally proposed mutual matching parameter η_{HL} .

The average length of the differential motion vector and mutual matching of motion fields are the direct measures of multiresolution correlations. Standard deviation of the differential motion vector field is the measure of motion field smoothness.

The experiments have been performed for sequences *Bus*, *Foreman*, *Football*, *Mobile*, *City* and *Crew* using SVC reference software version 4.0 [ISO06b]. Motion was independently estimated for each resolution and all inter-layer prediction modes were disabled. The following parameters have been set in the configuration file of the encoder:

- period between I frames: 96,
- group of pictures: I-B-P-B-P,
- number of reference frames: 3,
- motion vector search range: 96 samples,
- entropy coding: CABAC,
- range of motion estimation +/- 64 samples (full-pel units),
- range of quantization parameter Q_P : 25-41.

Correlation of the motion vectors have been calculated for the set of two different spatial resolutions: QCIF-CIF (*Bus*, *Foreman*, *Football* and *Mobile*) and CIF-4CIF (*City* and *Crew*).

5.4.1. Length of differential motion vectors in P-frames

In this section, results of estimation of differential motion vectors in P-frames are presented. In Fig. 5.3-5.6 the average length of differential motion vectors is presented for resolutions QCIF-CIF in sequences *Bus*, *Football*, *Foreman* and *Mobile*. In Fig. 5.7

and 5.8 the average length of differential motion vectors is presented for resolutions CIF-4CIF in sequences *City* and *Crew*. The values of differential motion vector length are given in $\frac{1}{4}$ -pel units.

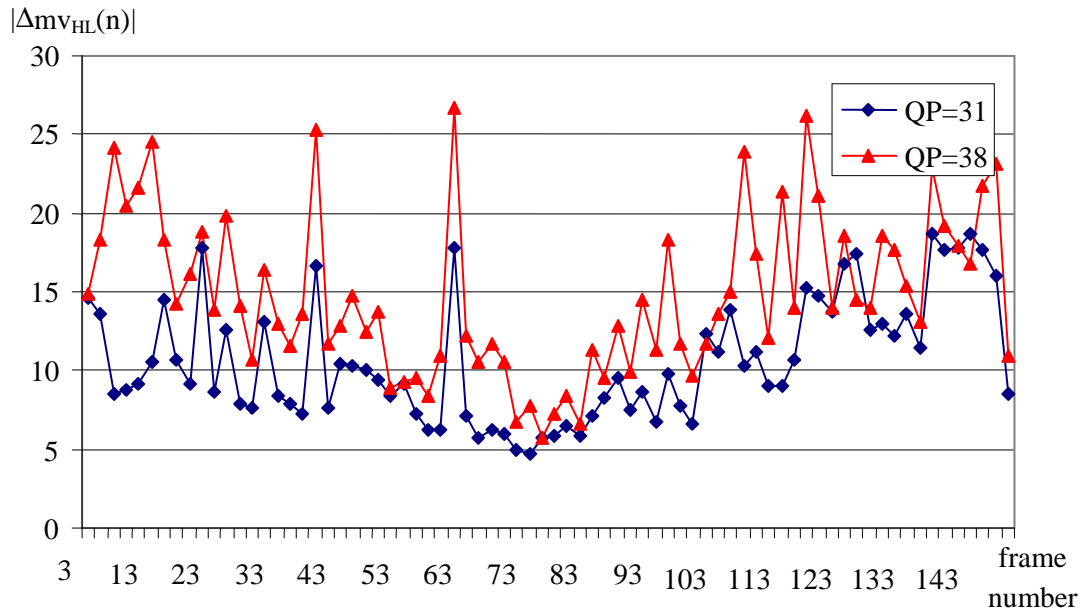


Fig. 5.3. Average length of differential motion vector in P-frames in *Bus* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

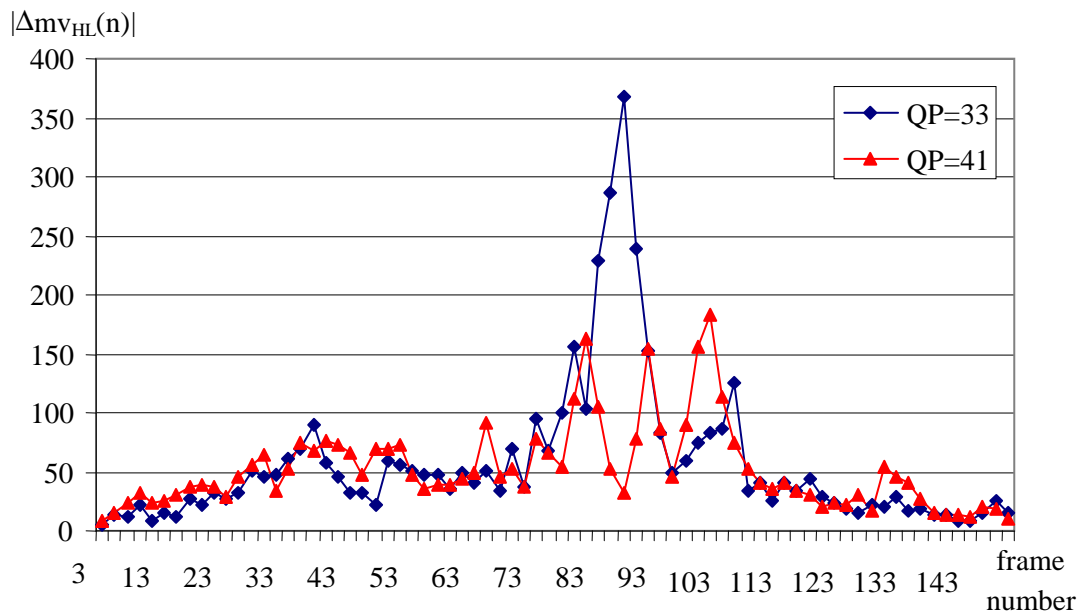


Fig. 5.4. Average length of differential motion vector in P-frames in *Football* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

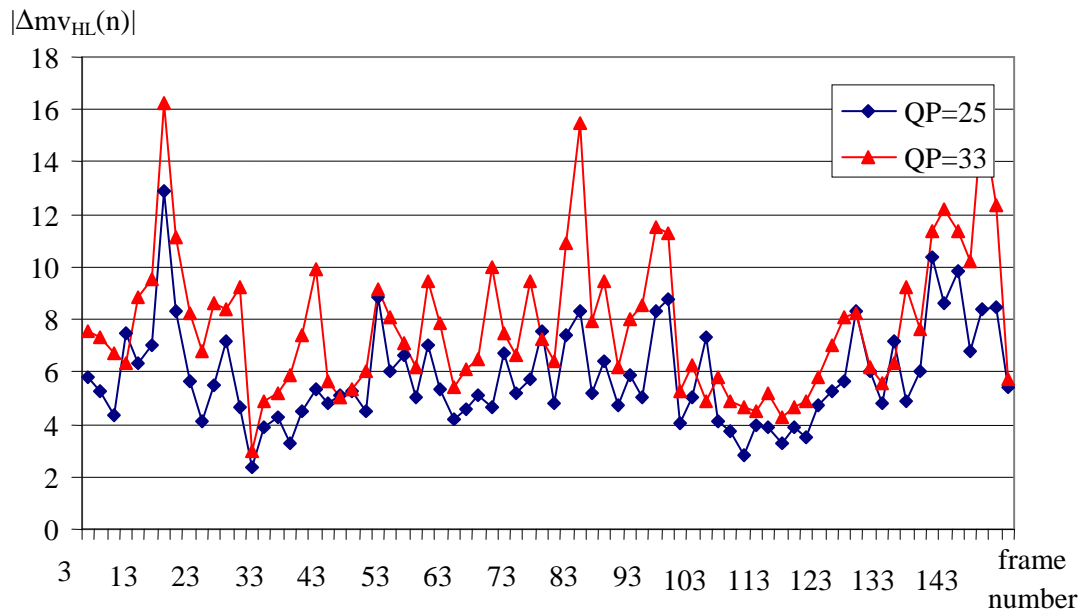


Fig. 5.5. Average length of differential motion vector in P-frames in *Foreman* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

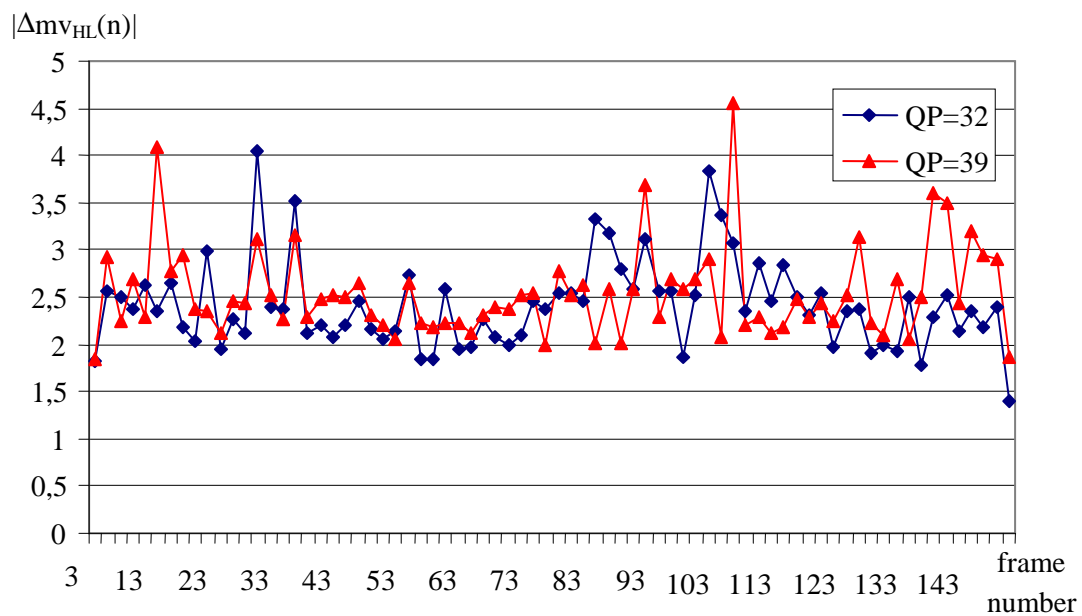


Fig. 5.6. Average length of differential motion vector in P-frames in *Mobile* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

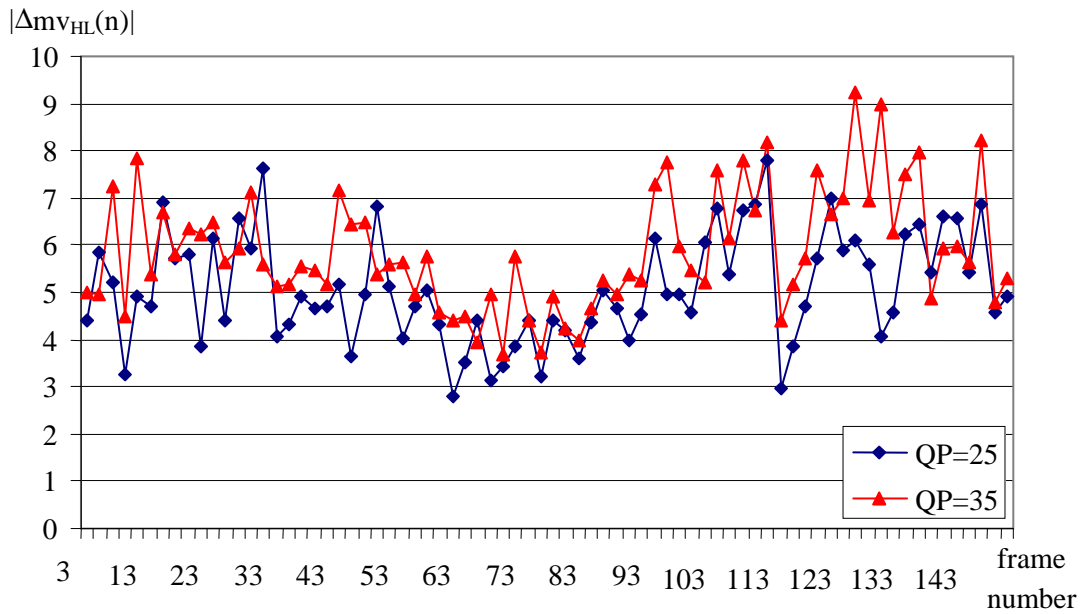


Fig. 5.7. Average length of differential motion vector in P-frames in *City* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

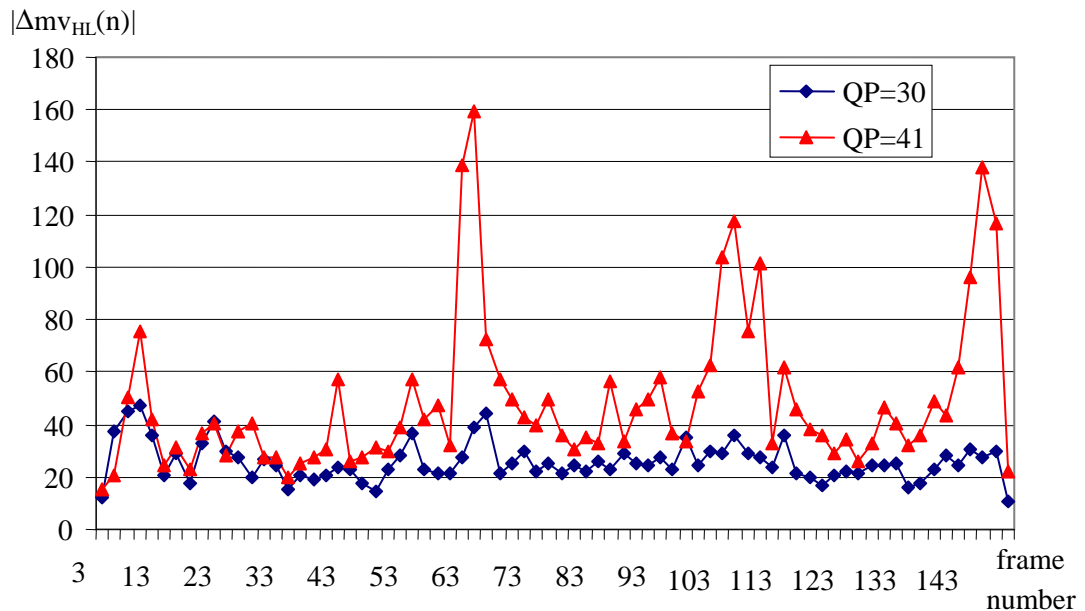


Fig. 5.8. Average length of differential motion vector in P-frames in *Crew* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

The length of differential motion vector $|\Delta mv_{HL}|$ varies for all tested sequences. It strongly depends on the contents of specific video frames, however, the value of this parameter is definitely higher in the sequences with very fast motion like *Bus* and *Football*. The peak of $|\Delta mv_{HL}|$ in *Football* sequence (frame number 87, compare visualization of motion field in Fig. 5.29) appears in the video frame that contains only a fast moving ball over the grassy background. In this case, block matching algorithm of motion estimation gives completely different results in each spatial resolution.

Moreover, in the video sequence that contains rapid global illumination changes (*Crew*), the average differential motion vector is also high, especially in the video frames with such global illumination changes (compare peaks in Fig. 5.8 and visualization of motion field in Fig. 5.37).

On the other hand, in sequences with rather slow motion (*Mobile*, *City*), the length of differential motion vector is low and its value stays almost constant for each video frame.

In most cases, the average length of differential motion vector has been higher for video sequences encoded with lower quantization parameter Q_p . Therefore, the conclusion is that multiresolution motion vector fields are more correlated for the video sequences encoded with high bitrate and high quality.

5.4.2. Length of differential motion vectors in B-frames

In this section, the results of estimation of differential motion vectors in B-frames are presented. In Fig. 5.9-5.12 the average length of differential motion vectors are presented for resolutions QCIF-CIF in sequences *Bus*, *Football*, *Foreman* and *Mobile*. In Fig. 5.13 and 5.14 the average length of differential motion vectors is presented for resolutions CIF-4CIF in sequences *City* and *Crew*. The values of differential motion vector length are given in $\frac{1}{4}$ -pel units.

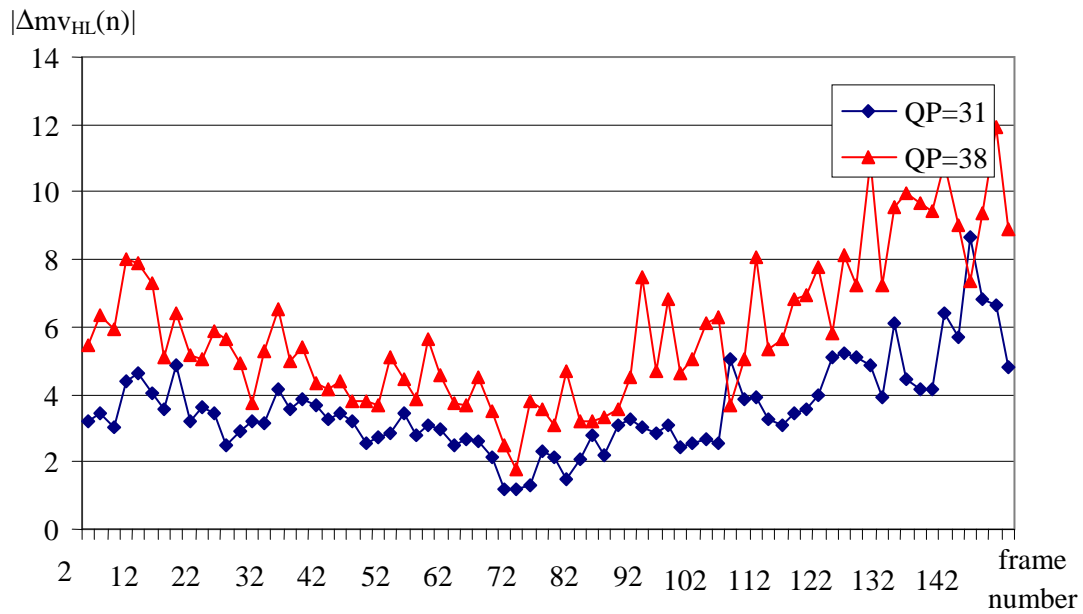


Fig. 5.9. Average length of differential motion vector in B-frames in *Bus* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

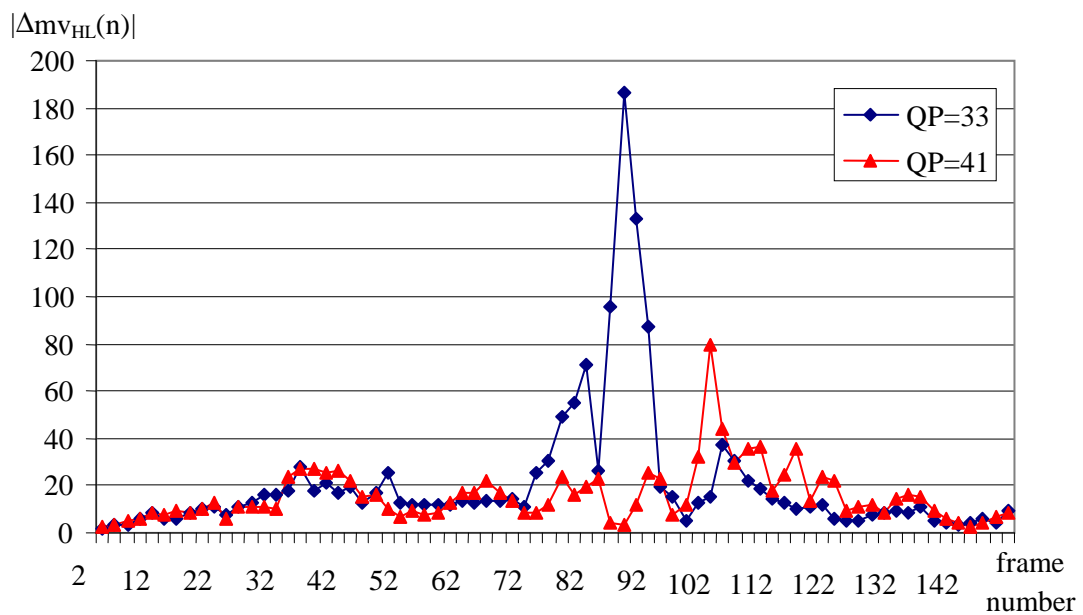


Fig. 5.10. Average length of differential motion vector in B-frames in *Football* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

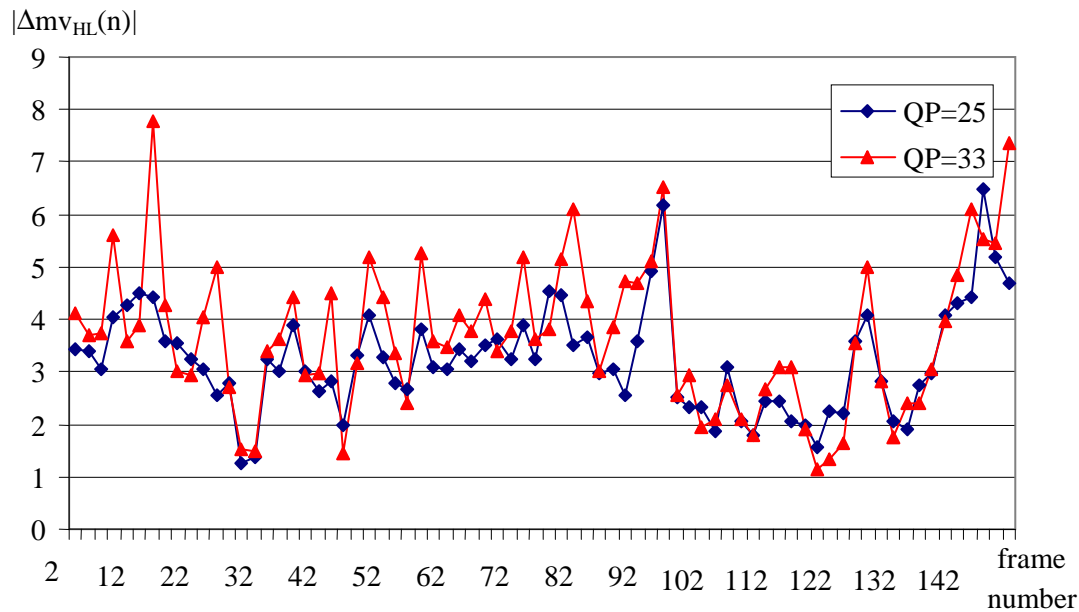


Fig. 5.11. Average length of differential motion vector in B-frames in *Foreman* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

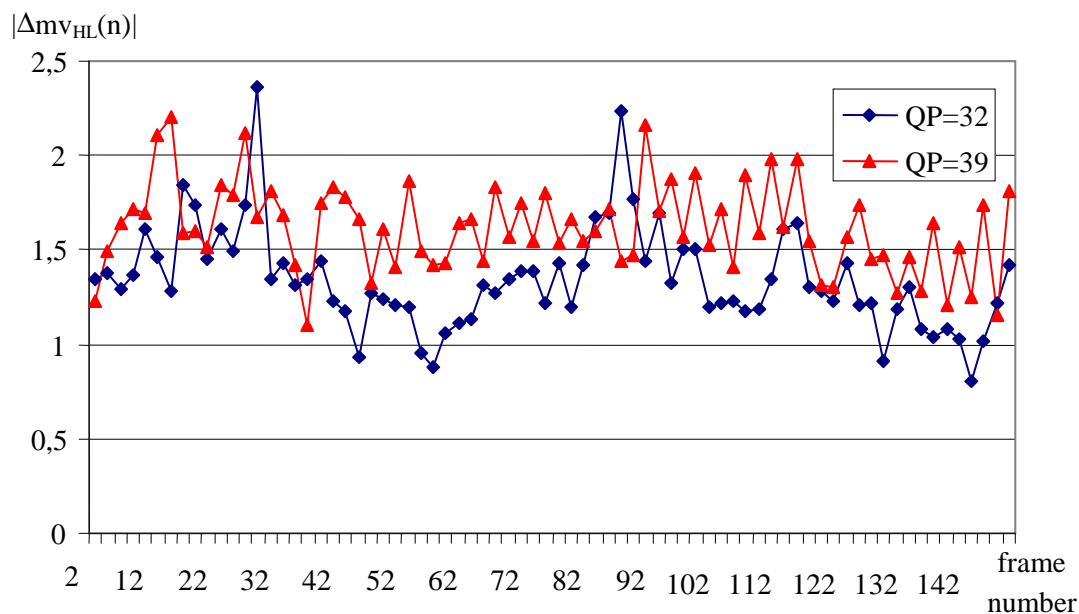


Fig. 5.12. Average length of differential motion vector in B-frames in *Mobile* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288 . Length of motion vector is given in the units of $\frac{1}{4}$ -sample.

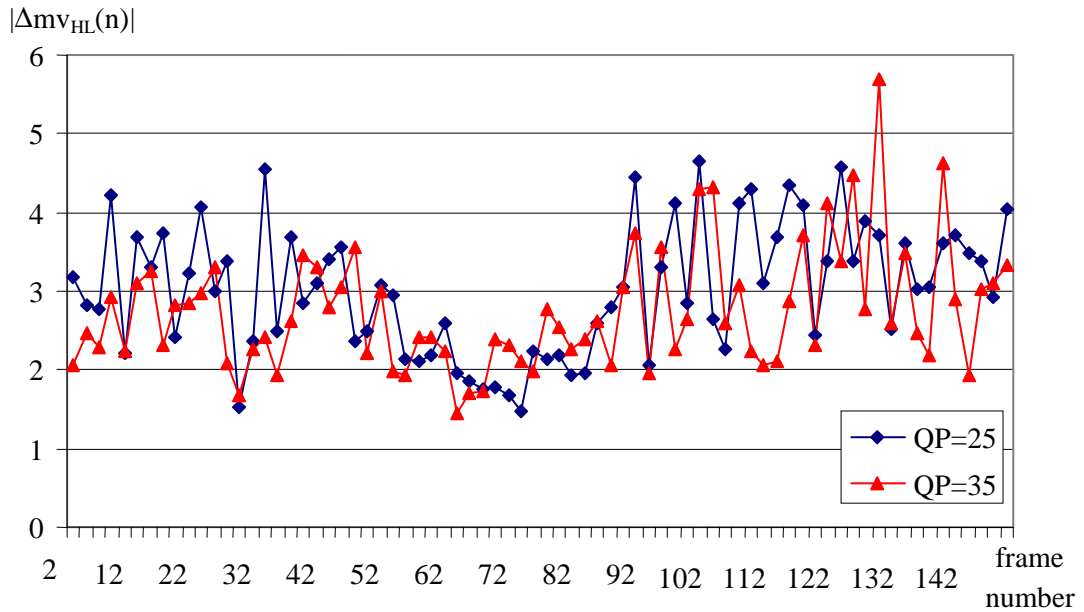


Fig. 5.13. Average length of differential motion vector in B-frames in *City* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576 . Length of motion vector is given in the units of $1/4$ -sample.

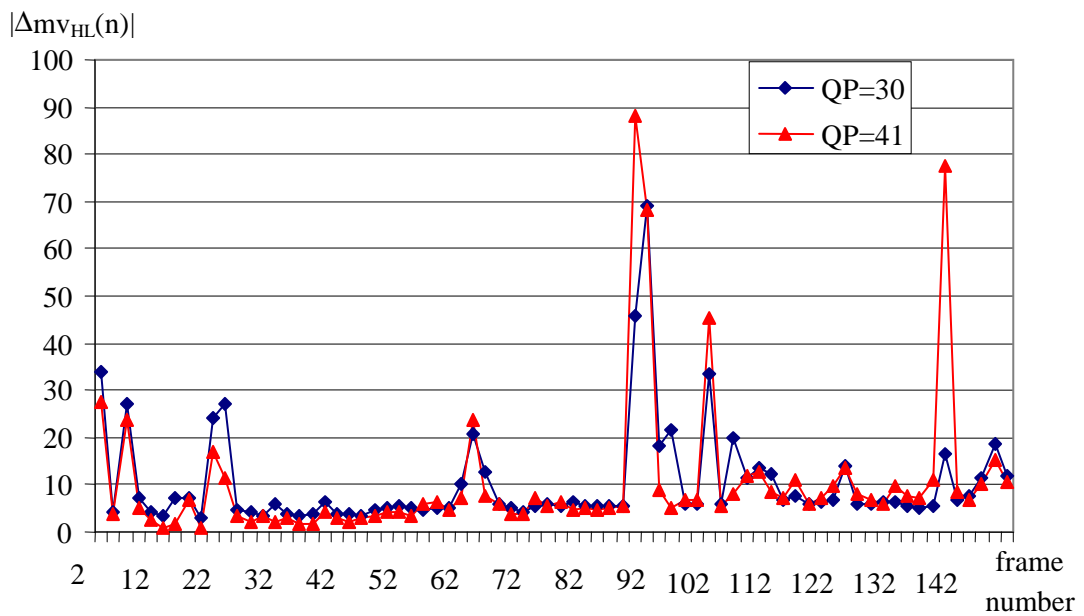


Fig. 5.14. Average length of differential motion vector in B-frames in *Crew* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576 . Length of motion vector is given in the units of $1/4$ -sample.

The actual values of length of differential motion vector $|\Delta mv_{HL}|$ are about twice as high in P-frames as in B-frames. However, the general trend in variations of $|\Delta mv_{HL}|$ remains the same for B-frames. Lower values of $|\Delta mv_{HL}|$ mean that multiresolution motion vector fields are even more correlated in B-frames than in P-frames.

5.4.3. Standard deviation of the differential motion vector length in P-frames

In this section the values of standard deviation of differential motion vectors length in P-frames are presented. In Fig. 5.15-5.18 the values of standard deviation of differential motion vectors length are presented for resolutions QCIF-CIF in sequences *Bus*, *Football*, *Foreman* and *Mobile*. In Fig. 5.19 and 5.20 the values of standard deviation of differential motion vectors length are presented for resolutions CIF-4CIF in sequences *City* and *Crew*. The values are given in $\frac{1}{4}$ -pel units.

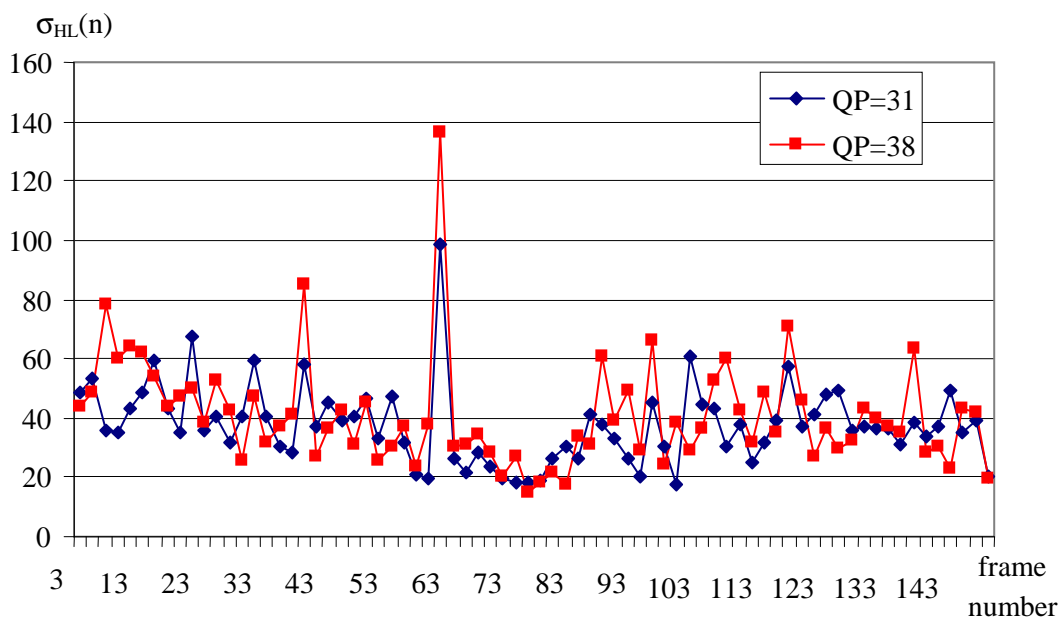


Fig. 5.15. Standard deviation of differential motion vector length in P-frames in *Bus* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

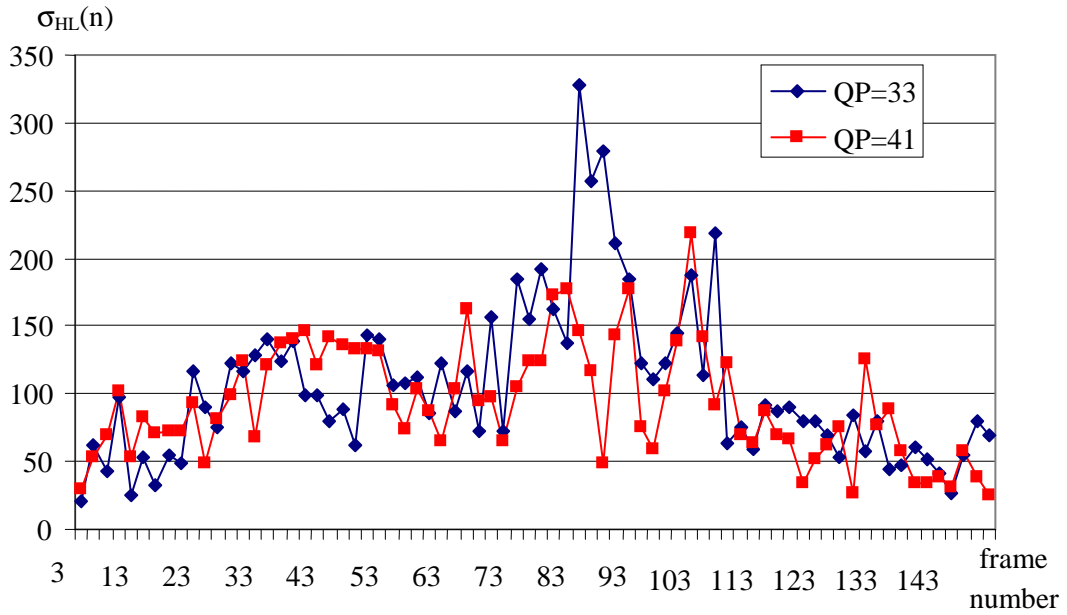


Fig. 5.16. Standard deviation of differential motion vector length in P-frames in *Football* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

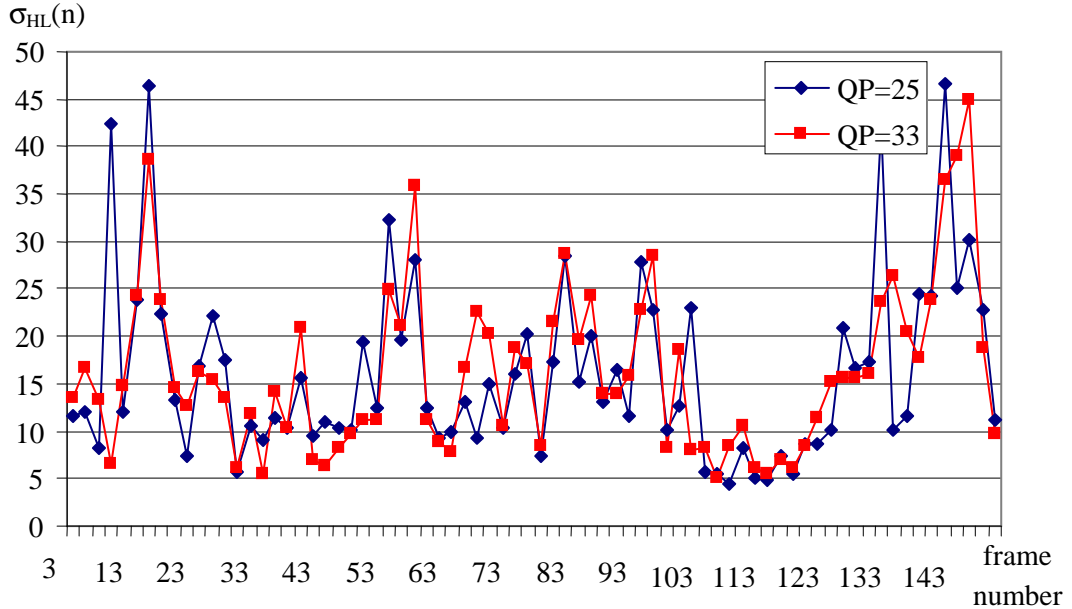


Fig. 5.17. Standard deviation of differential motion vector length in P-frames in *Foreman* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

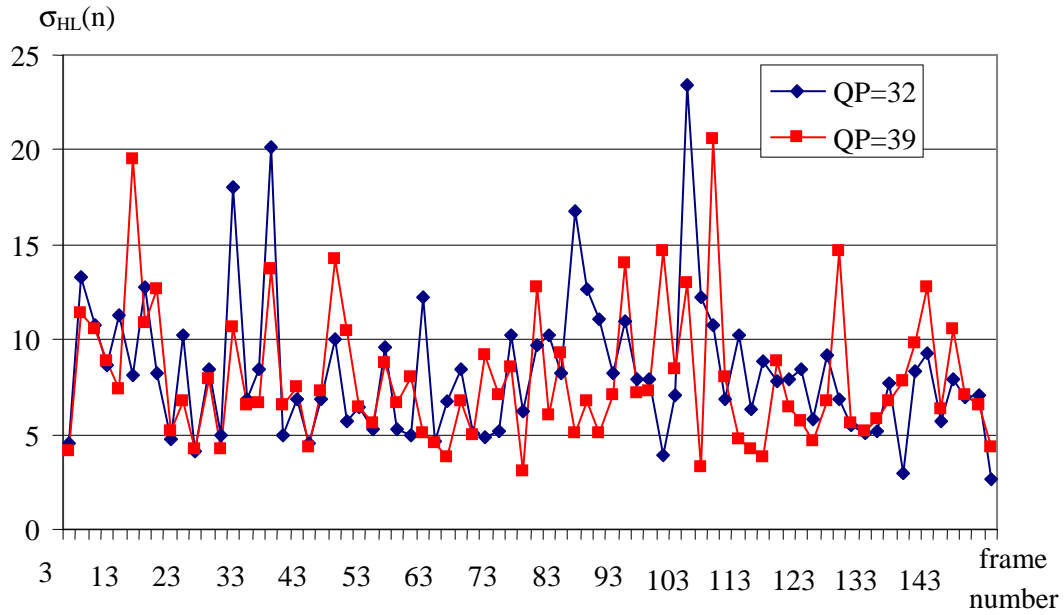


Fig. 5.18. Standard deviation of differential motion vector length in P-frames in *Mobile* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

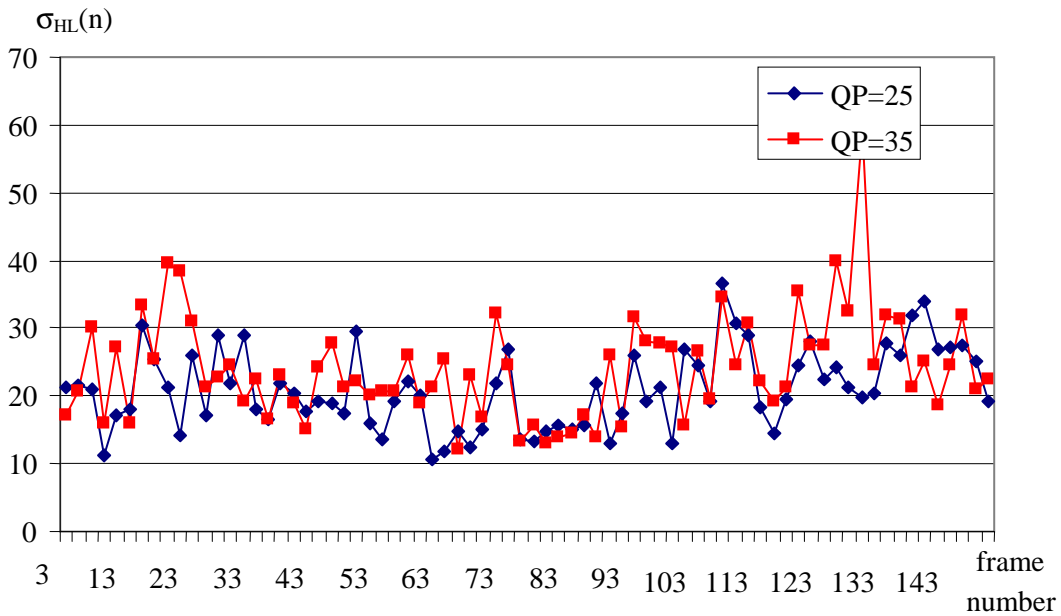


Fig. 5.19. Standard deviation of differential motion vector length in P-frames in *City* sequence for different quantization parameter Q_p . Values for resolutions 352×288 and 704×576 , given in the units of $\frac{1}{4}$ -sample.

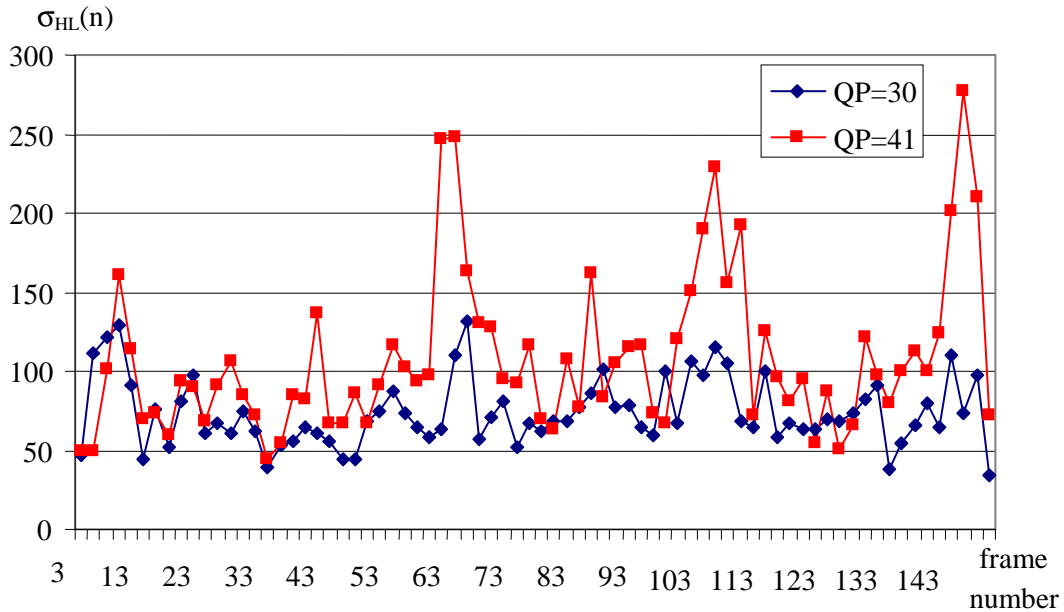


Fig. 5.20. Standard deviation of differential motion vector length in P-frames in *Crew* sequence for different quantization parameter Q_p . Values for resolutions 352×288 and 704×576 , given in the units of $\frac{1}{4}$ -sample.

The changes of the standard deviation of differential motion vector length reflects the smoothness of multiresolution differential motion vector field. Interestingly, changes of $\sigma_{HL}(n)$ value are correlated with changes of the length of differential motion vector, presented in Section 5.3.1.

The lowest values of the standard deviation have been observed in the *Mobile* and *City* sequences. Therefore, the smoothness of multiresolution differential motion vector field strongly depends on smoothness of the motion in the sequence.

The highest values of the standard deviation have been observed in the sequences *Football* and *Crew*. The peak value of $\sigma_{HL}(n)$ in P-frames is 328 in *Football* sequence ($Q_p=33$, frame 83) and 278 in *Crew* sequence ($Q_p=40$, frame 145).

5.4.4. Standard deviation of the differential motion vector length in B-frames

In this section the values of standard deviation of differential motion vectors length in B-frames are presented. In Fig. 5.21-5.24 the values of standard deviation of differential motion vectors length are presented for resolutions QCIF-CIF in sequences

Bus, *Football*, *Foreman* and *Mobile*. In Fig. 5.25 and 5.26 the values of standard deviation of differential motion vectors length are presented for resolutions CIF-4CIF in sequences *City* and *Crew*. The values are given in $\frac{1}{4}$ -pel units.

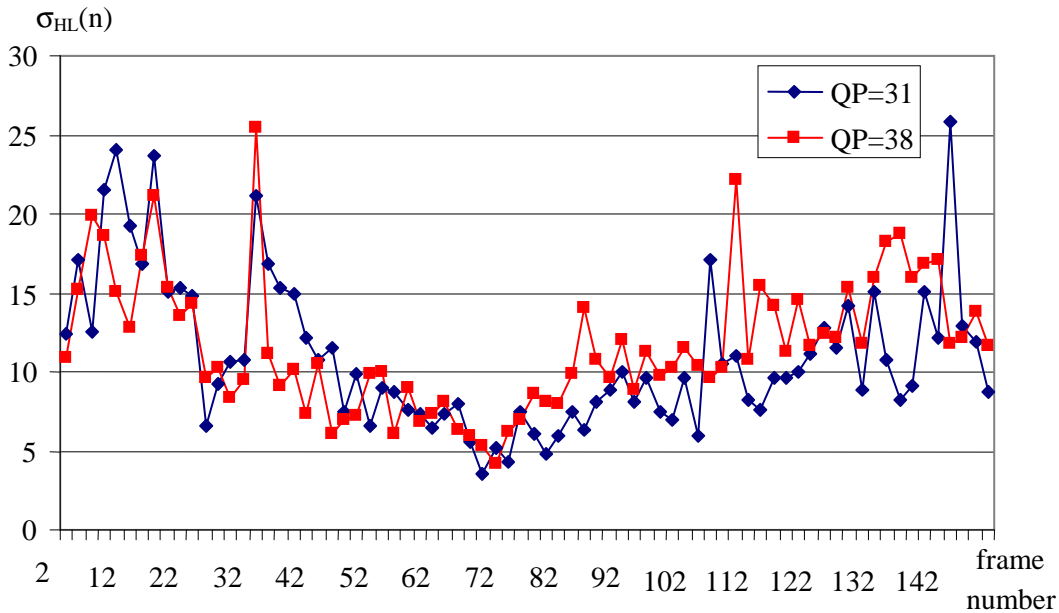


Fig. 5.21. Standard deviation of differential motion vector length in B-frames in *Bus* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

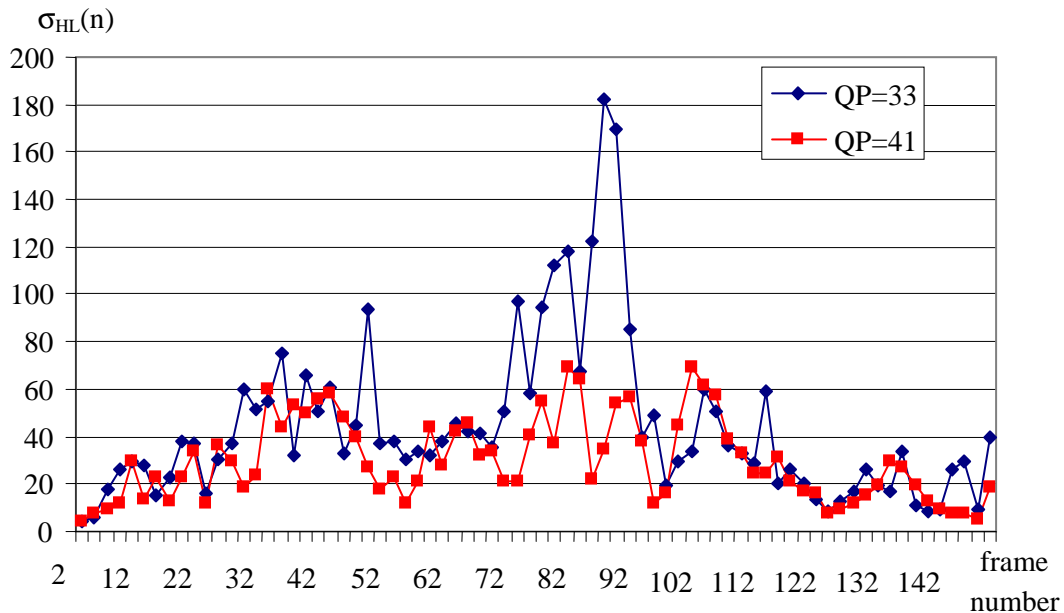


Fig. 5.22. Standard deviation of differential motion vector length in B-frames in *Football* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $\frac{1}{4}$ -sample.

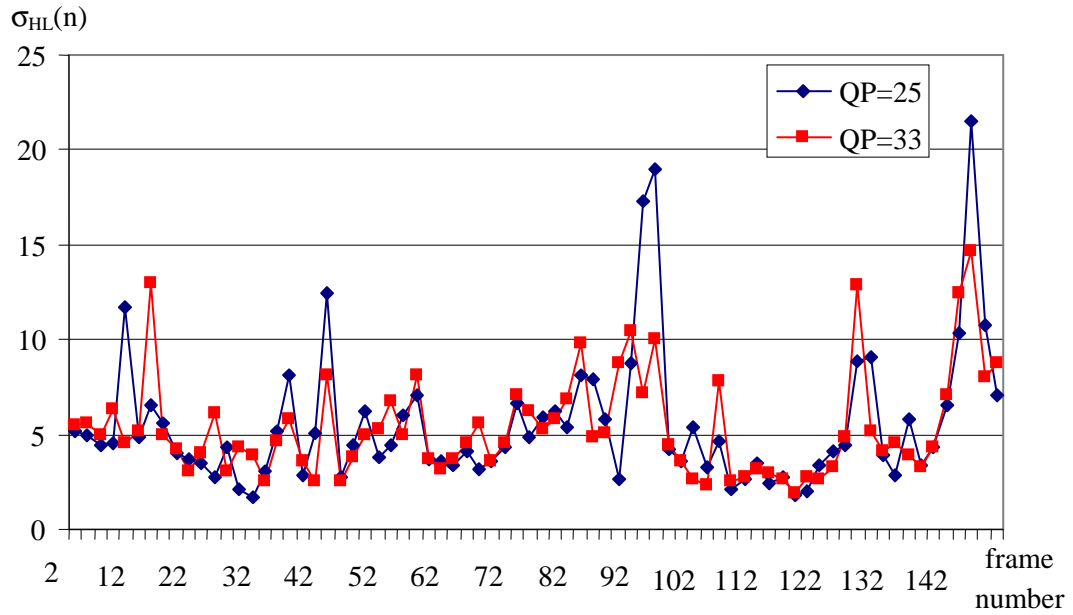


Fig. 5.23. Standard deviation of differential motion vector length in B-frames in *Foreman* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $1/4$ -sample.

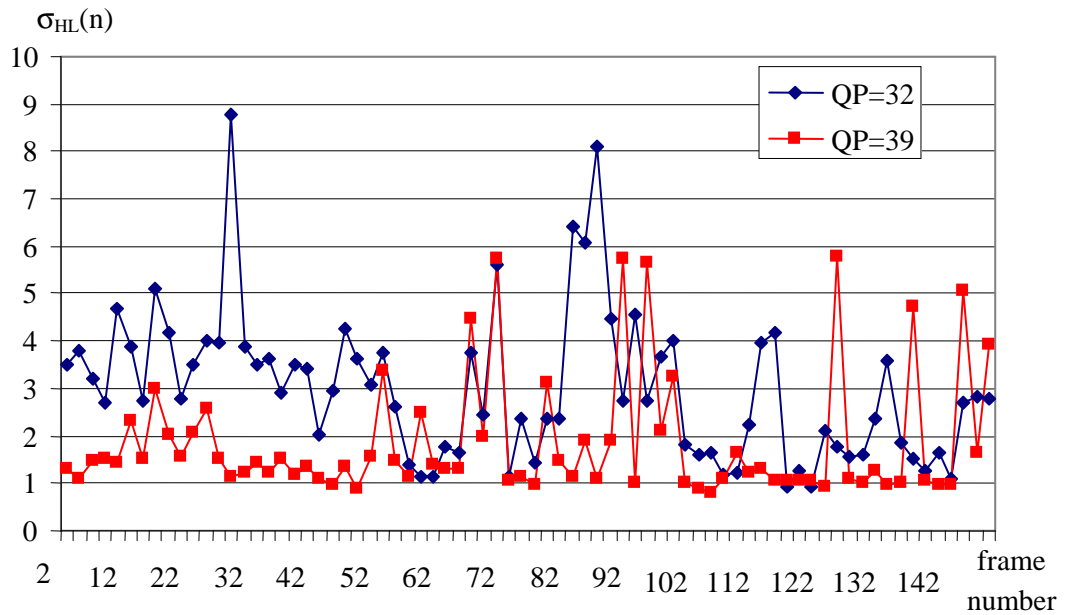


Fig. 5.24. Standard deviation of differential motion vector length in B-frames in *Mobile* sequence for different quantization parameter Q_p . Values for resolutions 176×144 and 352×288 , given in the units of $1/4$ -sample.

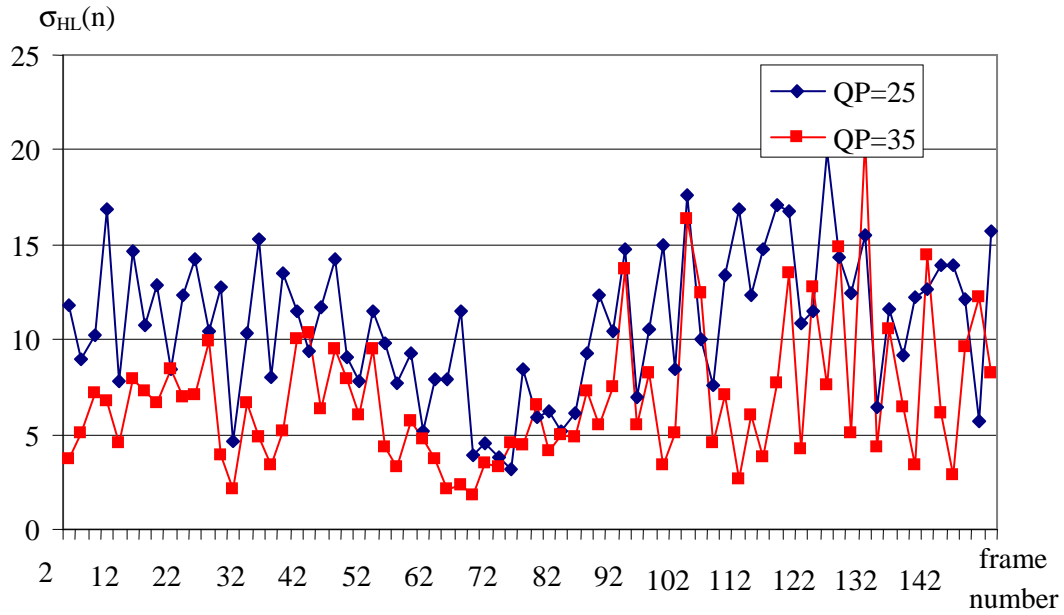


Fig. 5.25. Standard deviation of differential motion vector length in B-frames in *City* sequence for different quantization parameter Q_p . Values for resolutions 352×288 and 704×576 , given in the units of $\frac{1}{4}$ -sample.

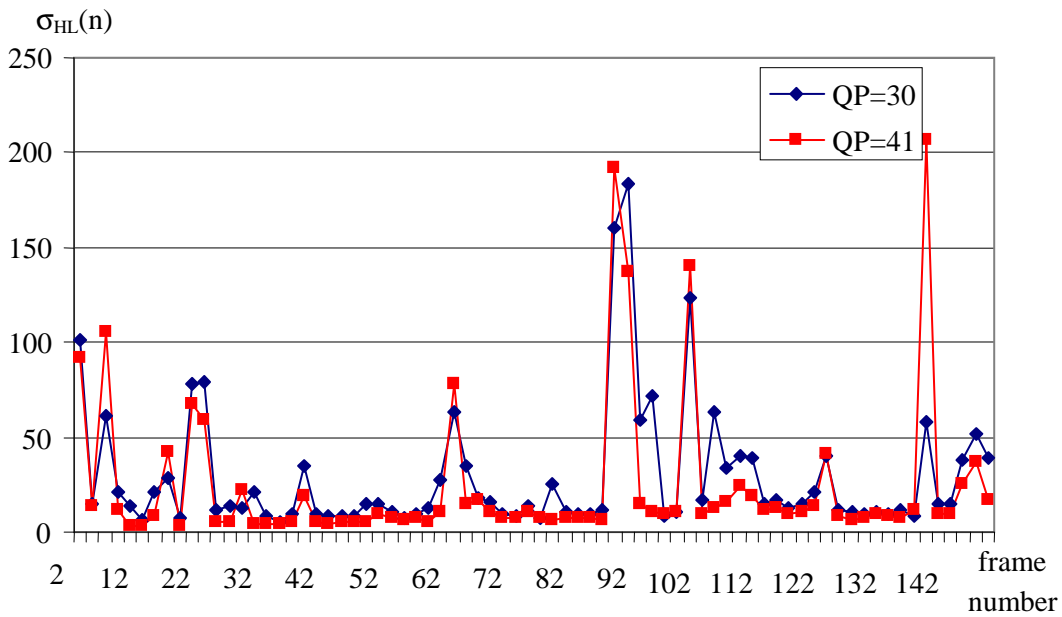


Fig. 5.26. Standard deviation of differential motion vector length in B-frames in *Ice* sequence for different quantization parameter Q_p . Values for resolutions 352×288 and 704×576 , given in the units of $\frac{1}{4}$ -sample.

The values of the standard deviation of differential motion vector length in B-frames are lower than the corresponding values obtained for P-frames, therefore, the multiresolution differential motion vector fields in B-frames are smoother.

Again, the highest values of the standard deviation have been observed in the sequences *Football* and *Crew*. The peak value of $\sigma_{HL}(n)$ in B-frames is 182 in *Football* sequence ($Q_P=33$, frame 86) and 206 in *Crew* sequence ($Q_P=40$, frame 138).

5.4.5. Average differential motion vector length and average standard deviation of the differential motion vector length

In Tab. 5.1-5.4 the average values of differential motion vectors length and standard deviation of differential motion vectors length are presented. The values have been averaged for the entire video sequence.

The experimental results for resolutions QCIF-CIF are shown in Tab 5.1-5.4. The experimental results for resolutions CIF-4CIF are shown in Tab 5.5 and Tab 5.6.

Tab. 5.1. Average values of differential motion vector length and standard deviation of differential motion vector length in *Bus* sequence for different quantization parameter Q_P . Motion vectors differentials for resolutions 176×144 and 352×288 . Values are given in the units of $\frac{1}{4}$ -sample.

Q_P	PSNR (dB)		P-frames		B-frames	
	176×144	352×288	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
31	33.48	33.82	10.600	37.577	3.554	11.002
33	31.93	32.39	10.862	35.928	3.994	12.122
35	30.50	31.06	12.104	37.848	4.272	11.386
37	29.02	29.64	13.931	38.659	5.177	11.438
38	28.43	29.09	15.271	42.652	5.827	11.677

Tab. 5.2. Average values of differential motion vector length and standard deviation of differential motion vector length in *Football* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288. Values are given in the units of 1/4-sample.

Q_p	PSNR (dB)		P-frames		B-frames	
	176×144	352×288	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
33	33.02	33.97	57.186	105.413	20.845	43.857
35	31.74	32.79	60.323	98.591	20.282	40.445
37	30.51	31.56	58.464	96.688	19.813	36.819
39	29.26	30.46	58.212	96.899	17.739	36.343
41	28.12	29.44	52.957	93.219	15.795	29.738

Tab. 5.3. Average values of differential motion vector length and standard deviation of differential motion vector length in *Foreman* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288. Values are given in the units of 1/4-sample.

Q_p	PSNR (dB)		P-frames		B-frames	
	176×144	352×288	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
25	39.84	39.78	5.833	16.092	3.230	5.539
27	38.43	38.59	5.983	14.073	3.436	5.400
29	37.10	37.44	6.567	15.363	3.651	5.908
31	35.65	36.19	7.017	15.833	3.746	5.694
33	34.28	34.99	7.750	16.133	3.682	5.373

Tab. 5.4. Average values of differential motion vector length and standard deviation of differential motion vector length in *Mobile* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 176×144 and 352×288. Values are given in the units of 1/4-sample.

Q_p	PSNR (dB)		P-frames		B-frames	
	176×144	352×288	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
32	34.08	35.45	2.412	8.264	1.343	2.988
33	33.53	34.98	2.433	8.380	1.431	3.050
35	32.55	34.21	2.488	8.387	1.463	2.439
37	31.24	33.08	2.486	7.947	1.516	2.072
39	30.40	32.38	2.538	7.923	1.617	1.855

Tab. 5.5. Average values of differential motion vector length and standard deviation of differential motion vector length in *City* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576. Values are given in the units of 1/4-sample.

Q_p	PSNR (dB)		P-frames		B-frames	
	352×288	704×576	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
25	38.45	38.19	5.071	20.986	3.021	10.791
28	36.19	36.48	5.147	21.276	3.038	10.475
30	34.69	35.30	5.276	21.205	2.928	9.801
33	32.50	33.47	5.640	22.925	2.797	8.165
35	31.20	32.31	5.888	24.085	2.745	6.817

Tab. 5.6. Average values of differential motion vector length and standard deviation of differential motion vector length in *Crew* sequence for different quantization parameter Q_p . Motion vectors differentials for resolutions 352×288 and 704×576. Values are given in the units of 1/4-sample.

Q_p	PSNR (dB)		P-frames		B-frames	
	352×288	704×576	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$	$\overline{\Delta mv_{HL}}$	$\overline{\sigma}_{HL}$
30	36.65	37.06	25.783	73.987	10.260	28.924
33	34.62	35.60	32.915	87.952	10.752	28.028
36	32.70	34.12	41.671	103.718	12.548	30.729
38	31.49	33.18	45.819	106.858	11.565	27.141
41	29.66	31.60	48.627	108.074	10.474	24.334

The comparison of the obtained average values of length of differential motion vector $|\Delta mv_{HL}|$ (Sections 5.3.1 and 5.3.2) with the values of the standard deviation of length of differential motion vector σ_{HL} (Sections 5.3.1 and 5.3.2) leads to conclusion, that when motion vectors estimated for a low-resolution video sequence are similar to the motion vectors estimated for a high-resolution video sequence, the parameters $|\Delta mv_{HL}|$ and σ_{HL} have low values. Moreover, usually, more similar motion vector fields are those estimated for video sequences containing slow and smooth motion.

5.4.6. Mutual matching of motion vector fields

In Tab. 5.7-5.12 mutual matching of motion vector fields is presented, which is characterized by average values of mutual matching parameter, defined in Section 5.2. The values have been averaged for entire video sequence. Experimental results for resolutions QCIF-CIF are showed in Tab 5.7-5.10. Experimental results for resolutions CIF-4CIF are showed in Tab 5.11 and Tab 5.12.

Tab. 5.7. Average mutual matching of motion vectors in *Bus* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 176×144 and 352×288, separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	176×144	352×288	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
31	33.48	33.82	0.956	0.609	0.786
33	31.93	32.39	0.959	0.561	0.776
35	30.50	31.06	0.962	0.522	0.776
37	29.02	29.64	0.965	0.513	0.740
38	28.43	29.09	0.966	0.509	0.725

Tab. 5.8. Average mutual matching of motion vectors in *Football* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 176×144 and 352×288, separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	176×144	352×288	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
33	33.48	33.82	0.723	0.551	0.703
35	31.93	32.39	0.735	0.563	0.693
37	30.50	31.06	0.751	0.564	0.717
39	29.02	29.64	0.771	0.593	0.716
41	28.43	29.09	0.793	0.604	0.721

Tab. 5.9. Average mutual matching of motion vectors in *Foreman* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 176×144 and 352×288 , separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	176×144	352×288	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
25	33.48	33.82	0.953	0.747	0.799
27	31.93	32.39	0.958	0.717	0.789
29	30.50	31.06	0.959	0.708	0.796
31	29.02	29.64	0.960	0.698	0.801
33	28.43	29.09	0.961	0.686	0.808

Tab. 5.10. Average mutual matching of motion vectors in *Mobile* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 176×144 and 352×288 , separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	176×144	352×288	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
32	33.48	33.82	0.998	0.888	0.876
33	31.93	32.39	0.997	0.870	0.863
35	30.50	31.06	0.997	0.836	0.823
37	29.02	29.64	0.998	0.798	0.805
39	28.43	29.09	0.998	0.753	0.802

Tab. 5.11. Average mutual matching of motion vectors in *City* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 352×288 and 704×576, separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	352×288	704×576	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
25	38.45	38.19	0.982	0.801	0.807
28	36.19	36.48	0.986	0.738	0.767
30	34.69	35.30	0.987	0.701	0.764
33	32.50	33.47	0.988	0.669	0.760
35	31.20	32.31	0.988	0.661	0.774

Tab. 5.12. Average mutual matching of motion vectors in *Crew* sequence for different quantization parameter Q_p . Parameter $\bar{\eta}_{HL}$ estimated for resolutions 352×288 and 704×576, separately for forward prediction ($\bar{\eta}_{HL,F}$) and backward prediction ($\bar{\eta}_{HL,B}$).

Q_p	PSNR (dB)		P-frames	B-frames	
	352×288	704×576	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,F}$	$\bar{\eta}_{HL,B}$
30	36.65	37.06	0.744	0.590	0.759
33	34.62	35.60	0.764	0.555	0.766
36	32.70	34.12	0.781	0.578	0.780
38	31.49	33.18	0.793	0.612	0.799
41	29.66	31.60	0.808	0.696	0.838

The parameter $\bar{\eta}_{HL}$ describes how many motion vectors in high-resolution layer have their equivalent co-located (matching) motion vectors in low-resolution layer. The higher value of mutual matching parameter $\bar{\eta}_{HL}$, the more number of 4×4 blocks in low-resolution layer have been encoded using the same type of motion-compensated prediction and using the same reference frame.

The highest values of mutual matching parameter $\bar{\eta}_{HL}$ have been obtained for the video sequences where the values of differential motion vector length $|\Delta mv_{HL}|$ as well as deviation of differential motion vector length σ_{HL} are very low: *Mobile* and *City*. In these sequences almost 100% of the motion vectors in high-resolution layer have matching motion vectors in low resolution-layer for P-frames ($\bar{\eta}_{HL} > 0.99$ for *Mobile* in P-frames, $\bar{\eta}_{HL} > 0.98$ for *City* in P-frames).

On the other hand, in *Football* and *Crew* video sequences, where the values of differential motion vector length $|\Delta mv_{HL}|$ and deviation of differential motion vector length σ_{HL} are definitely higher, the value of mutual matching parameter $\bar{\eta}_{HL}$ is relatively low ($\bar{\eta}_{HL} < 0.8$ for *Football* in P-frames, $\bar{\eta}_{HL} < 0.81$ for *Crew* in P-frames). However, still in all cases much more than 50% blocks in low-resolution layer have been encoded using the same type of motion-compensated prediction and using the same reference frame as corresponding blocks in high resolution layer.

In all cases, the mutual matching parameter has lower values in B-frames than in P-frames. This is because there are more prediction modes available in B-frames than in P-frames.

5.4.7. Visualization of motion vector fields

In this section, examples of motion vectors visualization are presented. Motion fields for low resolution and high resolution video sequences are shown in the video frames with the lowest and the highest values of parameter $\overline{\Delta mv_{HL}}$. Additionally, a visualization of differential motion vector field is shown for these video frames. In order to keep legibility, motion vector fields are shown for P-frames only.

In Fig. 5.27-5.34 motion vectors estimated for resolutions QCIF-CIF in sequences *Bus*, *Football*, *Foreman* and *Mobile* are presented. In Fig. 5.35-5.38 motion vectors estimated for resolutions CIF-4CIF in sequences *City* and *Crew* are presented.

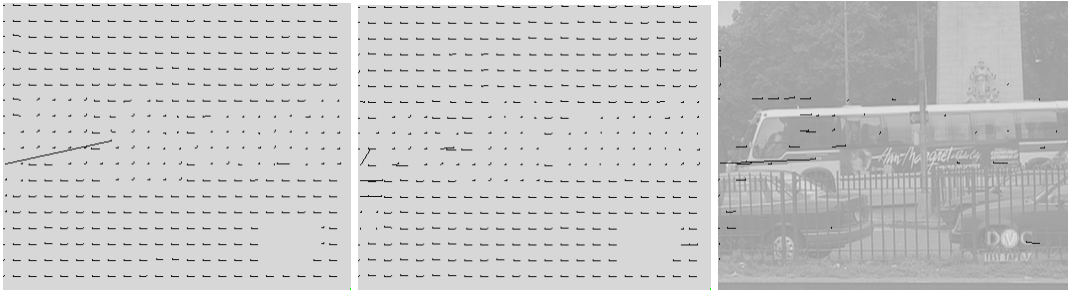


Fig. 5.27. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 4.735$, $\overline{\sigma_{HL}} = 18.051$). Frame number 73 from *Bus* sequence, $Q_P = 31$.

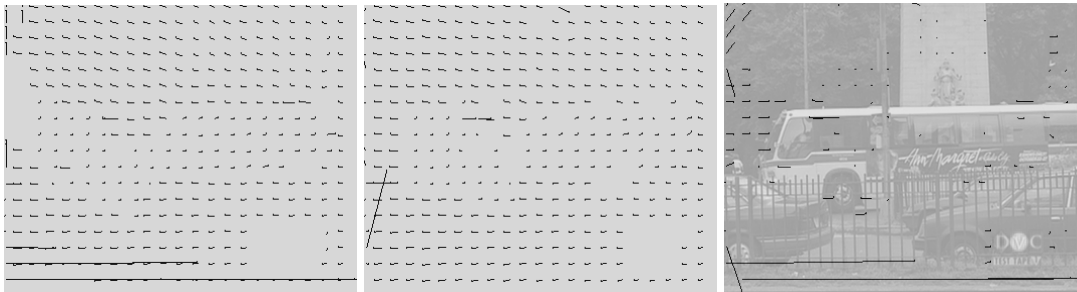


Fig. 5.28. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 26.672$, $\overline{\sigma_{HL}} = 136,323$). Frame number 61 from *Bus* sequence, $Q_P = 38$.

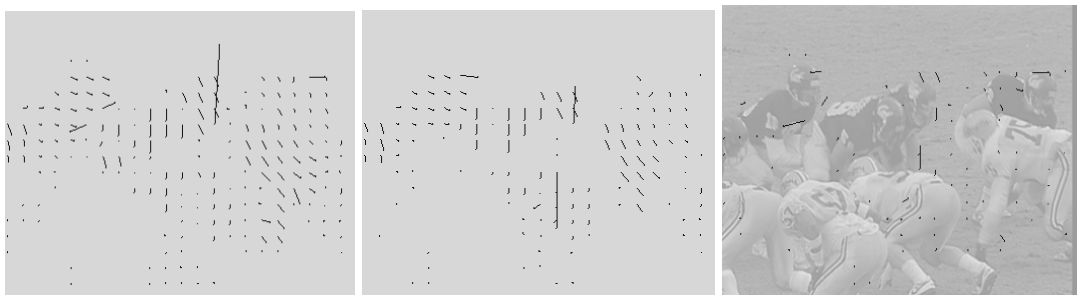


Fig. 5.29. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 5.095$, $\overline{\sigma_{HL}} = 20.322$). Frame number 3 from *Football* sequence, $Q_P = 33$.

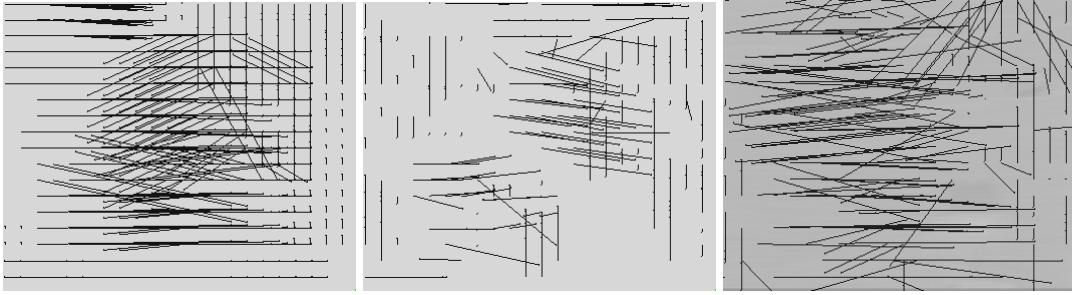


Fig. 5.30. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 367.138$, $\overline{\sigma_{HL}} = 279.839$). Frame number 87 from *Football* sequence, $Q_p=33$.

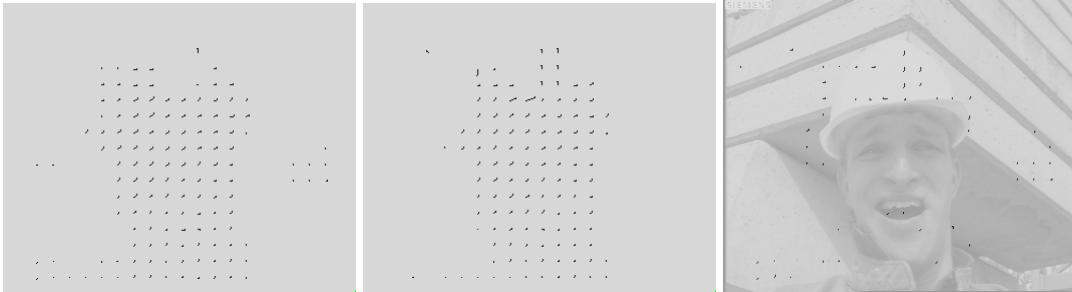


Fig. 5.31. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 2.3762$, $\overline{\sigma_{HL}} = 5.645$). Frame number 29 from *Foreman* sequence, $Q_p=25$.

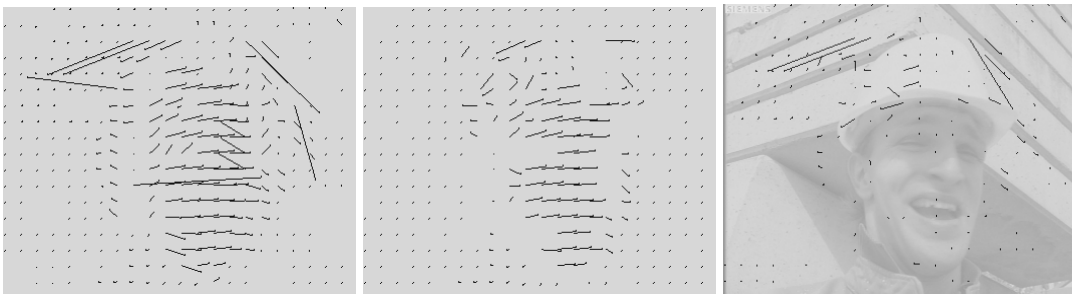


Fig. 5.32. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 16.263$, $\overline{\sigma_{HL}} = 38.508$). Frame number 15 from *Foreman* sequence, $Q_p=33$.

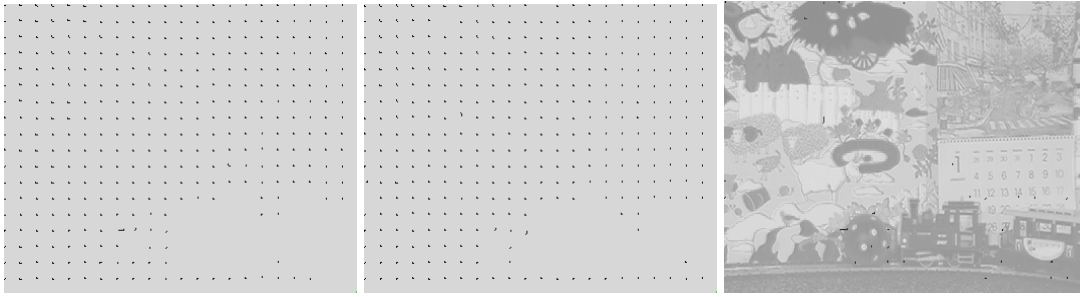


Fig. 5.33. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 1.405$, $\overline{\sigma_{HL}} = 2.602$). Frame number 148 from *Mobile* sequence, $Q_p=32$.

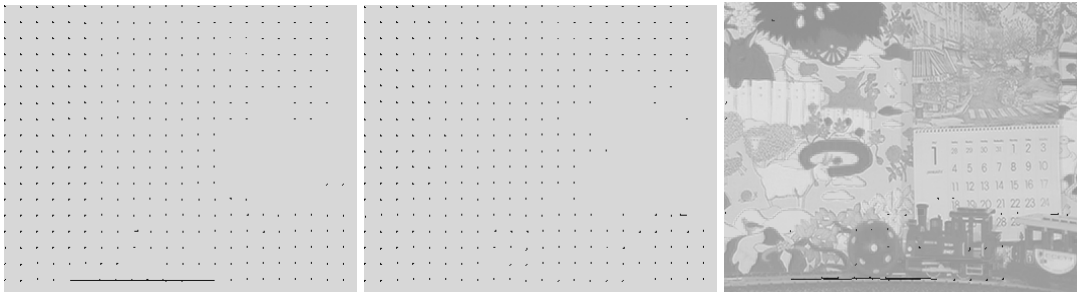


Fig. 5.34. Interpolated motion vectors estimated for resolution 176×144 , motion vectors estimated for resolution 352×288 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 4.545$, $\overline{\sigma_{HL}} = 20.517$). Frame number 104 from *Mobile* sequence, $Q_p=39$.

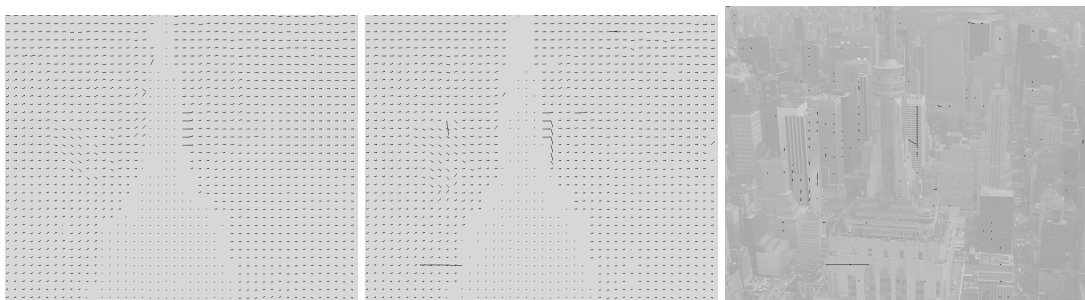


Fig. 5.35. Interpolated motion vectors estimated for resolution 352×288 , motion vectors estimated for resolution 704×576 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 2.813$, $\overline{\sigma_{HL}} = 10.738$). Frame number 61 from *City* sequence, $Q_p=25$.

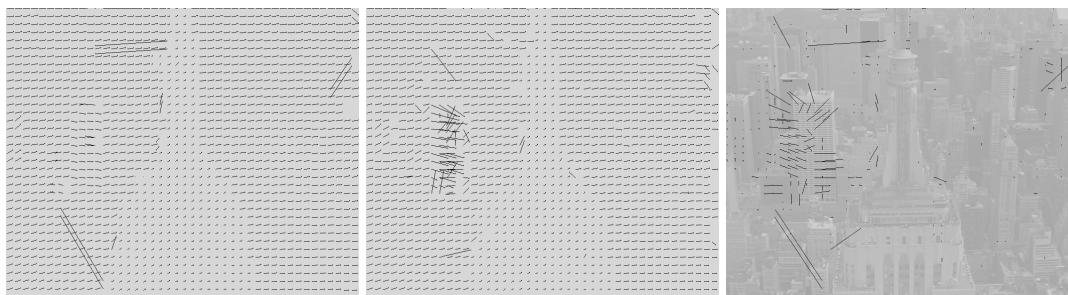


Fig. 5.36. Interpolated motion vectors estimated for resolution 352×288 , motion vectors estimated for resolution 704×576 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 9.239$, $\overline{\sigma_{HL}} = 39.769$). Frame number 124 from *City* sequence, $Q_P = 35$.

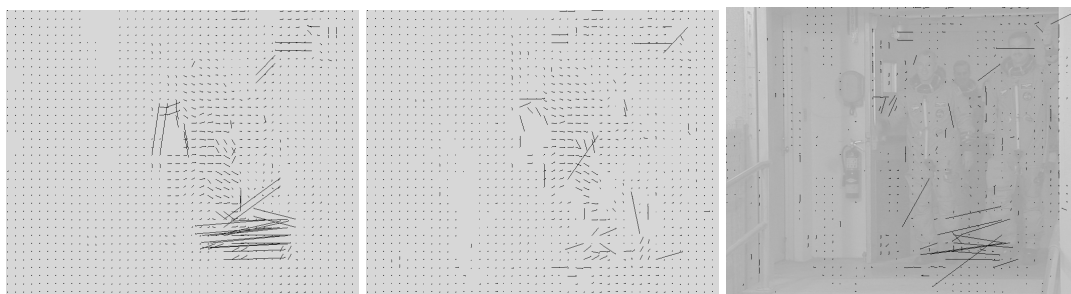


Fig. 5.37. Interpolated motion vectors estimated for resolution 352×288 , motion vectors estimated for resolution 704×576 and differential motion vectors in the frame with the lowest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 12.012$, $\overline{\sigma_{HL}} = 47.008$). Frame number 3 from *Crew* sequence, $Q_P = 30$.



Fig. 5.38. Interpolated motion vectors estimated for resolution 352×288 , motion vectors estimated for resolution 704×576 and differential motion vectors in the frame with the highest value of parameter $\overline{\Delta mv_{HL}}$ ($\overline{\Delta mv_{HL}} = 159.053$, $\overline{\sigma_{HL}} = 248.309$). Frame number 63 from *Crew* sequence, $Q_P = 41$.

Visualizations of motion vector fields presented in this section allow for comparison of obtained numerical values of chosen parameters $\overline{\Delta mv_{HL}}$, $\overline{\sigma_{HL}}$ and $\overline{\eta_{HL}}$ with subjective feelings about similarity of given motion vector fields.

Global smoothness of motion vector field can be observed for *Bus*, *Foreman*, *Mobile* and *City* sequences in Fig. 5.26, 5.27 and 5.30-5.35. On the other hand, rough motion vector fields are depicted in Fig. 5.29 and 5.37, for the chosen frames from *Football* and *Crew* sequences.

However, in all the presented visualizations, there can be observed similarities between motion vectors estimated for low-resolution video sequence and high-resolution video sequence.

5.5. Conclusions

Correlation of motion vectors has been researched between motion vectors in high spatial resolution video and low spatial resolution video. High correlation of motion vector fields is achieved mostly in sequences with slow and smooth motion (*Bus*, *Foreman*, *Mobile*, *City*). What is more, in all cases, the highest correlation occurs in frames predicted with bidirectional motion-compensated prediction (B-frames). For example, in *Mobile* sequence, the average length of a differential motion vector in full-pel units is always less than 1.2 samples in P-frames and always less than 0.625 samples in B-frames, as depicted in Fig. 5.6 and 5.14 respectively. In *Bus* and *Foreman* video sequences, the average length of differential motion vector in full-pel units is usually less than 2.5 samples in P-frames and less than 2 samples in B-frames.

On the other hand, in the video sequence with fast and rough motion (*Football*) the correlation between motion vectors from low resolution and motion vectors from high resolution is lower – the differential motion vector length tends to have high values, especially in P-frames: the maximum value of differential motion vector length is 91.8 samples in 87 frame of sequence *Football*, as depicted in Fig. 5.10. The average values of differential motion vector length are also high in *Football*: 13-15 samples in P-frames and 3.75-4 in B-frames.

The lower correlation between motion vector fields occurs also in the video sequence with rapid global illumination changes, like *Crew* (Fig. 5.8 and 5.14).

Lower correlation between low-resolution motion vector field and high-resolution motion vector field in the sequences containing fast motion and global illumination changes is caused by the operation of block matching algorithm of motion estimation. In BMA algorithm, estimated motion vectors point to the most similar blocks in reference pictures, therefore they do not represent the real motion (see Section 2.3). When the content of a video sequence changes rapidly (fast motion), BMA algorithm finds different blocks in low-resolution video sequence and high-resolution video sequence.

The increase of correlation causes the decrease of standard deviation of the differential motion vector field, which is especially noticeable in *Foreman* and *Mobile* video sequence (Fig 5.17, 5.18, 5.23 and 5.24). In this sequences differential motion fields are smoother, as depicted in Fig. 5.31 – 5.34.

Mutual matching of motion vectors from low resolution and high resolution video sequence, which is defined by mutual matching parameter $\bar{\eta}_{HL}$, is very high. In all video sequences except *Football* and *Crew*, more than 95% of the 4×4 blocks in P-frames in both resolutions are predicted using the same reference frame. Matching of motion vectors in B-frames is lower, because of more available prediction modes (forward, backward and bidirectional), but it is still up to 80% in *Bus*, *Foreman* and *City* sequences, up to 87.6% in *Mobile* sequence and up to 83.8% in *Crew* sequence.

High values of $\bar{\eta}_{HL}$ mean that in most cases, a motion vector from high-resolution video has its equivalent motion vector in low-resolution video. Therefore, the motion vector from low-resolution video can be used in order to encode motion vector from high-resolution video sequence.

For almost all video sequences (except *Football*), mutual correlation between multiresolution motion vectors increases with the increase of quality, as presented in Tab. 5.1 – Tab. 5.6. However, it remains almost constant, regardless of bitrate and quality.

Differential motion vector length for consecutive video frames insignificantly varies, as shown in Fig. 5.3 – 5.14. However, global correlation between motion vectors from high and low resolution can be observed, especially in video sequences with slow and moderate motion.

5.6. Summary

As it has been proven in this chapter, there are correlations between motion vectors estimated for different resolutions of the same video sequence, even when motion estimation is performed independently for each spatial resolution. These implicit correlations are somehow obvious – the motion is estimated for the same video content. Even if motion vectors do not always match the real, geometrical displacements in the video sequence, the general character of motion is kept, regardless of the spatial resolution. This is especially true for video sequences with slow and moderate motion. As the similarities between motion vectors estimated for low-resolution video and high-resolution video are high, motion vectors from base layer can be further exploited in enhancement layer of the scalable video coder in order to improve coding efficiency.

Another very important conclusion is that for most motion vectors in high-resolution layer there are matching motion vectors in low-resolution layer. It has been proven by high values of mutual matching parameter. Homogeneous motion fields in both layers of scalable video codec allow for joint encoding of motion information.

Therefore, in the following chapters, there are made attempts in order to improve the coding efficiency of motion vectors in scalable video coding.

In Chapter 6, further increase of correlation between motion vectors from base and enhancement layer is introduced by optical flow technique of motion estimation. Motion vectors are then encoded jointly in a bitstream.

In Chapter 7 the technique of inter-layer motion vectors prediction is proposed and applied into scalable codecs. This technique exploits existing correlations in independently estimated motion vectors.

In Chapter 8 the algorithm of simplified encoding of motion vectors in temporally scalable video codec has been proposed. The proposal significantly outperforms other techniques in a sense of encoder complexity with just a small impact on achieved compression efficiency.

Chapter 6.

Joint encoding of multiresolution motion vectors

6.1. Towards new standard of scalable video coding

In July 2002, the final draft of AVC/H.264 video coding specification has been approved [MP02-20]. The first version of the AVC/H.264 specification did not supported scalability. On the other hand, scalable profiles of former video coding standards, like MPEG-2 [ISO94], H.263 [ITU05] or MPEG-4 Visual [ISO98] have never been widely used, mainly because of complex algorithms and poor efficiency as compared to non-scalable profiles [Dom04].

However, during MPEG meeting on which the AVC/H.264 specification was approved, so called “ad hoc group” (AHG) on scalable video coding was established [MP02-35]. Soon after, requirements and applications for a new scalable video coding standard have been formulated and announced by MPEG organization [MP03-25]. It formally began the process of developing of a new scalable video coding standard.

In parallel, just after establishing AVC/H.264 recommendation, the author of this dissertation, began his research on representation of motion vectors in scalable video coding. The first technique was presented in author’s master thesis [Lan03a] and published later in September 2003 [Lan03]. Unfortunately, the proposed technique was not satisfactory enough, so further investigations were developed by the author.

In the end of 2003, "Call for Proposals on Scalable Video Coding Technology" was announced by MPEG [MP03-93]. As the answer, 21 proposals of scalable video codec were submitted. Among others, a proposal of Poznań University of Technology was presented [Bła04b], which exploited inter-layer prediction of motion vectors, developed by the author. Another important answer for “Call for Proposal” was "Scalable Extension of H.264/AVC" submitted by Heinrich Hertz Institute (HHI) from Berlin, which did not, however, exploit inter-layer correlations of motion vectors [Sch04]. These two algorithms proved to be extremely good in subjective comparison of coding efficiency organized by MPEG [Bar04]. In Fig. 6.1, the loss of Mean Opinion Score (MOS) of all submitted codecs against AVC/H.264 codec has been presented for one of coding scenarios.

The HHI proposal was later chosen by MPEG as the basis for Scalable Video Model (SVM) development, in order to establish Scalable Video Coding (SVC) specification. In the first version of SVM description [MP04-72], the following tools were mentioned as possible solutions for motion vectors representation:

- non-scalable coding,
- resolution scalable coding,
- quality scalable coding,
- block size scalable coding.

Additionally, CABAC-based entropy coding and VLC-based entropy coding of motion vectors was considered.

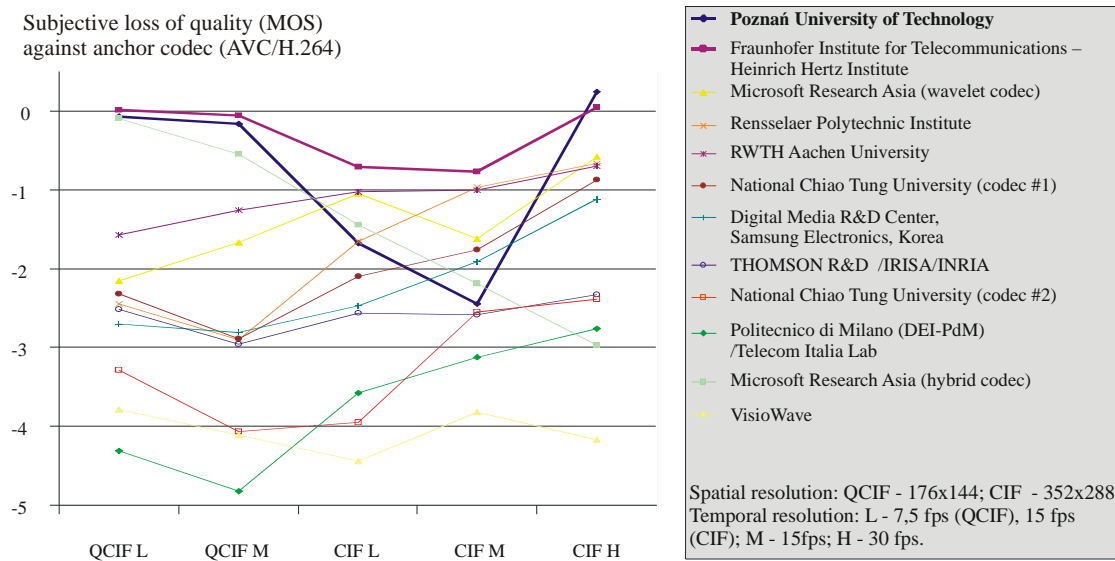


Fig. 6.1. The loss of Mean Opinion Score (MOS) against AVC/H.264 in subjective quality comparison of scalable codecs, March 2004 [Bar04].

Later on, the author continued his investigations on various aspects of inter-layer representation of motion vectors in scalable video codecs. Some of them have been presented further in this dissertation.

The subsequent version of Scalable Video Model [MP04-20] was approved during the next MPEG meeting in July 2004. Two modes of inter-layer motion prediction appeared in the second version of SVM and stay unchanged during later development. These modes are widely described in this dissertation in Section 7.7.2.

In Tab. 6.1, the timeline of parallel activities of MPEG and the author of this thesis have been presented. The presented events regard the hottest period during the development of a Scalable Video Coding.

At this time (September 2006), SVC standard is in the stage of final agreements and should be approved in the near future.

Tab. 6.1. The timeline of MPEG works on Scalable Video Coding compared against author's activities.

date	author's activity	MPEG activity
2002.07		– Final draft of version 1 of AVC/H.264 specification [MP02-20]. – MPEG establishes AHG on Scalable Video Coding.
2003.01	The beginnings of research on motion vectors coding in scalable video coder.	
2003.09	First author's publication on motion vectors coding in scalable video coder [Lan03].	
2003.10		"Requirements and Applications for Scalable Video Coding" [MP03-25]
2003.10 –	Development of inter-layer motion vectors representation in scalable AVC codec.	
2003.12		"Call for Proposals on Scalable Video Coding Technology" [MP03-93]
2004.02-2004.03	The comparison of scalable codecs submitted for MPEG's Call for Proposals.	
2004.03	"Scalable AVC Codec" [Bla04b] with author's inter-layer prediction of motion vectors.	– „Subjective Test Results for the CfP on Scalable Video Coding Technology“ [Bar04] – "Scalable Extension of H.264/AVC" [Sch04] is chosen as the basis for developing of a new standard, no inter-layer prediction of motion vectors in the codec. – "Scalable Video Model V 1.0" [MP04-72]
2004.03 –	Further works on multiresolution coding of motion vectors.	
2004.07		"Scalable Video Model 2.0" contains inter-layer prediction of motion vectors [MP04-20].
2006.01 2006.04	Improvement proposals and reports on efficiency of motion vectors coding in SVC [Lan06, Lan06a, Lan06b]	

This, and the following chapters present original results obtained by the author during his research on multiresolution representation of motion vectors in scalable video codecs.

In this chapter, the very first author's method of joint motion estimation and joint encoding of motion vectors in a scalable video coder has been presented. In Chapter 7, an original method of inter-layer motion vectors prediction has been introduced and experimentally tested in two scalable codecs: AVC-based scalable codec developed at Poznań University of Technology and SVC video codec, developed by MPEG. In Chapter 8, a method of very fast and yet efficient encoding of multiresolution motion vectors in temporally scalable codec has been presented and experimentally tested.

6.2. Introduction to joint encoding of multiresolution motion vectors

In Chapter 5, there has been proven general correlation between motion vectors estimated for different spatial resolutions of the same video sequence. However, independent motion estimation using block matching algorithm introduces some local disturbance and mismatches between motion vectors from low and high resolution of the video sequence.

The idea behind the proposal of joint encoding of motion vectors in scalable video coder is to increase correlation between motion fields for different spatial resolutions. Residual signal is obtained by subtracting motion vectors in low-resolution video from co-located motion vectors in high-resolution video. Motion vector residuals are spatially predicted and encoded jointly. The proposed approach assumes smooth motion vector field in each layer of the scalable coder. In order to increase the inter-layer correlation, motion vectors are estimated using optical flow technique. The trade-off between smoothness of motion field and energy of prediction error is obtained by modification of motion estimation algorithm in the scalable hybrid coder.

Smooth motion vector field is expected to be encoded efficiently in the base layer due to differential encoding of motion vector residuals. On the other hand, encoding of motion vectors in enhancement layer relying on motion data from base layer can yield a profit due to correlation between motion vector field from the base layer and motion vector field from the enhancement layer.

Details and more precise experimental results of the presented method of joint encoding of multiresolution motion vector field are given in Master of Science thesis “Estimation and coding of motion vectors in scalable video coders” [Lan93a].

6.3. Modification of scalable coder

In the following experiments the basis for modifications was spatially scalable video coder described in Section 2.4.2. In the researched configuration, the coder consists of two hybrid sub-coders (Fig. 6.2) that produce bitstreams corresponding to different levels of spatial decomposition. The base layer represents a video sequence with reduced spatial resolution (QCIF) while the enhancement layer represents a video sequence with full spatial resolution (CIF). Originally each of the sub-coders had its own prediction loop with independent motion estimation [Dom03, Bła03]. The low-resolution sub-coder was implemented as a standard motion-compensated hybrid AVC/H.264 coder. The high-resolution sub-coder was a modified AVC/H.264 coder that was able to exploit decoded samples from base-layer bitstream, as described in Section 2.4.2.

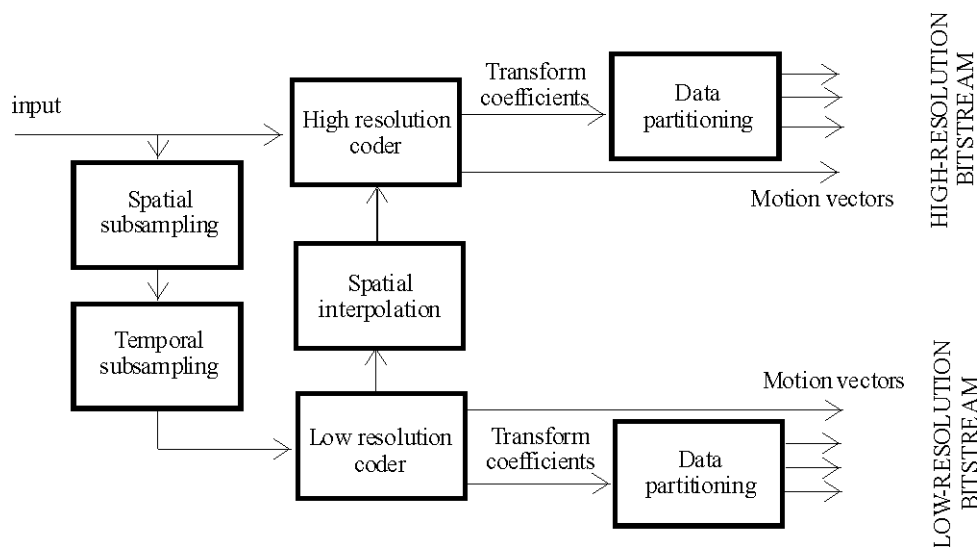


Fig. 6.2. Basic structure of scalable hybrid video coder with independent motion estimation and compensation.

Some modifications have been introduced into scalable video coder from Fig. 6.2 in order to jointly encode the motion vectors from the base and enhancement layers. First, the standard block matching motion estimation has been replaced with optical flow motion estimation in both base-layer and enhancement-layer sub-coders.

Another modification has been introduced into motion vector coding scheme in the enhancement layer. Motion vector prediction has been changed: median prediction of difference between enhancement-layer motion vector and corresponding base-layer motion vector is applied. Base-layer motion vectors are used only when corresponding macroblocks are coded using motion-compensated prediction.

The structure of modified scalable coder with joint motion estimation and joint coding of motion vectors is depicted in Fig. 6.3.

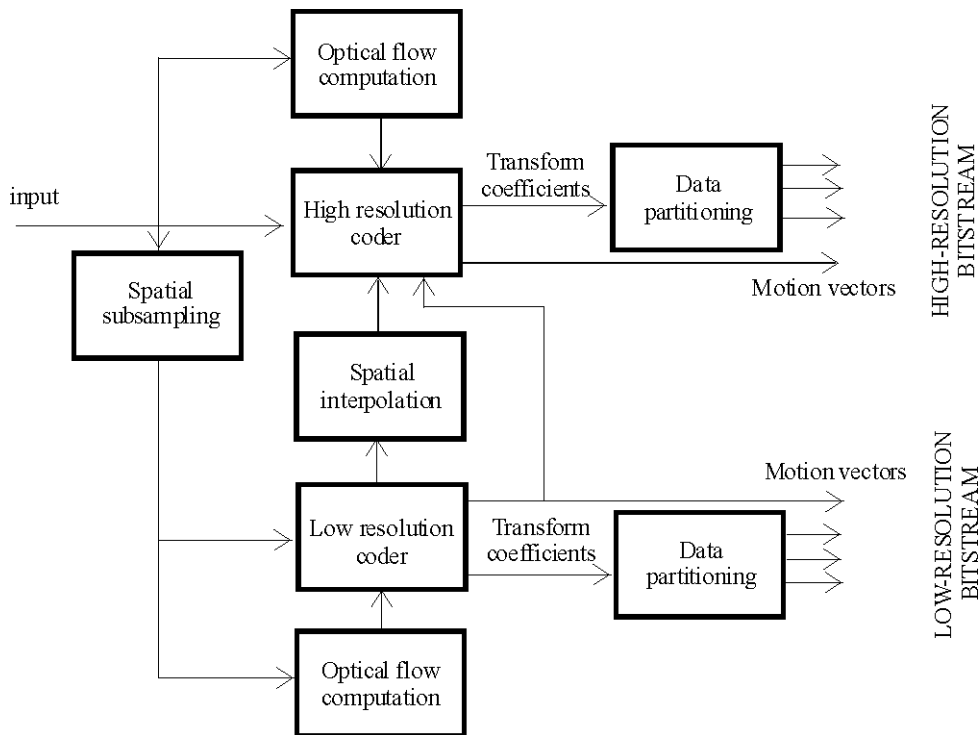


Fig. 6.3. Structure of scalable coder with joint motion estimation and motion vector encoding.

6.3.1. Joint motion estimation

The increase of correlation between motion vector fields from base and enhancement layers is achieved by using an optical flow based algorithm of motion estimation.

Optical flow field is the two-dimensional distribution of velocities of luminance values, defined for every pixel over image area. Optical flow is a kind of estimate of optical displacement in the video sequence.

Some limitations in estimation of motion by the optical flow algorithm are worth noticing: for example, the optical flow is not equal to zero for a stationary scene with changes of brightness. However, in most cases, optical flow is a good approximation of true motion in video sequence [Bar94, Kri97a].

In the video coder from Fig. 6.3 Horn-Schunck algorithm was applied in order to estimate optical flow in a low resolution and a high resolution video sequence. The algorithm combines the gradient constraint with a global smoothness term to constrain the estimated optical flow field $v(x,y,t)$. In order to estimate the optical flow, the value of following integral is minimized:

$$\int_D (\nabla I(x, y, t) * \underline{v}(x, y, t) + \frac{\partial I(x, y, t)}{\partial t})^2 + \lambda^2 (\|\nabla u(x, y, t)\|_2^2 + \|\nabla v(x, y, t)\|_2^2) dx \quad (6.1)$$

where:

$I(x,y,t)$ – the values of luminance samples at the location (x,y) in the moment t ,

$\underline{v}(x,y,t)$ – the optical flow vector at the location (x,y) in the moment t ,

$$\underline{v}(x, y, t) = \begin{bmatrix} u(x, y, t) \\ v(x, y, t) \end{bmatrix}$$

λ – parameter controlling smoothness of optical flow field,

$\| \cdot \|_2$ – vector norm l_2 .

The integral (6.1) is defined over a domain D , which is the image area. The parameter λ is used in order to control the influence of smoothness term. It was chosen experimentally ($\lambda = 0.01$) [Lan03a].

Algorithmically, iterative equations are used in order to minimize (6.1) for each estimated vector $\underline{v}(x,y,t)$:

$$u^{k+1} = \bar{u}^{-k} - \frac{I_x[I_x * \bar{u}^{-k} + I_y * \bar{v}^{-k} + I_t]}{\lambda^2 + I_x^2 + I_y^2} \quad (6.2)$$

$$v^{k+1} = \bar{v}^{-k} - \frac{I_y[I_x * \bar{u}^{-k} + I_y * \bar{v}^{-k} + I_t]}{\lambda^2 + I_x^2 + I_y^2} \quad (6.3)$$

where:

- \bar{u}, \bar{v} – the component-wise averages of adjacent vectors,
- k – iteration step,
- I_x, I_y, I_t – partial derivatives, $I_x = \frac{\partial I(x, y, t)}{\partial x}$, $I_y = \frac{\partial I(x, y, t)}{\partial y}$, $I_t = \frac{\partial I(x, y, t)}{\partial t}$.

Estimated optical flow field is very dense. Therefore, a problem of decimation of this field appears in order to use it in application of video coding with block-based motion-compensated prediction. The author proposed the following method of motion field decimation in the scalable video coder: a motion vector that appears most often in a block of given size is chosen as the representative motion vector. The chosen full-pel motion vector is used as the initial vector to perform sub-pel motion vector estimation in order to minimize prediction error. Sub-pel motion vector refinement is performed using block matching algorithm. Such an algorithm is used both in the base layer and in the enhancement layer of the scalable coder. It produces smooth motion vector fields in both layers.

Since motion field is estimated based on optical displacements, the increased correlation between motion vectors in the base layer and motion vectors in the enhancement layer is assured. On the other hand, the sub-pel refinement of motion vectors using block matching algorithm minimizes the prediction error.

6.3.2. Joint multiresolution motion representation

In the following scheme, the motion vectors from the base layer are used for differential encoding of the motion vectors from the enhancement layer of the scalable codec. First, the base-layer motion field is interpolated as depicted in Fig. 6.4.

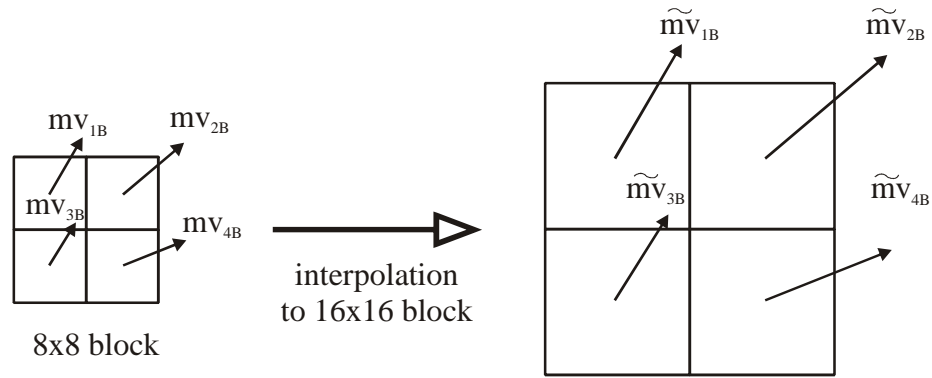


Fig. 6.4. Interpolation of the base-layer motion vectors.

The values of interpolated motion vectors are rescaled according to equations:

$$\tilde{mv}_{1B} = 2 \cdot mv_{1B} \quad \tilde{mv}_{2B} = 2 \cdot mv_{2B} \quad \tilde{mv}_{3B} = 2 \cdot mv_{3B} \quad (6.4)$$

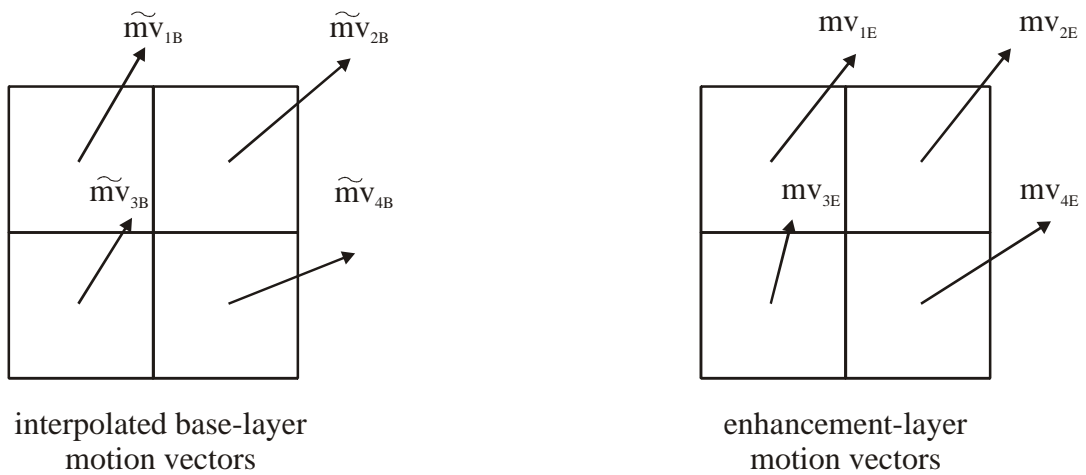


Fig. 6.5. Motion vectors from base layer and enhancement layer used in joint motion vectors representation.

The difference between corresponding motion vectors from the base layer and motion vectors from the enhancement layer (Fig. 6.5.) is calculated in order to form the residual value. Spatial motion vector prediction is then performed using median scheme (see Section 4.2) and the residual prediction value Δmv (6.7) is represented in the bitstream using entropy coding engine. The foregoing process is described by equations:

$$\Delta mv_1 = med(mv_{1E} - \tilde{mv}_{1B}, mv_{2E} - \tilde{mv}_{2B}, mv_{3E} - \tilde{mv}_{3B}) \quad (6.5)$$

$$\Delta mv_2 = mv_{4E} - \tilde{mv}_{4B} \quad (6.6)$$

$$\Delta mv = \Delta mv_1 - \Delta mv_2 \quad (6.7)$$

When any of co-located base-layer motion vectors is not present due to intra mode, coding of motion vectors in enhancement layer is performed in standard way described in AVC specification.

6.4. Joint encoding of multiresolution motion vectors – experimental results

In the experiments two methods of motion estimation have been compared in AVC/H.264-based scalable video coder: block matching motion estimation and optical flow motion estimation. The efficiency of scalable coder was researched, when joint motion estimation using optical flow and joint motion vectors coding was applied using the technique described in Section 6.3.

In this chapter, only a few examples of obtained results are presented. More experimental results can be found in [Lan93a].

6.4.1. Motion estimation using optical flow technique

In Fig. 6.6 and Fig. 6.7 block matching motion estimation is compared against optical flow technique in the base layer of scalable bitstream. Bitrates of the different parts of the bitstream are shown for *Basket* and *Fun* video sequences coded with AVC/H.264-based scalable video coder..

The experiments were performed [Lan03a] using AVC/H.264-based scalable video coder described in section 2.4.2. In Fig. 6.8 and Fig. 6.9 the comparisons of estimated motion vector field are depicted for different method of motion estimation.

The following parameters have been set in the configuration file of the encoder:

- only first picture coded as I-frame,
- group of pictures: I-P-P-P-P,
- number of reference frames: 1,

- entropy coding: CABAC,
- range of quantization parameter Q_p : 16-32.

Bitrate was measured separately for motion vectors and transform coefficients. The total bitrate was also measured in order to compare the overall efficiency of the video codec.

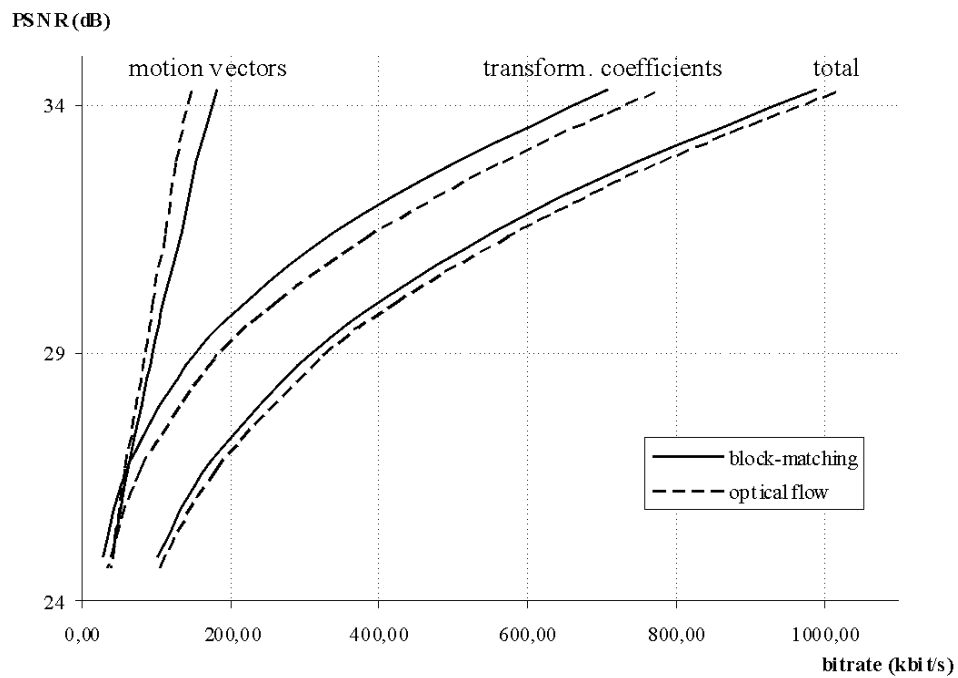


Fig. 6.6. Bitrates of motion vectors, transform coefficients and total bitrate in *Basket* sequence (352×288, IPPP), AVC/H.264 video codec. Various algorithms of motion estimation have been used.

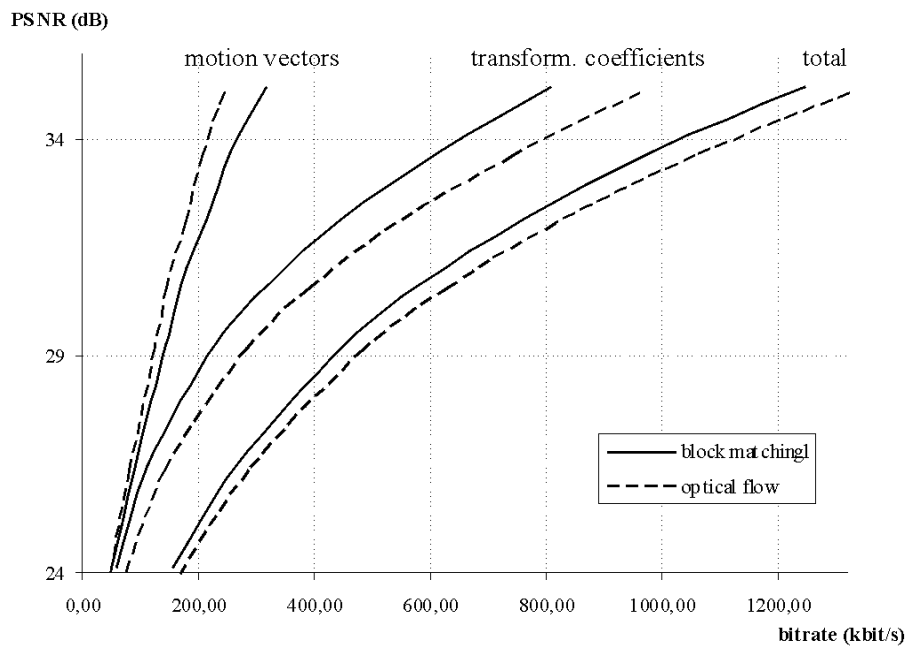


Fig. 6.7. Bitrates of motion vectors, transform coefficients and total bitrate in *Fun* sequence (352×288, IPPP), AVC/H.264 video codec. Various algorithms of motion estimation have been used.

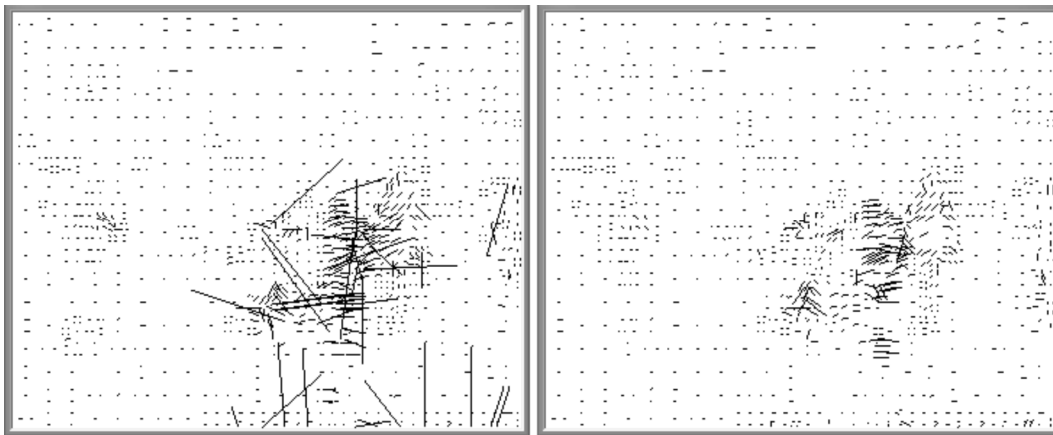


Fig. 6.8. Motion vectors estimated using block matching (on the left) and optical flow (on the right). Frame number 17 from *Basket* sequence.

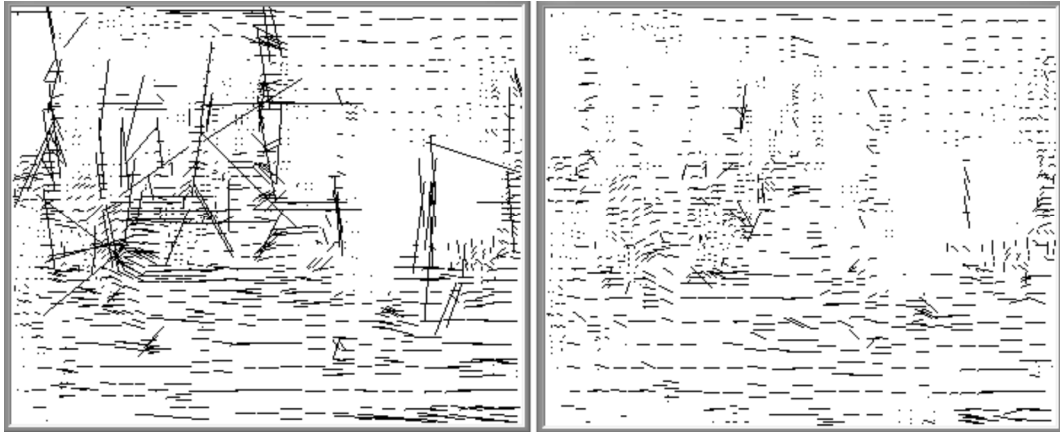


Fig. 6.9. Motion vectors estimated using block matching (on the left) and optical flow (on the right). Frame number 10 from *Fun* sequence.

When motion estimation is performed based on optical flow technique the motion vector bitrate is lower than when block-matching algorithm is used (Fig. 6.6 and 6.7). Since estimated motion field is smooth (Fig. 6.8 and 6.9), spatial prediction is very efficient and gives low prediction residuals. On the other hand, the modified technique of motion estimation gives worse efficiency of motion-compensated prediction, thus the total bitrate increases.

6.4.2. Joint encoding of motion vectors in scalable video codec

Figures 6.10 to 6.14 depict bitrates achieved in the experiments, when coding of enhancement-layer motion vectors using interpolated base-layer motion vectors was applied in the scalable video coder. Five different test sequences were used: *Basket*, *Stefan*, *Fun*, *Football* and *Cheer*.

The experiments were performed [Lan03a] using AVC/H.264-based scalable video coder described in section 2.4.2. The following parameters have been set in the configuration file of the scalable, AVC-based encoder:

- only first picture coded as I-frame,
- group of pictures: I-P-P-P-P,
- number of reference frames: 1,
- entropy coding: CABAC,
- quantization parameter in base layer $Q_P = 16$.
- range of quantization parameter in enhancement layer Q_P : 16-32.

Bitrate was measured separately for motion vectors and transform coefficients. The total bitrate was also measured in order to compare overall efficiency of modified scalable video codec.

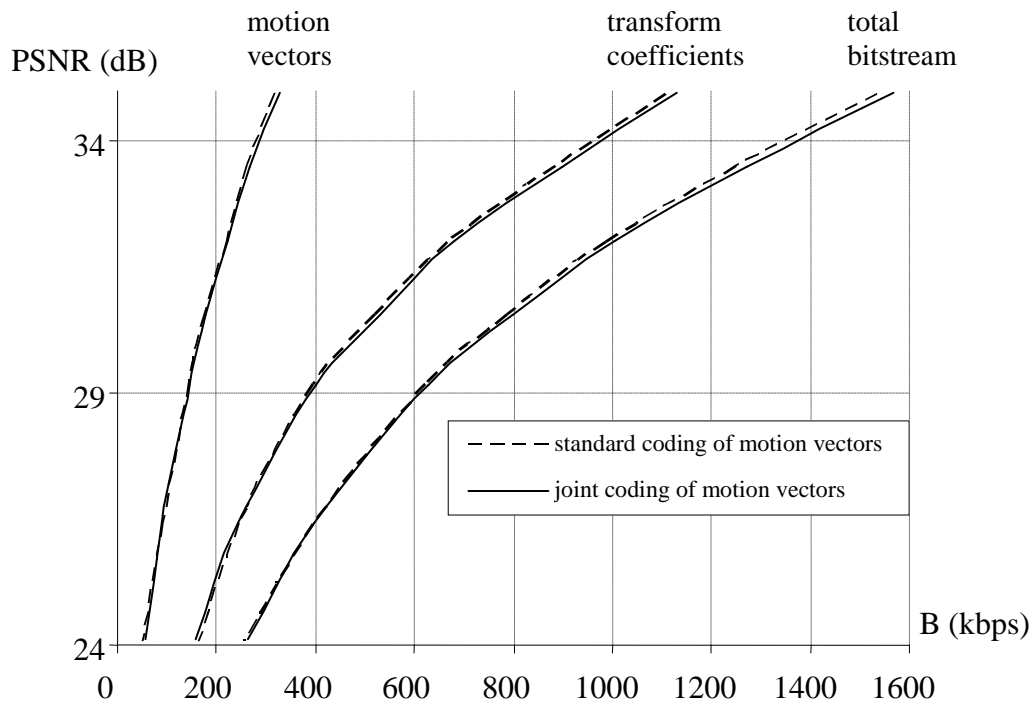


Fig. 6.10. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in *Basket* sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.

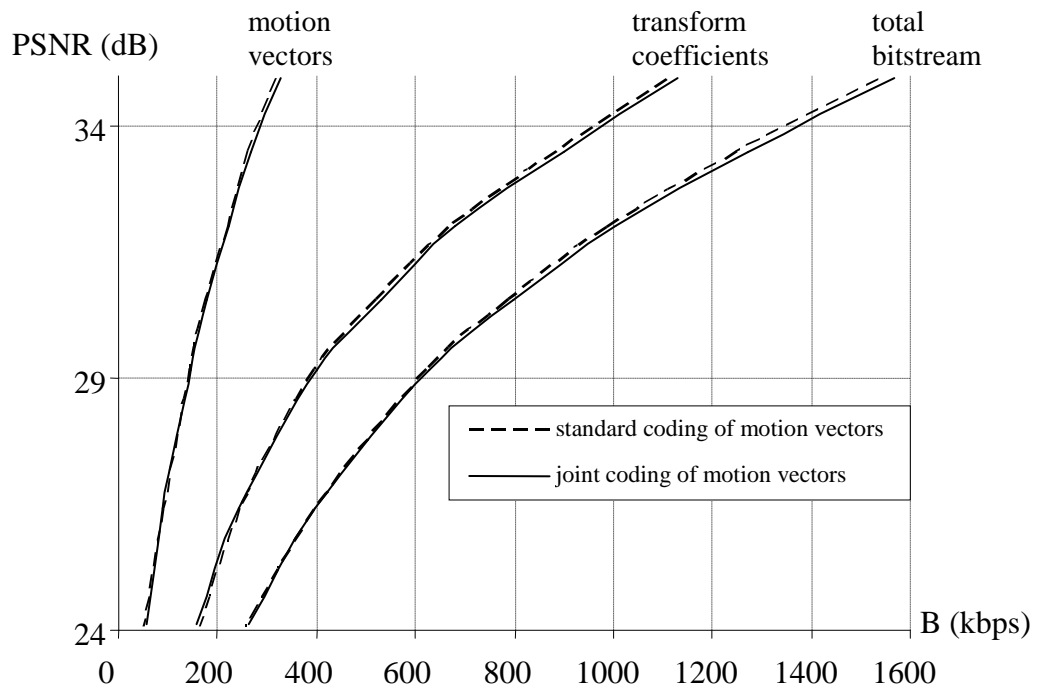


Fig. 6.11. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in *Stefan* sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.

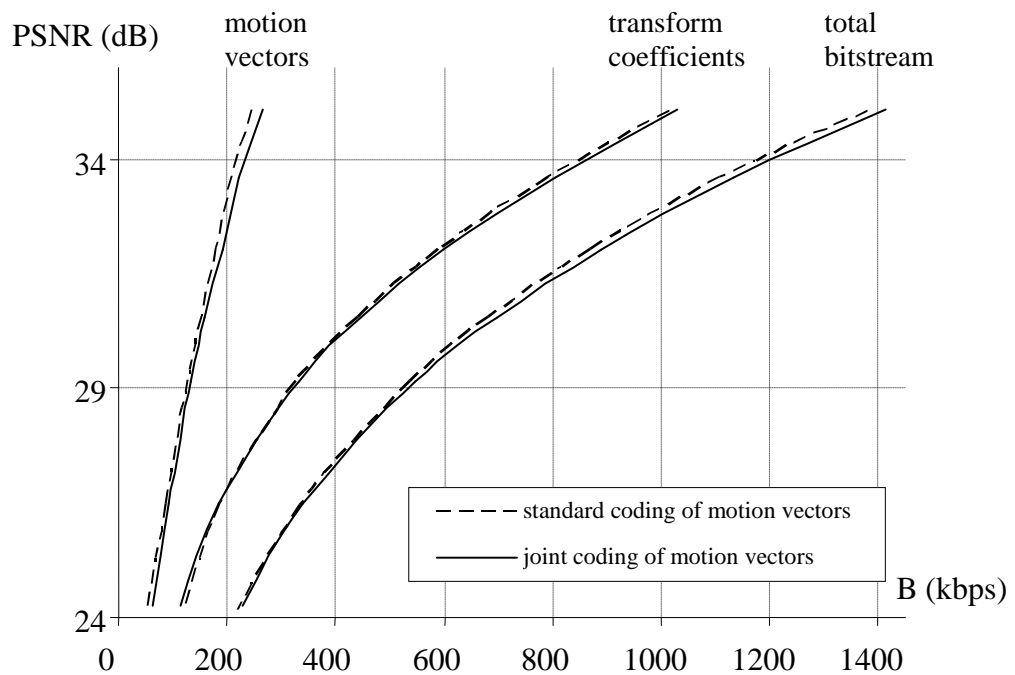


Fig. 6.12. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in *Fun* sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.

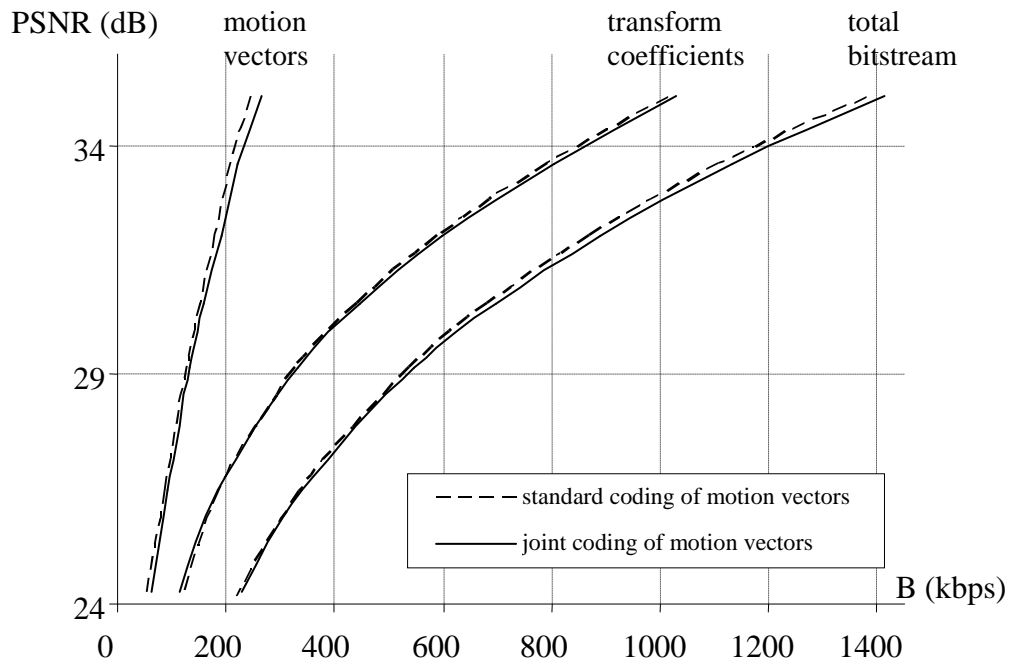


Fig. 6.13. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in *Football* sequence (352×288, IPPP), scalable video coding.

Various algorithms of motion vectors encoding.

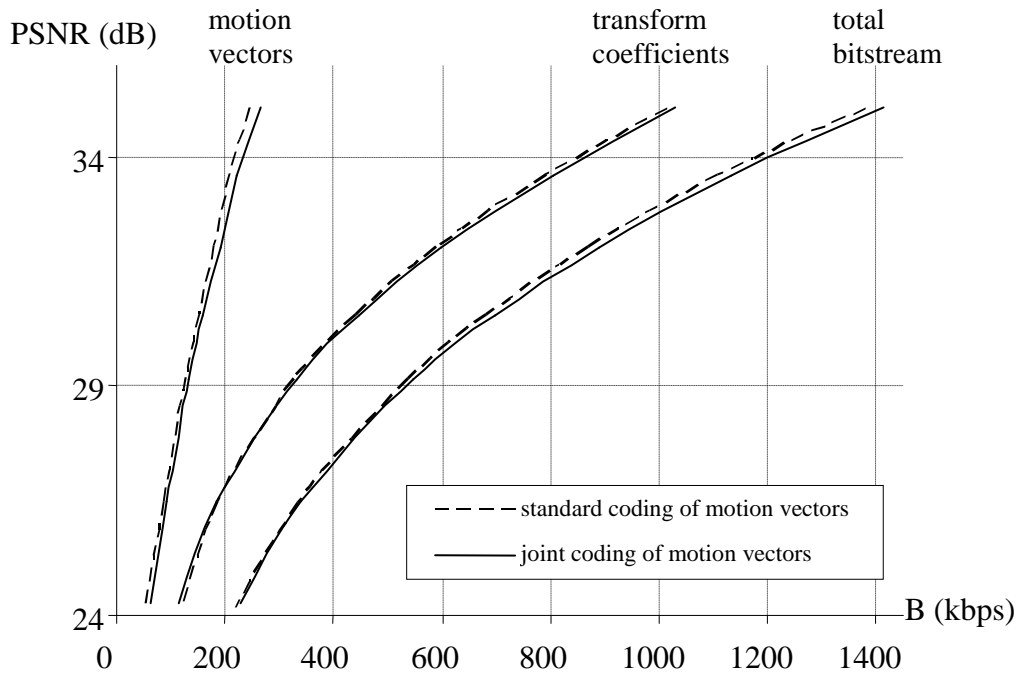


Fig. 6.14. Bitrates of motion vectors, transform coefficients and total bitrate in enhancement layer in *Cheer* sequence (352×288, IPPP), scalable video coding. Various algorithms of motion vectors encoding.

The proposed algorithm of motion vector representation in the enhancement layer is less efficient than the original algorithm with independent motion vector coding. In all cases, the proposal gives higher bitrate in the enhancement layer. Higher bitrate is obtained either for motion vector bitstream and transform coefficient bitstream.

6.5. Conclusions

The following two experiments have been described in previous sections:

1. comparison of a video encoder with block-matching motion estimation against video encoder using motion estimation based on optical flow technique,
2. comparison of two types of motion vectors encoding in scalable video codec with motion estimation based on optical flow technique:
 - independent coding of motion vectors in each layer,
 - author's original method of joint coding of motion vectors in base and enhancement layers using differential scheme with median prediction.

In Fig. 6.6 and Fig. 6.7, the results of the first experiment are presented. According to the results depicted in these figures, smooth motion field obtained using optical flow technique brings decrease of motion vector bitrate as compared to block matching algorithm of motion estimation. The decrease of bitrate is up to 15% and depends on the bitrate and content of a video sequence. This is in accordance with the expectations: optical flow gives smooth motion vector field, in which adjacent motion vectors are highly correlated (as depicted in Fig. 6.8 and 6.9). Therefore, spatial prediction of motion vector components works perfectly and produces very low residuals which are efficiently represented in a bitstream.

However, motion compensation using vectors estimated with optical flow usually gives worse prediction of samples, because optical flow does not explicitly minimize prediction error. As a result, the transform coefficients bitrate increases. The increase of transform coefficients bitstream is almost independent of the bitrate; it varies from about 10% in *Basket* sequence to about 20% in *Fun* video sequence.

Transform coefficients bitstream makes up a majority of a total bitstream. Therefore, with the increase of coefficients bitrate, the total bitrate increases as well (about 4% in

Basket, about 8% in *Fun*). As a result, motion estimation with optical flow technique has degraded overall coding efficiency.

Using joint encoding of motion vectors in scalable video coding (Section 6.4.2) has not brought the expected improvement. In all video sequences, motion vector bitstream, transform coefficients bitstream and total bitstream in enhancement layer increased when joint coding of motion vectors was applied. The increase of total bitstream is very low (from 0.8% in *Football* up to 3% in *Fun*), however it excludes the proposed technique.

Possible causes of unsatisfactory results of the experiments are complex motion model and extraordinary efficiency of standard motion vector coding scheme used in AVC/H.264. On the other hand, binarization and context modelling in CABAC entropy coding was designed exclusively for given statistical distribution of motion vector residuals. After modification of coding algorithm, the distribution has changed, and the entropy coding engine has not matched the new values of motion vectors residuals.

6.6. Summary

In this chapter, the very first author's approach to coding of motion vectors in scalable video codec has been presented. The increase of correlation between motion vectors from the base layer and motion vectors from the enhancement layer has been achieved by motion estimation using optical flow technique. Vectors in the enhancement layer have been jointly coded using vectors from the base layer. However, the proposal has not improved overall coding efficiency: the obtained bitrates have been worse than when standard method was applied.

It turned out that it is not worth interfering in rate-distortion optimized video coding. The best performance of scalable codec has been obtained, when independent motion estimation has been performed in the base and enhancement layer using block matching algorithm.

On the other hand, existing strong correlations between motion vectors from low-resolution layer and high-resolution layer can be exploited in a different way. The attempt is presented in the following chapter.

Chapter 7.

Multiresolution prediction of motion vectors

7.1. Introduction

Predictive coding of motion vectors is very important for efficient compression of motion vectors in a hybrid video codec. Moreover, a scalable video encoder that produces a bitstream containing many layers of spatially scalable video representation provides even more motion data that has to be transmitted to the receiver. Because of the multiple motion vector fields, the problem of efficient representation of motion data becomes more crucial. Beside intra-layer AVC-like motion vector prediction, additional inter-layer motion correlation can be exploited.

Experiments from Chapter 6 have proven that joint motion estimation in the low-resolution and high-resolution video sequence does not improve the coding efficiency. The best compression has been achieved using independent motion estimation in the low-resolution layer and the high-resolution layer of the scalable video coder.

On the other hand, the experimental results from Chapter 4 have shown that standard methods of intra-layer motion vectors prediction are very efficient. There is only small room for improvements, since most prediction residuals are very small (up to 75% of residuals are equal to zero).

However, motion vectors estimated for a low-resolution video sequence and the respective high-resolution video sequence are highly correlated, even when independent motion estimation algorithms are used. It has been proven in Chapter 5. Therefore, the idea that has been presented in this chapter exploits interpolated motion vectors from the low-resolution layer when no proper motion vectors exist in high-resolution layer. Interpolated motion vectors are then used in standard motion vectors prediction.

The presented approach was originally developed for scalable video coder, based on state-of-the-art AVC technology, which has been shortly discussed in Section 2.4.2. As described in Section 6.1, it has been very first successful implementation of the so-called “inter-layer motion prediction” in advanced scalable video coding.

7.2. Difficulties in spatial prediction of motion vectors

In advanced algorithms of video coding, sophisticated methods of motion vector prediction are used in order to represent motion field compactly. A codec based on

AVC/H.264 technology uses two schemes of spatial motion vector prediction: directional prediction and median prediction, as was discussed in Section 4.2.

It has been proven that in most cases, these standard predictions that use spatially adjacent motion vectors, produce very small prediction residuals. In consequence, residual motion vectors data are represented in a bitstream with a small number of bits.

However, the experiments from Section 4.2.3 proved that there are still some cases, when motion vector prediction does not match the actual value of the coded motion vector. In such a case, the produced residual has a significant value, thus it is less efficiently represented in a bitstream. For example in Tab. 4.3 as much as 14.6% of the motion vector residuals have the values more than 10 in the units of $\frac{1}{4}$ -sample in the sequence *Ice* at bitrate 3 Mbit/s. In these cases, standard prediction of motion vectors worked less efficiently.

The problem is that we do not know a priori all cases in which spatially adjacent motion vectors are just a little correlated with the current motion vector. In other words, we do not know a priori, in which cases, standard method of motion vector prediction gives large values of residuals.

However, there are still some cases that we do know a priori that the adjacent motion vectors are weakly correlated with the coded motion vector. These are when:

- adjacent blocks do not use the same type of motion-compensated prediction as the current block,
- adjacent motion vectors refer to a different reference frame than the current motion vector,
- current macroblock is a boundary macroblock,
- adjacent macroblocks belongs to a different slice of macroblocks,
- adjacent macroblocks use intra-frame coding.

In all these case, motion vector prediction is poor, because there are no proper motion vectors to perform a good prediction.

7.2.1. Boundary of frame or slice

Video coding algorithms allow macroblocks to be organized in the structures called *slices of macroblocks*. Slices introduce data partitioning and error resilience into

transmission of digital video [Wie03, Ohm04]. The beginning of a new slice resets the state of the decoder, which allows for independent decoding of each slice. Macroblocks from two different slices are decodable completely independently from each other – no prediction is performed across slice boundary.

On the other hand, partitioning macroblocks into slices allows for grouping the macroblocks with similar content, for example solid background or rich texture. Encoder sets independently the encoding parameters for each slice in order to encode these macroblocks in the most efficient way.

In AVC/H.264 coding algorithm, macroblocks are allowed to be grouped together in many ways using the technique named Flexible Macroblock Ordering (FMO). Different modes of grouping macroblocks into slices are depicted in Fig. 7.1 [Dho05, ISO06].

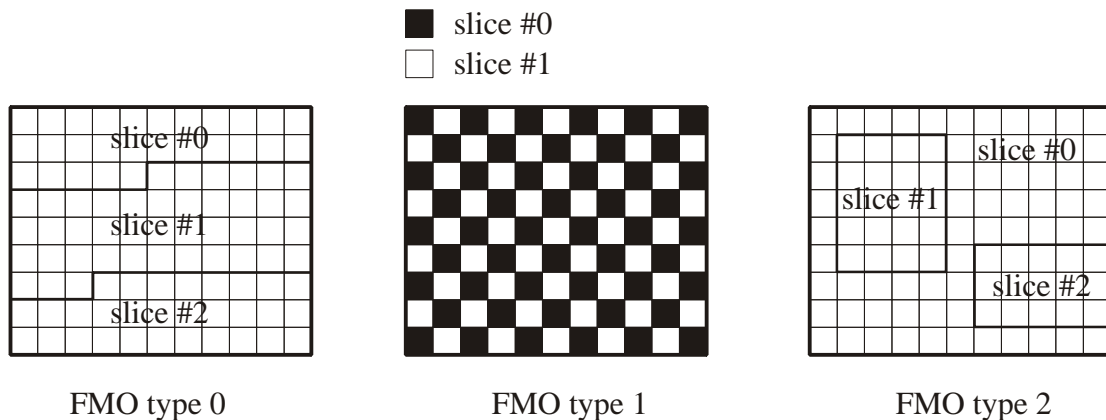


Fig. 7.1. Examples of grouping macroblocks into slices using FMO in AVC/H.264 codec.

FMO technique has been introduced mainly in order to protect bitstream against transmission errors. However, compression efficiency can be also improved in some cases.

For example, FMO type 0 from Fig. 7.1 prevents errors from spreading across a frame. Since each slice is independently decodable, the prediction can not be performed across slices boundary. FMO type 1 allows for random distributing reconstruction artifacts in the case of transmission errors. On the other hand, possible errors can be easily concealed in such a case by spatial interpolation [Wan98]. FMO type 2 is used in order to group homogenous areas in the video frames. They can be encoded using specific tools and given bitrate.

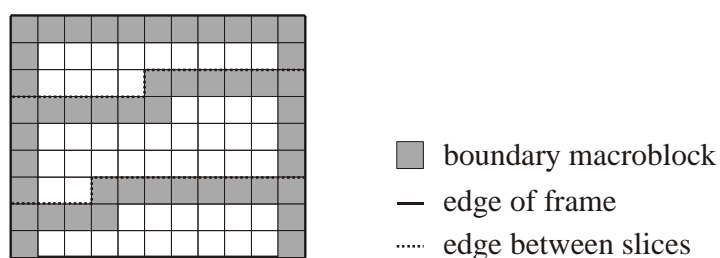


Fig. 7.2. Boundary macroblocks.

When many slices appear in a single video frame, many boundary macroblocks occur as depicted in Fig. 7.2. A macroblock is treated as a boundary macroblock when it exists at the edge of a frame or a slice. When the macroblock that is currently encoded is the boundary macroblock, the adjacent blocks from some of the neighboring macroblocks are not available for prediction of motion vectors. Therefore, the efficiency of motion vector coding is badly affected.

7.2.2. Intra-frame coding in adjacent macroblocks

In frames coded using motion-compensated prediction (P-frames and B-frames), some macroblocks may be encoded in the intra-frame mode. It may happen in the case of fast and complex motion or in the case of reveal of covered regions in a video sequence. It has been reported [Dzi05] that in AVC/H.264 video codec 5% -90% macroblocks in P-frames are coded using intra-frame coding technique. In B-frames the number of intra-coded macroblocks is up to 60%. The number of intra-coded macroblocks depends mostly on the content of a video sequence and the target bitrate.

An intra-coded macroblock does not have motion vector assigned to any of its blocks, thus the prediction of motion vector in the neighboring macroblocks can not be performed efficiently, because of the lack of proper motion vectors for prediction.

7.2.3. Different types of prediction in adjacent blocks

In bidirectionally coded frames (B-frames), all types of motion-compensated prediction can be used: forward prediction, backward prediction and bidirectional

prediction. The macroblocks with various types of prediction may be mixed in a frame, depending on the decision of the encoder. This situation is depicted in Fig. 7.3, where neighboring macroblocks use various prediction modes.

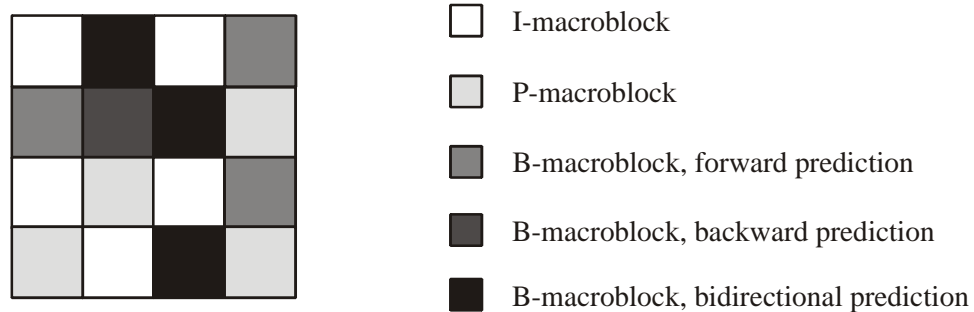


Fig. 7.3. Neighboring macroblocks use various prediction modes, fragment of hypothetical video frame.

When adjacent blocks are coded using prediction type that does not exploit a motion vector that refers to the specific temporal direction (the same as the current motion vector), the efficiency of motion vector prediction in the current block is reduced.

7.2.4. Different reference frames in adjacent blocks

In advanced video coders, many reference frames are used in motion-compensated prediction, as discussed in Section 3.2.3. Signal reconstruction is performed using a motion vector and an index of a reference frame. However, motion vectors that refer to different reference frames have different time scale. In fact, another dimension – temporal dimension – is added to motion vector.

Some attempts have been made regarding rescaling of the neighboring motion vectors that refer to different reference frames in order to match the current motion vector [Che02]. However the operation of motion vector rescaling is complex and the efficiency of the method is limited to some specific cases, thus this technique has not been adopted into AVC/H.264 algorithm.

When adjacent blocks use different reference frames than current motion vector, the efficiency of motion vector prediction decreases.

7.3. Inter-layer motion vector prediction

7.3.1. Original proposal of inter-layer motion vector prediction – Implicit Inter-Layer Prediction (IILP)

In cases when no good prediction is present in the vicinity of the current motion vector, interpolated motion vectors from the low-resolution motion field can be used instead. This inter-resolution technique can be also called inter-layer motion prediction, especially when it regards scalable video coding with layered approach.

Implicit Inter-Layer Prediction (IILP) – the original idea presented in this thesis, incorporates the technique of inter-layer motion prediction into the existing intra-layer motion prediction scheme. The original proposal of inter-layer motion prediction consists in the use of the interpolated motion vector from the base layer in the standard algorithm of motion vector prediction in the following cases:

- when current macroblock is a boundary macroblock,
- when adjacent macroblocks belongs to a different slice of macroblocks,
- when adjacent macroblocks use intra-frame coding,
- when adjacent blocks do not use the same type of motion-compensated prediction as the current block,
- when adjacent motion vectors refer to a different reference frame than the current motion vector.

In the above cases all or some of the motion vectors normally used for prediction are not available – they are “missing”. These “missing” motion vectors are replaced by the co-located motion vector from the low-resolution layer, as depicted in Fig. 7.4.

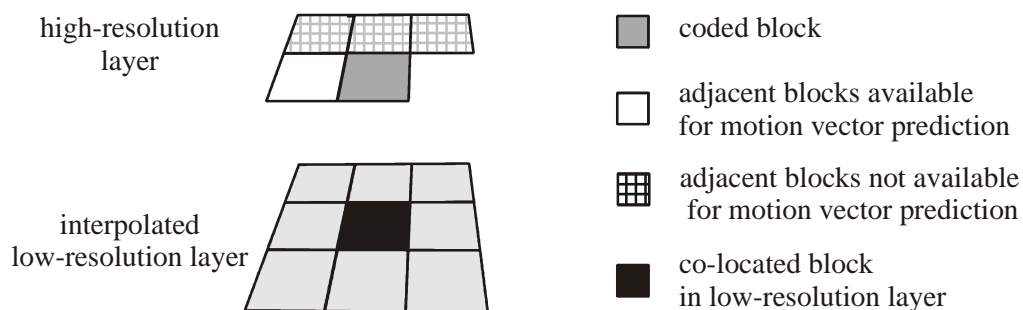


Fig. 7.4. Blocks used in inter-layer prediction of motion vectors.

Because the replacement of “missing” motion vectors is independently deduced by the encoder and the decoder, the author named this method Implicit Inter-Layer Prediction (IILP).

7.3.2. Possible approaches to inter-layer motion vector prediction

There is no guarantee that the co-located block in the low-resolution layer uses the same type of motion-compensated prediction and that it references to the same reference frame. Therefore, two algorithms of searching for the motion vector in the low-resolution layer were proposed. They are discussed in the following sections.

7.3.2.1. IILP using directly co-located block

In the following approach, the missing motion vector from the high-resolution video sequence is replaced with the motion vector of the directly co-located block from the low-resolution video sequence, regardless the prediction type of this block and regardless the reference frame that is used in motion-compensated prediction. The block used in inter-layer motion prediction is depicted in Fig. 7.5.

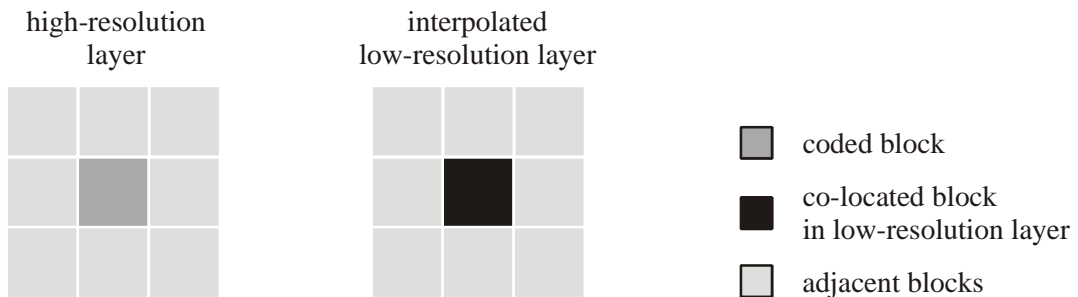


Fig. 7.5. IILP motion vector prediction using directly co-located block.

The advantage of such an approach is that directly co-located motion vector from low-resolution layer is usually most correlated with the currently coded motion vector. However in some situations, the motion vector from the low-resolution layer may not exist or may use different reference frame for motion-compensated prediction than the current motion vector.

7.3.2.2. IILP using the best block

The second proposal of the implementation of IILP is somewhat more sophisticated. It is applied also when the standard intra-layer prediction can not be performed due to the reasons mentioned in Section 7.3.1.

First, the algorithm checks whether the co-located motion vector from the base layer can be used for inter-layer motion prediction. When directly co-located block from the low-resolution video sequence is intra-frame coded, it uses a different type of motion-compensated prediction or references to a different reference frame than the current motion vector, then further searches are performed in order to find the best matching motion vector.

The algorithm searches the set of adjacent blocks in the low-resolution layer and tries to find a motion vector that utilizes the same type of inter-frame prediction as the currently coded motion vector. This is depicted in Fig. 7.6. Furthermore, the searched motion vector should refer to the same reference frame as the current motion vector. In such a case it is “the best” motion vector from the low-resolution layer.

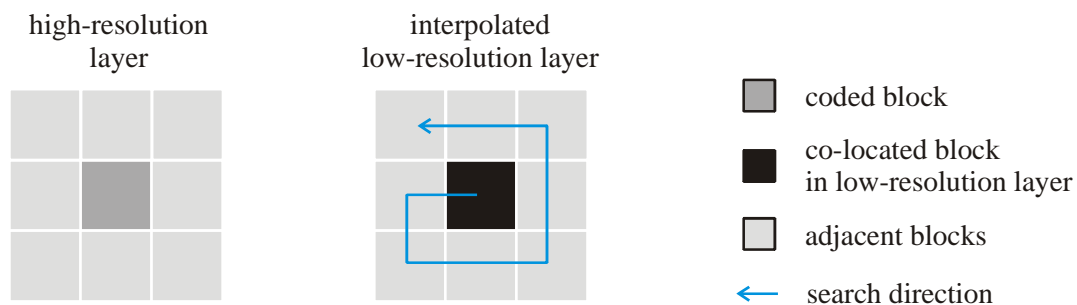


Fig. 7.6. IILP motion vector prediction using the best motion vector from the low-resolution layer. Blocks in low-resolution layer used in the process of searching of the best motion vector.

When the appropriate motion vector is found, the missing motion vector from the high-resolution video sequence is replaced by the vector from low-resolution video sequence. Standard motion vector prediction is then performed.

7.3.2.3. The impact of reduced accuracy of interpolated motion vectors from the base layer

Motion vectors from low-resolution layer are interpolated in order to be used for encoding of high-resolution video. The accuracy of interpolated motion vectors is lower than the accuracy of actual motion vectors estimated using high-resolution video.

Two variants of exploiting motion vectors from the base layer are proposed and experimentally tested in this thesis. The difference between the proposed variants is the number of times that the missing motion vector from the high-resolution layer is replaced by the motion vector from the base layer.

In the first scenario, the missing motion vector in high-resolution video sequence is replaced no more than once during a single prediction of motion vector. It prevents from decreasing of accuracy of the predicted motion vector.

In the second scenario, the missing motion vector in high-resolution video sequence can be replaced many times, as many, as many missing motion vectors appear in high-resolution layer during a single prediction. Such a solution assures that there is always a valid number of input vectors for median prediction, however it can result in decrease of prediction accuracy.

7.3.3. Proposed variants of IILP

A combination of the techniques described in Sections 7.3.2.1, 7.3.2.2 and 7.3.2.3 gives four variants of IILP method. Missing motion vectors from high-resolution layer are replaced in the following ways (see Tab 7.1):

- using the motion vectors from the base layer only once during single prediction,
- using the motion vectors from the base layer as many times as many missing motion vectors appear during single prediction,
- using the directly co-located motion vector from the low-resolution layer,
- searching for the best motion vector in the low-resolution layer.

Tab. 7.1. Techniques of the proposed multiresolution motion vectors prediction.

		vector from the base layer that is used for inter-layer prediction	
		directly co-located	the best chosen
how many times vector from the base layer is used	once	inter-layer 3	inter-layer 4
	many times	inter-layer 1	inter-layer 2

The possible results of motion vector prediction using various techniques of IILP are showed in Tab. 7.2. The marking of adjacent blocks is the same as in Fig. 4.4, additionally, block “E” is the block from the low-resolution layer (either the co-located one, either the best chosen, depending on technique).

Tab. 7.2. The result of inter-layer motion vector prediction, depending on availability of adjacent macroblocks and the reference frame used in inter-frame prediction.

adjacent block					technique #1, technique #2	technique #3, technique #4
A	B	C	D	E		
a	a	a			med(A,B,C)	med(A,B,C)
a	a	x	a		med(A,B,D)	med(A,B,D)
a	a	b,x	b,x	a,b,x	med(A,B,E)	med(A,B,E)
a	b,x	a		a,b,x	med(A,C,E)	med(A,C,E)
a	b,x	x	a	a,b,x	med(A,D,E)	med(A,D,E)
b,x	a	a		a,b,x	med(B,C,E)	med(B,C,E)
b,x	a	b,x	a	a,b,x	med(B,C,E)	med(B,C,E)
a	b,x	b,x	b,x	b,x	A	A
a	b,x	b,x	b,x	a	E	med(A,B,E)
b,x	a	b,x	b,x	b,x	B	B
b,x	a	b,x	b,x	a	E	med(A,B,E)
b,x	b,x	a		b,x	C	C
b,x	b,x	a		a	E	med(B,C,E)
b,x	b,x	x	a	b,x	D	D
b,x	b,x	x	a	a	E	med(B,D,E)
b,x	b,x	b,x	b,x	a	E	E
b,x	b,x	b,x	b,x	b,x	E	E

a – available (the same reference frame),

b – available (different reference frame or intra coded*),

x – unavailable*,

* when block is unavailable or intra coded, both motion vector components are equal to 0.

7.4. Modification of the scalable coder

The proposed algorithms of inter-layer motion vector prediction have been implemented in the scalable codec that has been briefly discussed in Section 2.4.2. The module of interpolation of motion vector field has been added, as depicted in Fig. 7.7.

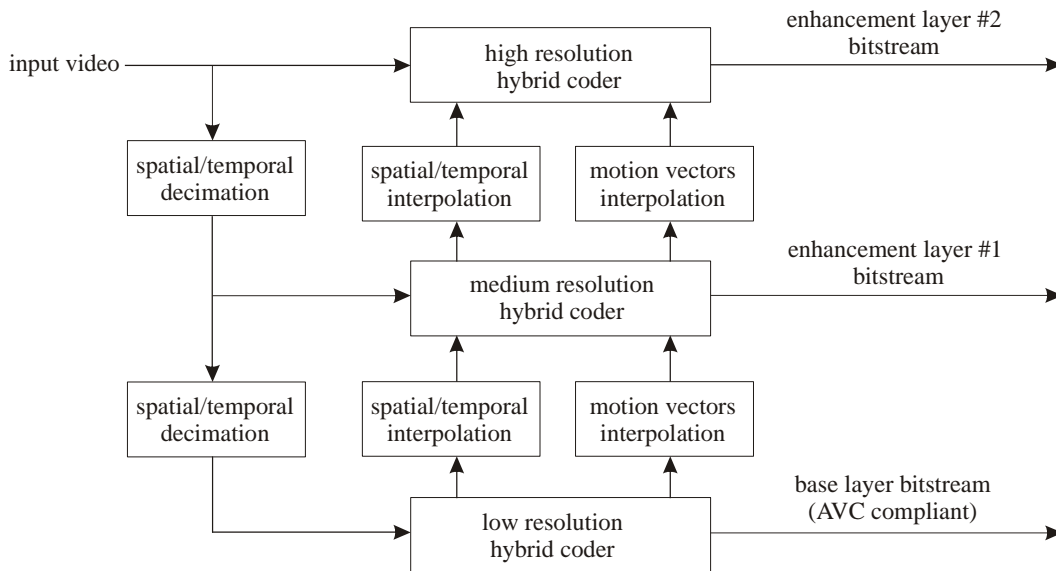


Fig. 7.7. Modification of scalable video hybrid coder with inter-layer motion vector prediction.

Motion vectors are estimated separately for the low-resolution and the high-resolution video sequence. Low-resolution motion vectors are then interpolated and stored. They are used during encoding of high-resolution layer as additional motion vectors for motion vector prediction.

Interpolation of the motion vectors is performed by simple upsampling of the motion vector field using a scheme of the nearest neighbor interpolation [Tek95]. Other techniques of motion vectors interpolation were also considered at the preliminary stage of the research. More sophisticated algorithms of motion vector interpolation include Boundary Matching technique [Lam93], Lagrange interpolation [Zhe03] and interpolation using the polynomial model [Zhe05]. However, these algorithms had been used mainly for recovery of lost motion vectors in the case of bitstream error. On the other hand, in applications of multiresolution motion estimation for video compression, usually simple nearest neighbor algorithm is applied in order to upsample given motion

vector field [Zaf93, Kri97a, Xu04, JVT06-02]. Thus finally, upsampling of motion vectors using the nearest neighbor interpolation has been applied in the author's verification model of the scalable codec.

7.5. IILP prediction of motion vectors – experimental results

The goal of the experiments is to check the efficiency of the IILP motion vector coding and to choose the most efficient solution among proposed variants of IILP. The experimental comparisons of four proposed techniques against independent coding of motion vectors have been performed. The tests have been performed for the CIF sequences *Bus*, *Foreman*, *Football* and *Mobile* and 4CIF sequences *City* and *Crew* using AVC-based scalable video codec, described in Section 2.4.2. The coder was set for producing two spatial layers, both temporally scalable using B-frames. The base layer represented a video sequence with a half of the spatial resolution of the video sequence that was represented in the enhancement layer.

For each video sequence the average motion vector prediction error have been measured as well as the overall bitrate for proposed techniques of IILP inter-layer motion vector prediction. Additionally, subjective evaluations of the quality of the encoded video sequences *Bus* and *Football* were performed using the SSMM technique described in Section 1.5.

The following parameters have been set in the configuration file of the encoder:

- period between I frames: 64,
- group of pictures: I-B-P-B-P,
- number of reference frames: 3,
- entropy coding: CABAC,
- range of motion estimation +/- 64 samples (full-pel units),
- range of bitrate: 200 kbps – 3000 kbps.

The results of the experiments are grouped in the following sections:

- bitrate and distortion,
- absolute value of motion vector prediction residuals,
- decrease of average absolute value of motion vector prediction residuals,
- subjective evaluation of the quality of the encoded video sequences residuals when IILP prediction was used.

7.5.1. Bitrate and distortion

In this section, the impact of inter-layer motion vectors prediction on bitrate and the distortion measure is presented. The achieved bitrate and the values of PSNR in CIF sequences for different quantization parameter Q_P are shown in Tab. 7.3-Tab. 7.6. The achieved bitrate and the values of PSNR in 4CIF sequences for different quantization parameter Q_P are shown in Tab. 7.7 and 7.8. The obtained efficiency proposed techniques can be compared using the rate-distortion curves that are depicted in Fig. 7.8 – Fig. 7.13.

Tab. 7.3. Bitrate (kbps) and PSNR (dB) in *Bus* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		$Q_P = 31$	$Q_P = 33$	$Q_P = 35$	$Q_P = 37$	$Q_P = 38$
base layer		209.1/31.72	149.2/30.13	107.5/28.66	80.3/27.42	66.9/26.69
enhancement layer, prediction type:	no inter-layer	684.2/32.49	486.1/30.93	352.0/29.46	264.0/28.19	222.1/27.45
	inter-layer 1	679.7/32.52	484.6/30.96	349.1/29.48	261.7/28.23	219.3/27.46
	inter-layer 2	677.7/32.5	482.7/30.94	349.7/29.47	262.6/28.22	218.2/27.45
	inter-layer 3	679.4/32.52	480.7/30.96	351.0/29.48	262.0/28.22	218.7/27.46
	inter-layer 4	679.2/32.5	482.6/30.96	348.8/29.48	260.3/28.19	220.1/27.47

Tab. 7.4. Bitrate (kbps) and PSNR (dB) in *Football* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		$Q_P = 33$	$Q_P = 35$	$Q_P = 37$	$Q_P = 39$	$Q_P = 41$
base layer		320.5/30.54	240.8/30.29	182.5/27.86	133.6/26.57	94.8/25.28
enhancement layer, prediction technique:	no inter-layer	735.5/31.73	561.3/30.29	439.7/29.11	328.8/27.83	249.5/26.64
	inter-layer 1	724.8/31.74	553.1/30.31	431.3/29.12	322.0/27.85	242.9/26.66
	inter-layer 2	726.7/31.75	552.4/30.3	432.3/29.13	322.5/27.85	244.3/26.65
	inter-layer 3	726.1/31.74	553.1/30.31	431.9/29.13	322.8/27.86	243.3/26.66
	inter-layer 4	726.6/31.74	553.2/30.29	431.6/29.12	322.1/27.85	243.0/26.64

Tab. 7.5. Bitrate (kbps) and PSNR (dB) in *Foreman* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		Q_P=25	Q_P=27	Q_P=29	Q_P=31	Q_P=33
base layer		171.7/38.43	124.5/36.92	93.7/35.68	72.5/34.42	55.0/33.15
enhancement layer, prediction type:	no inter-layer	517.6/38.59	363.4/37.24	259.4/36	195.4/34.86	144.7/33.68
	inter-layer 1	510.8/38.62	358.1/37.26	256.3/36.02	192.9/34.9	141.9/33.72
	inter-layer 2	512.6/38.61	356.7/37.25	255.9/36.02	193.2/34.9	141.0/33.71
	inter-layer 3	513.3/38.62	360.1/37.28	256.1/36.03	193.7/34.9	142.1/33.74
	inter-layer 4	512.7/38.62	358.4/37.27	255.0/36.01	193.7/34.9	141.7/33.72

Tab. 7.6. Bitrate (kbps) and PSNR (dB) in *Mobile* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		Q_P=32	Q_P=33	Q_P=35	Q_P=37	Q_P=39
base layer		152.3/29.92	125.5/29.22	89.6/27.76	67.5/26.99	49.6/25.18
enhancement layer, prediction type:	no inter-layer	671.9/30.65	539.3/29.91	352.7/28.35	243.9/26.99	169.2/25.52
	inter-layer 1	664.6/30.68	537.0/29.93	350.5/28.36	242.8/27.03	167.8/25.55
	inter-layer 2	664.1/30.67	538.2/29.93	349.6/28.38	242.7/27.02	167.2/25.54
	inter-layer 3	665.4/30.68	537.3/29.95	351.6/28.4	242.1/27.02	168.8/25.57
	inter-layer 4	665.3/30.69	538.9/29.95	349.6/28.38	242.1/27.03	167.3/25.55

Tab. 7.7. Bitrate (kbps) and PSNR (dB) in *City* (704×576, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		Q_P=31	Q_P=33	Q_P=35	Q_P=37	Q_P=38
base layer		241.5/32.86	168.3/31.48	119.1/30.18	86.9/29.07	73.5/28.43
enhancement layer, prediction type:	no inter-layer	927.5/33.52	624.5/32.2	440.9/30.9	322.7/29.74	271.1/29.07
	inter-layer 1	915.5/33.54	614.0/32.23	433.0/30.91	316.3/29.76	268.3/29.1
	inter-layer 2	915.3/33.53	613.9/32.22	434.0/30.91	316.7/29.75	268.3/29.09
	inter-layer 3	917.4/33.54	615.7/32.23	435.0/30.9	316.8/29.75	268.2/29.09
	inter-layer 4	914.8/33.55	613.6/32.22	433.9/30.91	318.4/29.75	268.8/29.09

Tab. 7.8. Bitrate (kbps) and PSNR (dB) in *Crew* (704×576, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in scalable AVC-based video codec.

(kbps)/(dB)		$Q_P=31$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=38$
base layer		415.7/35.01	293.1/33.76	210.7/32.53	154.1/31.49	128.0/30.91
enhancement layer, prediction type:	no inter-layer	1123.4/35.72	802.6/34.67	591.2/33.58	442.6/32.62	375.1/32.06
	inter-layer 1	1097.9/35.74	781.6/34.69	573.3/33.61	427.4/32.63	360.6/32.06
	inter-layer 2	1100.5/35.74	783.7/34.69	575.2/33.61	429.2/32.63	362.8/32.08
	inter-layer 3	1098.0/35.74	782.8/34.69	573.8/33.61	427.8/32.63	362.7/32.09
	inter-layer 4	1100.2/35.74	784.1/34.7	575.5/33.61	428.6/32.63	362.7/32.09

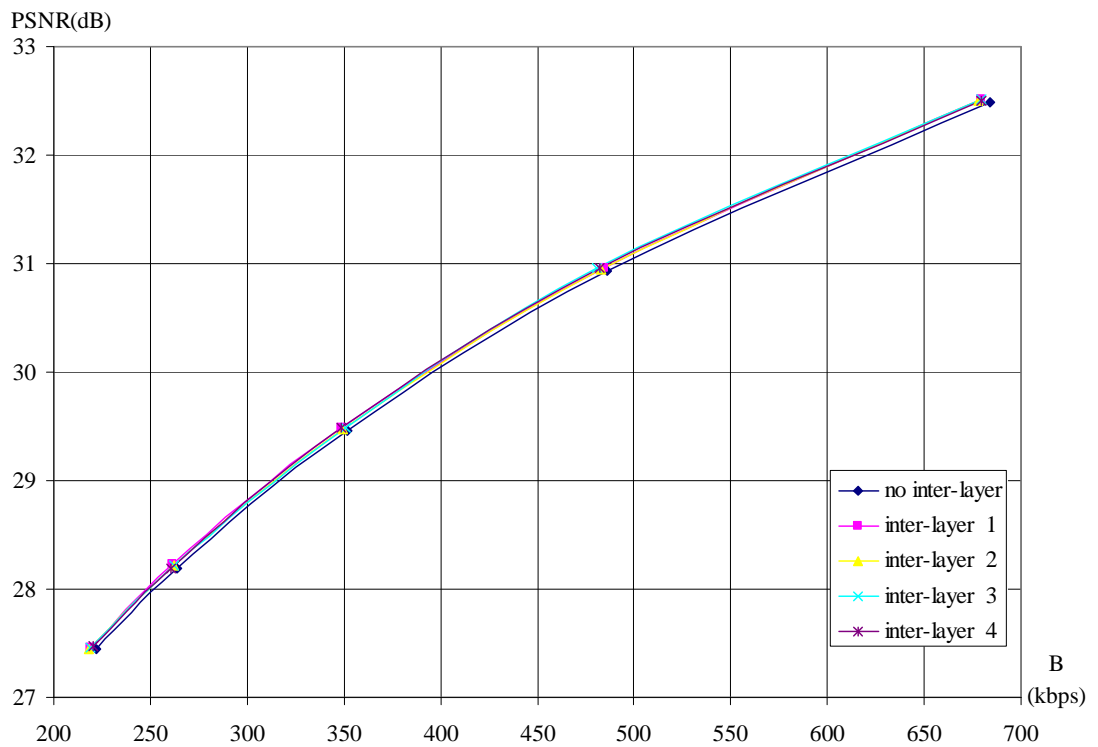


Fig. 7.8. R-D curves for various techniques of inter-layer prediction of motion vectors in *Bus* sequence (352×288); scalable AVC-based video codec.

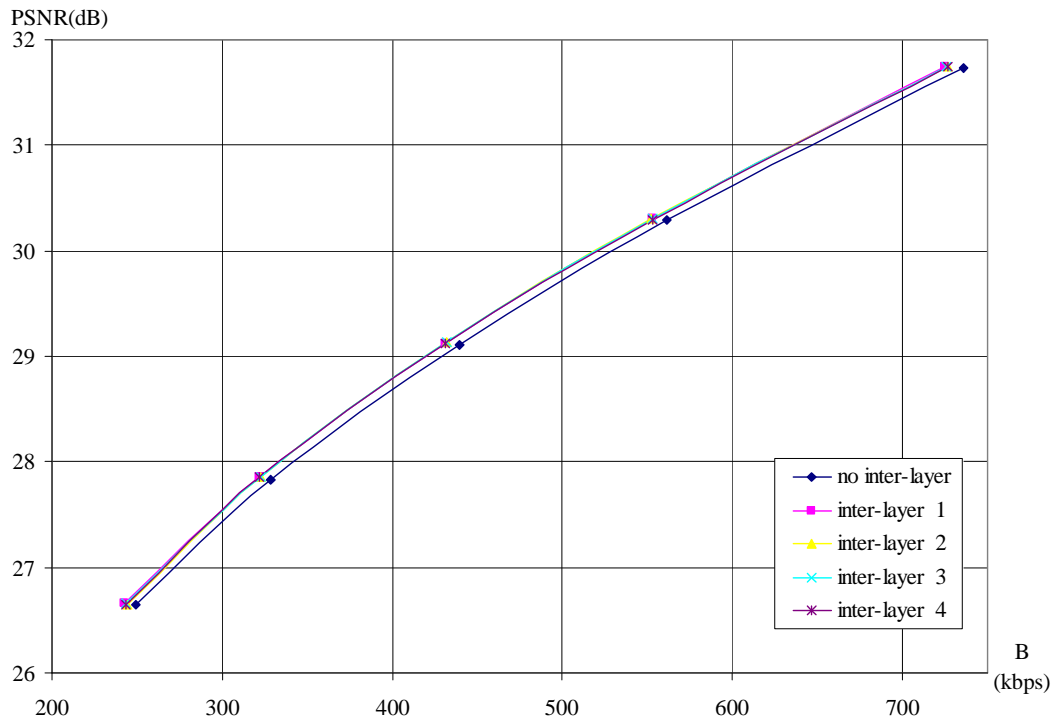


Fig. 7.9. R-D curves for various techniques of inter-layer prediction of motion vectors in *Football* sequence (352×288); scalable AVC-based video codec.

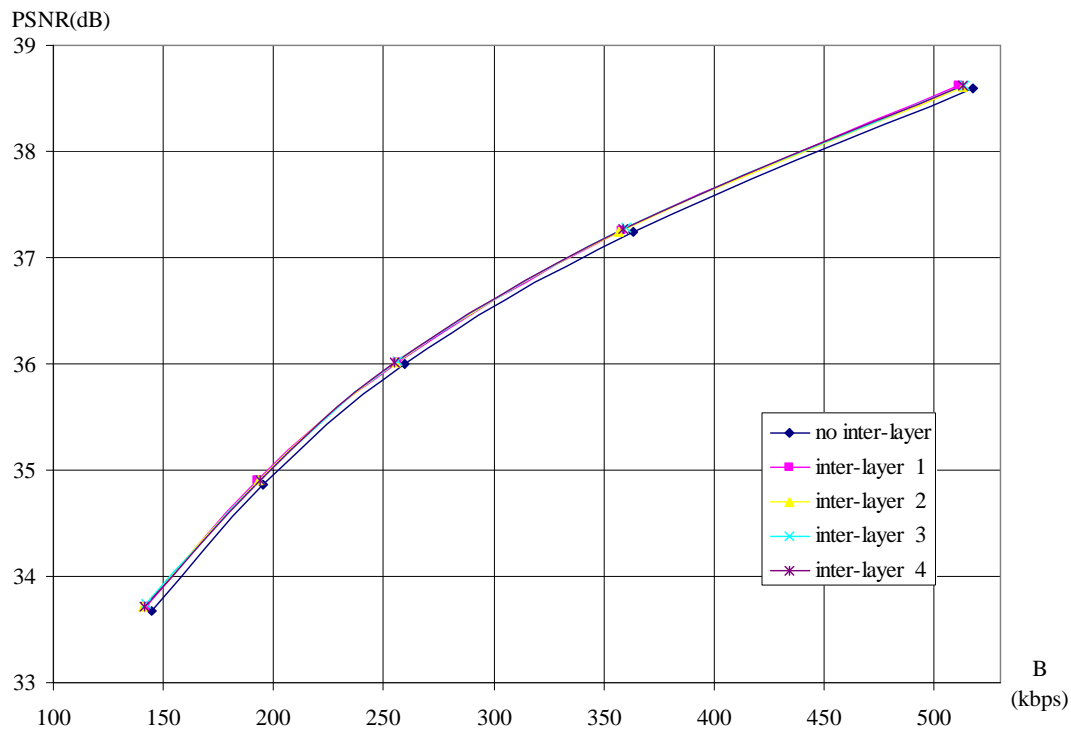


Fig. 7.10. R-D curves for various techniques of inter-layer prediction of motion vectors in *Foreman* sequence (352×288); scalable AVC-based video codec.

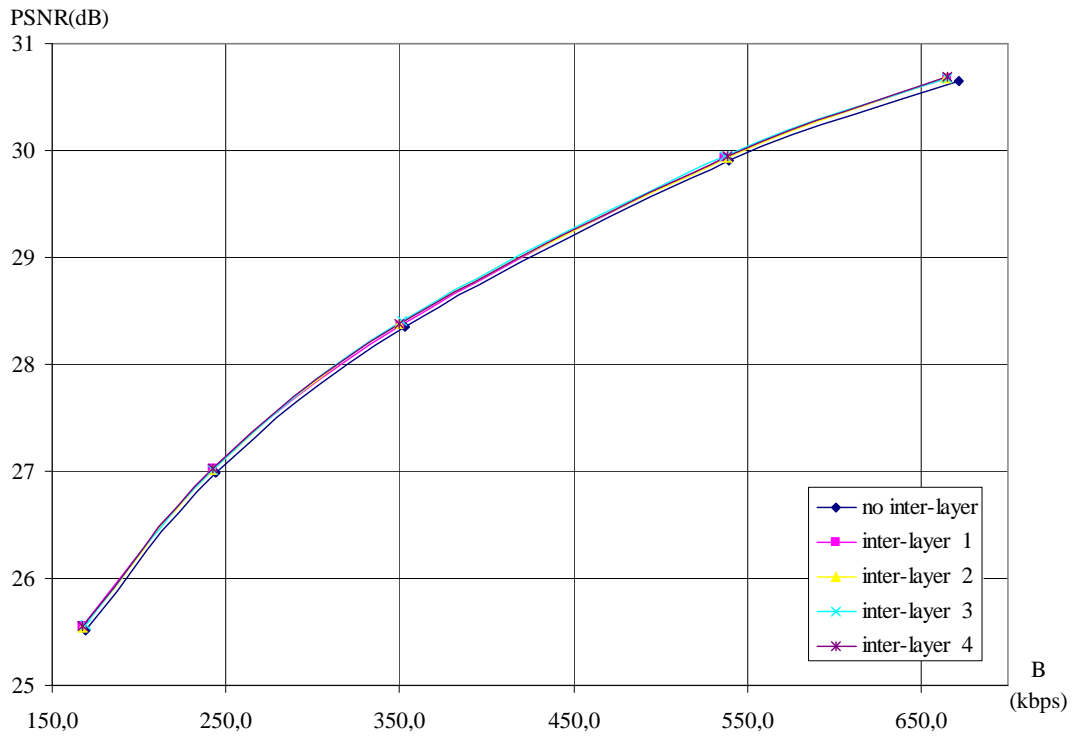


Fig. 7.11. R-D curves for various techniques of inter-layer prediction of motion vectors in *Mobile* sequence (352×288); scalable AVC-based video codec.

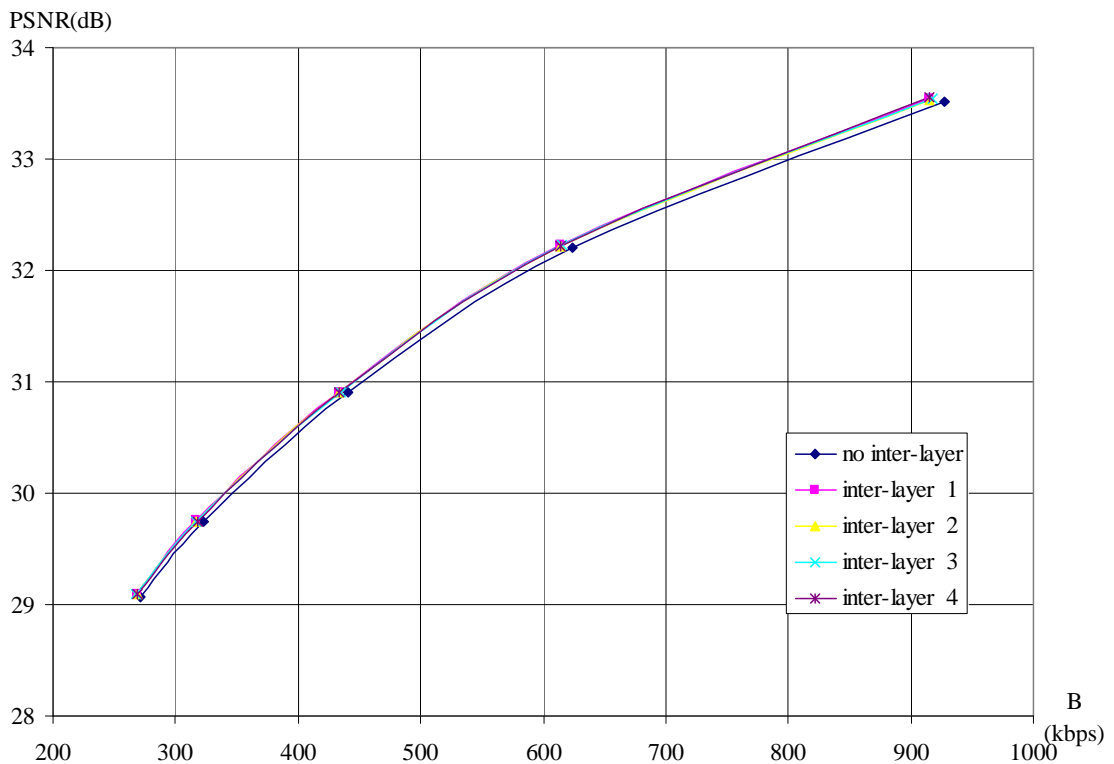


Fig. 7.12. R-D curves for various techniques of inter-layer prediction of motion vectors in *City* sequence (704×576); scalable AVC-based video codec.

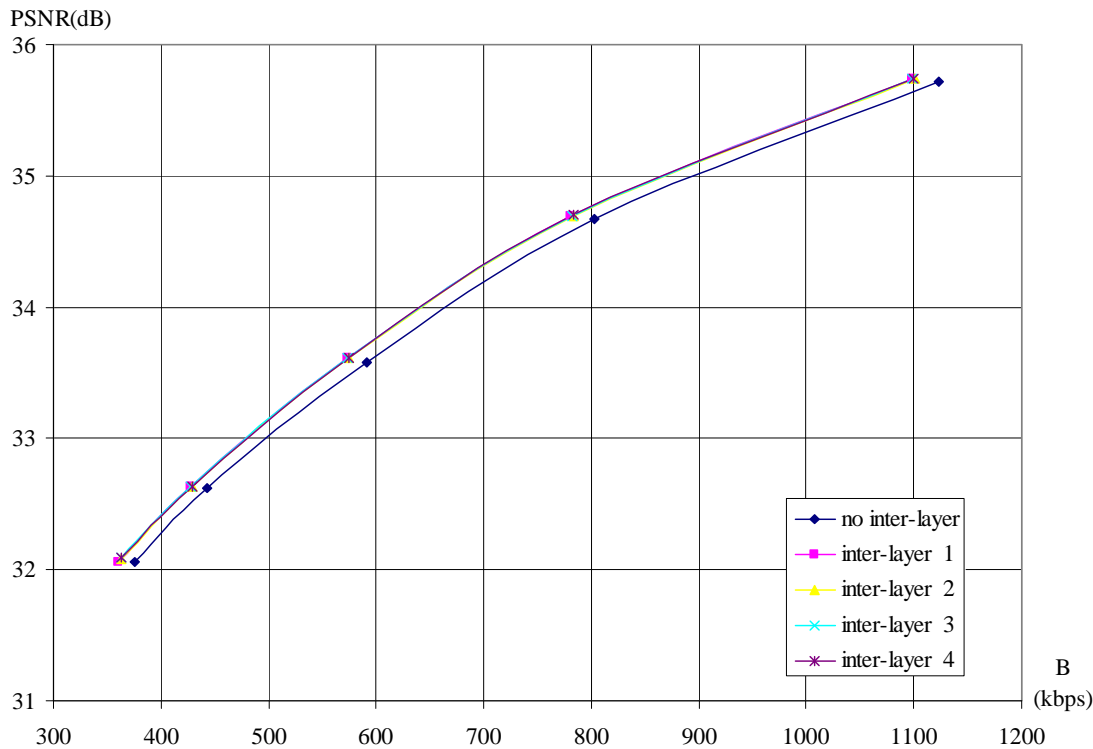


Fig. 7.13. R-D curves for various techniques of inter-layer prediction of motion vectors in *Crew* sequence (704×576); scalable AVC-based video codec.

When Implicit Inter-Layer Prediction was used for motion vector encoding, the bitrate reduction has been observed for all test sequences. The bitrate of enhancement layer decreased both in CIF sequences (Tab. 7.3-7.6) and in 4CIF sequences (Tab. 7.7 and 7.8).

All IILP variants outperform the technique of independent encoding of motion vectors (Fig. 7.8-7.13). Some differences between the results obtained using proposed variants of IILP can be observed, however, they are hardly visible in the figures containing R-D curves.

7.5.2. Average absolute values of motion vector prediction residuals

In this section there is presented the impact of inter-layer motion vectors prediction on average motion vector prediction residual. The average absolute value of motion vector prediction residuals in CIF sequences for various techniques of inter-layer motion vector prediction are depicted in Fig. 7.14 – Fig. 7.17. The average absolute values of motion vector prediction residuals in 4CIF sequences are depicted in Fig. 7.18 and 7.19.

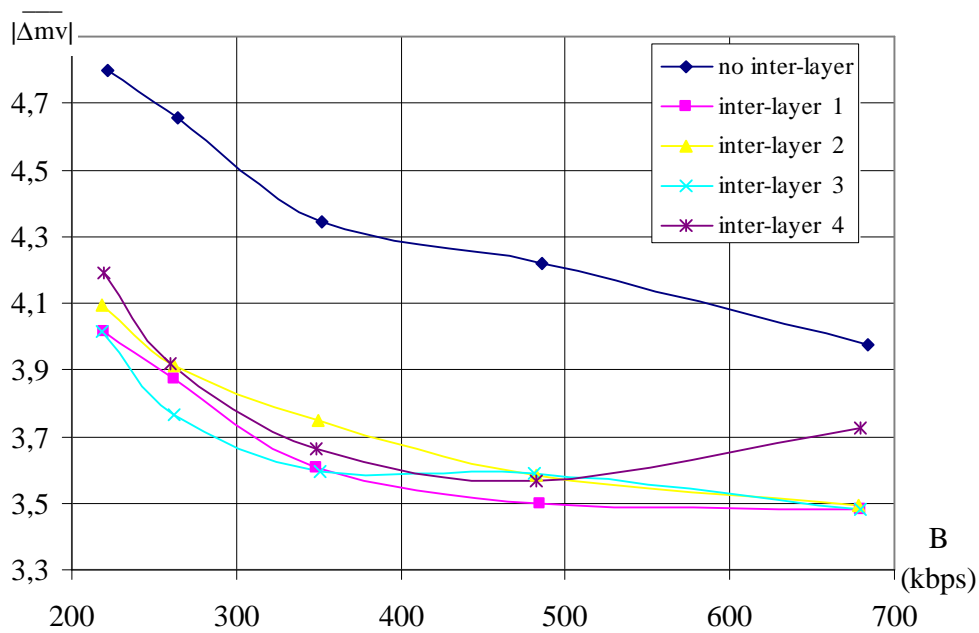


Fig. 7.14. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Bus* sequence (352×288); scalable AVC-based video codec.

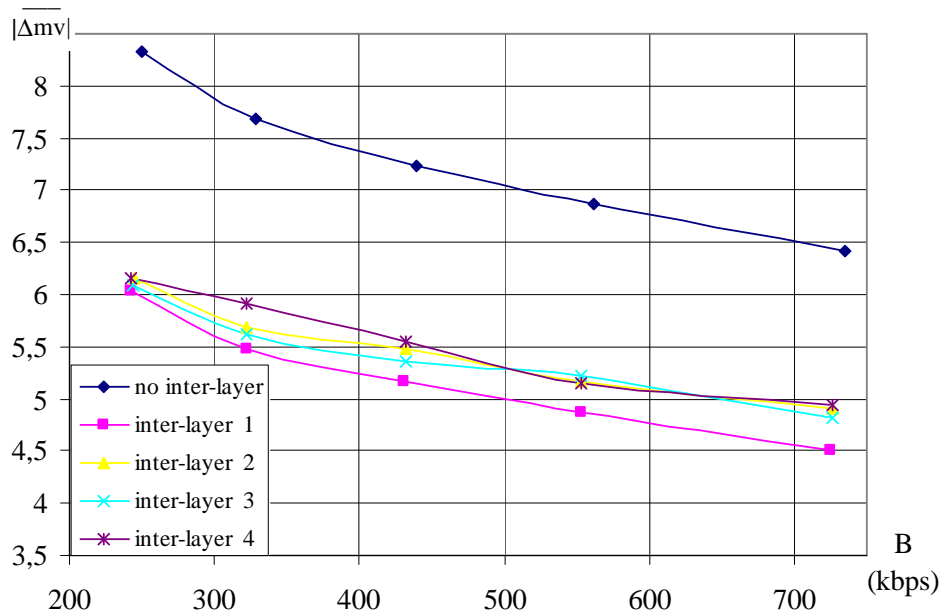


Fig. 7.15. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Football* sequence (352×288); scalable AVC-based video codec.

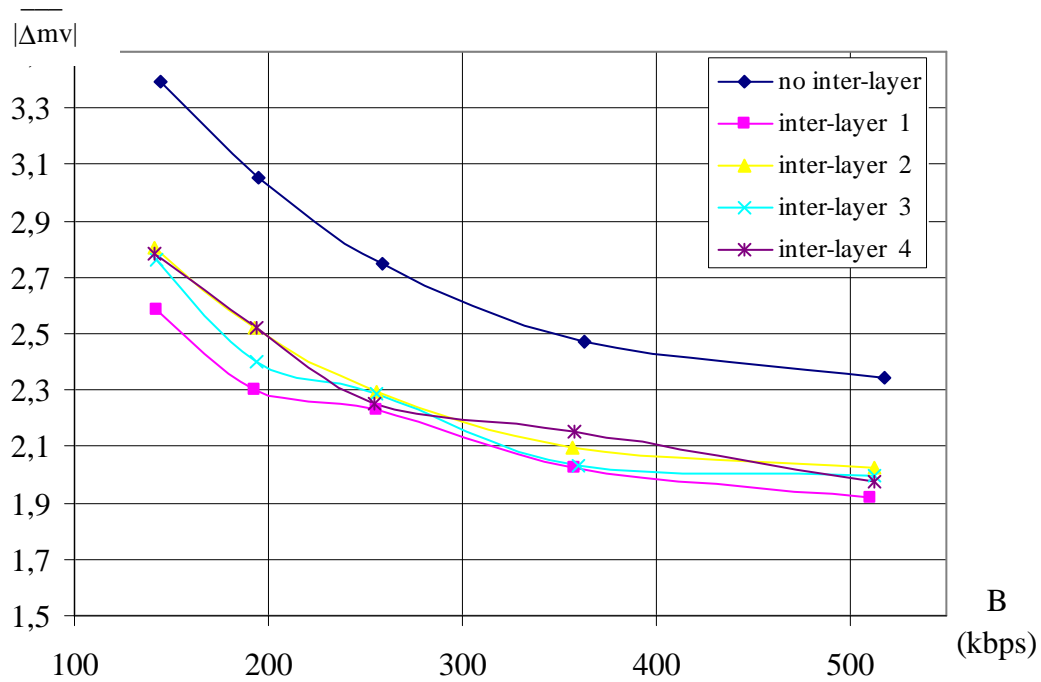


Fig. 7.16. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Foreman* sequence (352×288); scalable AVC-based video codec.

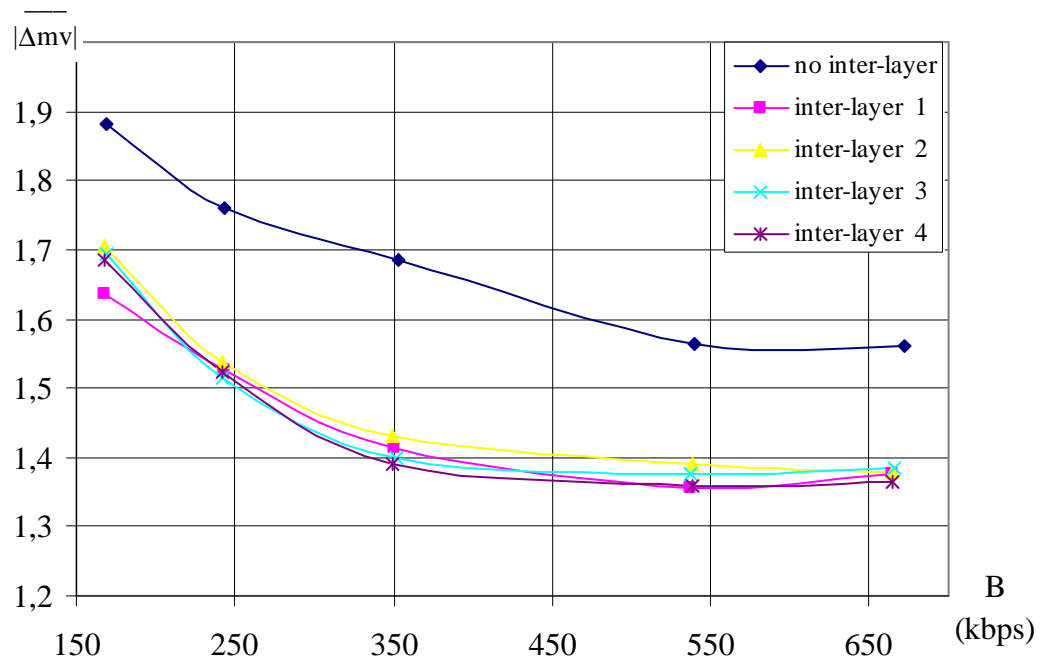


Fig. 7.17. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Mobile* sequence (352×288); scalable AVC-based video codec.

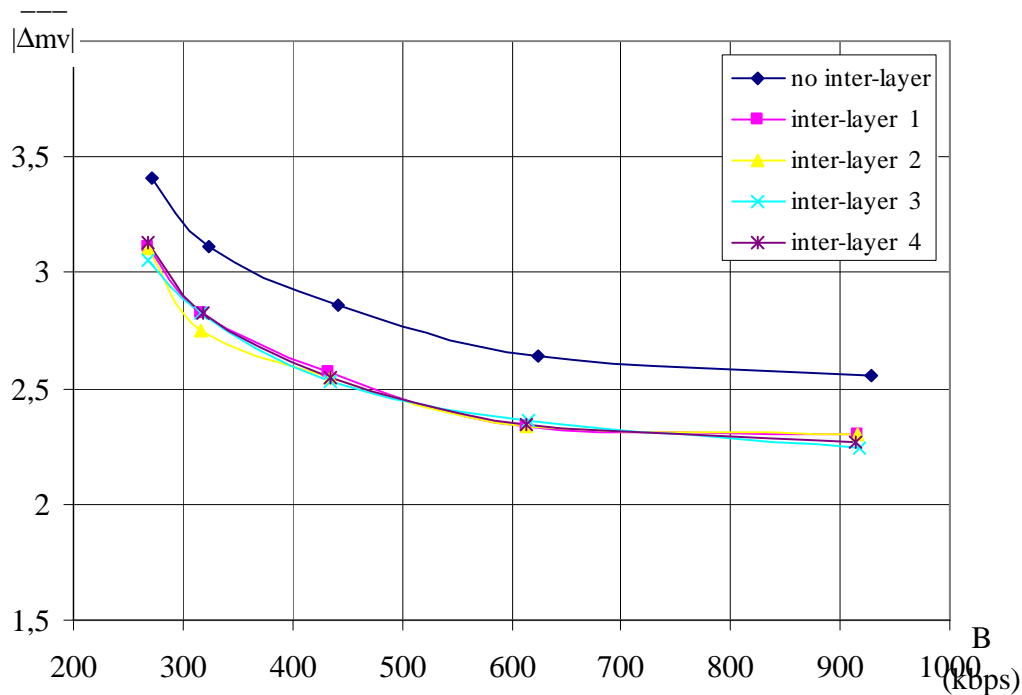


Fig. 7.18. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *City* sequence (704×576); scalable AVC-based video codec.

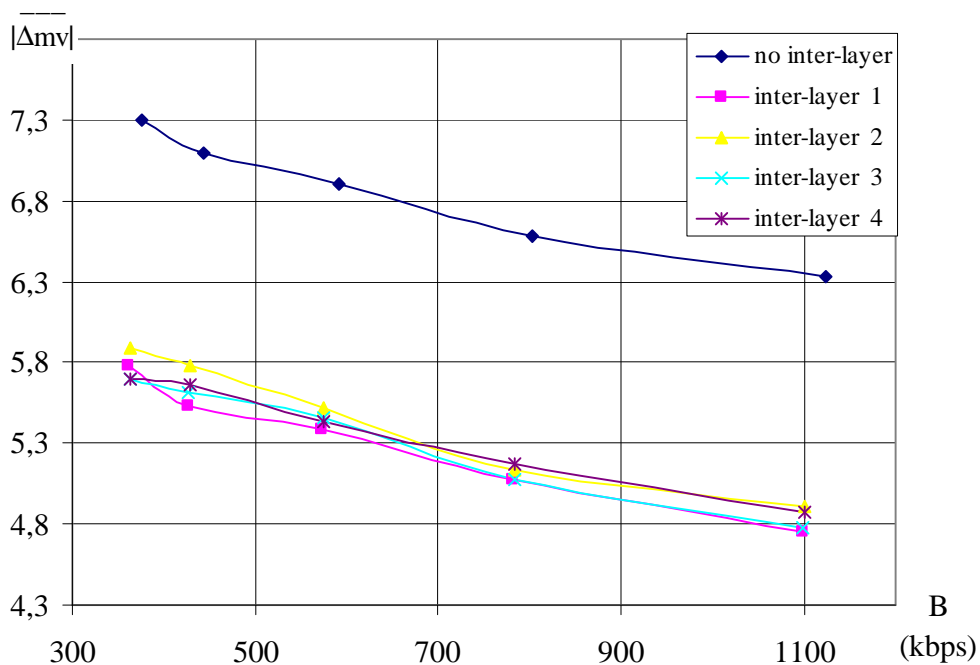


Fig. 7.19. Average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Crew* sequence (704×576); scalable AVC-based video codec.

In all cases the average absolute value of components of motion vector residual decreased when IILP technique was used. The obtained results are very similar for all proposed variants of IILP. Numerical results of achieved gain are presented in the following section.

7.5.3. The decrease of average absolute value of motion vector prediction residuals

In order to choose the best method of inter-layer motion vectors prediction, the decrease of motion vector prediction residual ($\Delta\text{res}_{\text{mv}}$) has been estimated, compared against the value obtained using standard, independent encoding of motion vectors in each layer. In Tab. 7.9- Tab 7.12 the decrease of average absolute values of components of motion vector residuals is given for different IILP techniques in CIF sequences. In Tab. 7.13 and 7.14 the same parameter is presented for 4CIF sequences. The average values of $\Delta\text{res}_{\text{mv}}$ for all values of quantization parameters Q_P have been calculated and are given in the following tables as well.

Tab. 7.9. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *Bus* sequence (352×288); scalable AVC-based video codec.

	$\Delta\text{res}_{\text{mv}}$ (%)					
	$Q_P=31$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=38$	average
inter-layer 1	12.52	17.09	17.02	16.84	16.31	15.96
inter-layer 2	12.18	15.07	13.78	15.96	14.61	14.32
inter-layer 3	12.49	14.95	17.22	19.13	16.24	16.01
inter-layer 4	6.34	15.42	15.67	15.83	12.60	13.17

Tab. 7.10. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *Football* sequence (352×288); scalable AVC-based video codec.

	$\Delta\text{res}_{\text{mv}}$ (%)					
	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=39$	$Q_P=41$	average
inter-layer 1	29.69	29.00	28.64	28.70	27.56	28.72
inter-layer 2	23.50	24.82	24.29	25.86	26.07	24.91
inter-layer 3	25.00	24.07	25.94	26.88	26.90	25.76
inter-layer 4	22.92	25.04	23.28	22.96	26.04	24.05

Tab. 7.11. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *Foreman* sequence (352×288); scalable AVC-based video codec.

	$\Delta\text{res}_{\text{mv}}$ (%)					
	$Q_P=25$	$Q_P=27$	$Q_P=29$	$Q_P=31$	$Q_P=33$	average
inter-layer 1	18.12	17.97	18.81	24.51	23.79	20.64
inter-layer 2	13.80	14.99	16.51	17.23	17.36	15.98
inter-layer 3	15.03	17.57	16.84	21.36	18.53	17.87
inter-layer 4	15.81	12.89	18.00	17.44	17.93	16.41

Tab. 7.12. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *Mobile* sequence (352×288); scalable AVC-based video codec.

	$\Delta\text{res}_{\text{mv}}$ (%)					
	$Q_P=32$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=39$	average
inter-layer 1	11.95	13.34	16.19	13.24	13.04	13.55
inter-layer 2	11.69	11.07	15.15	12.61	9.27	11.96
inter-layer 3	11.43	11.91	16.98	13.97	9.89	12.84
inter-layer 4	12.65	13.05	17.41	13.47	10.45	13.41

Tab. 7.13. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *City* sequence (704×576); scalable AVC-based video codec.

	Δres_{mv} (%)					
	$Q_P=31$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=38$	average
inter-layer 1	10,10	11,76	10,07	9,31	8,80	10,01
inter-layer 2	9,93	11,78	10,97	11,57	8,99	10,65
inter-layer 3	12,49	10,65	11,47	9,29	10,53	10,88
inter-layer 4	11,24	11,43	11,00	9,27	8,16	10,22

Tab. 7.14. Decrease of average absolute value of components of motion vector residual for various techniques of inter-layer prediction of motion vectors in *Crew* sequence (704×576); scalable AVC-based video codec.

	Δres_{mv} (%)					
	$Q_P=31$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=38$	average
inter-layer 1	24.91	22.88	21.94	22.11	20.78	22.52
inter-layer 2	22.35	21.95	20.09	18.62	19.36	20.47
inter-layer 3	24.43	22.85	20.92	20.99	22.02	22.24
inter-layer 4	22.92	21.41	21.19	20.33	22.02	21.57

The maximum decrease of average motion vector residual Δres_{mv} has been observed in video sequences with rather rough motion vector fields like *Football* and *Crew* (Tab 7.10 and 7.14). Surprisingly, in these sequences, the inter-layer correlations researched in Chapter 5 were relatively lower as compared with other video sequences (e.g. compare Fig. 5.4, 5.10, 5.8 and 5.14). On the other hand, since spatial correlation in these rough motion vector fields is lower as well, the standard prediction of the motion vectors gives poor results (compare with the results obtained in Chapter 4). Therefore, the additional inter-layer prediction improves the overall efficiency of the codec.

7.5.4. Subjective evaluation of quality

In order to verify the results obtained in Section 7.5.1, a subjective comparison of the efficiency of the proposed IILP techniques was performed by the author. The Single Stimulus Multimedia (SSMM) method [Bar04] of subjective assessment of quality, described in Section 1.5, was chosen. Because of economical reasons, only two video sequences were evaluated (*Bus* and *Football*). Each sequence was encoded with two various values of quantization parameter Q_P . The results of the subjective assessments are depicted in Fig. 7.20 and Fig. 7.21.

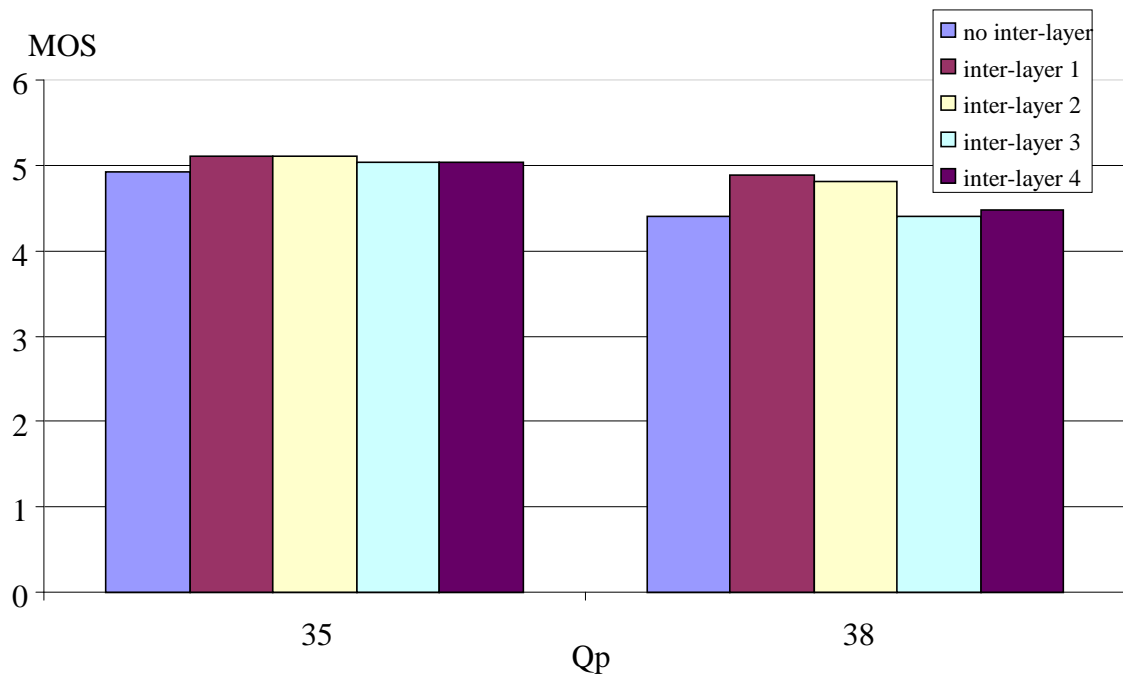


Fig. 7.20. The results of subjective assessment of the quality of *Bus* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality tests performed for *Bus* video sequence encoded with the value of quantization parameter $Q_P=35$, the highest value of Mean Opinion Score (MOS=5.1) was achieved for a scalable codec using the IILP motion vector prediction variants 1 and 2. Codecs with any variant of inter-layer motion vector prediction achieved slightly higher MOS (5.0-5.1) than a scalable video codec not using the algorithm of inter-layer motion vector prediction (MOS=4.9). In the tests performed for *Bus* video sequence encoded with the value of quantization parameter $Q_P=38$, the highest value of Mean Opinion Score (MOS=4.9) was achieved for the scalable codec

using the IILP motion vector prediction variant 1. The codec that did not use IILP technique and the codec using IILP variant 3 achieved the lowest values of MOS equal to 4.4.

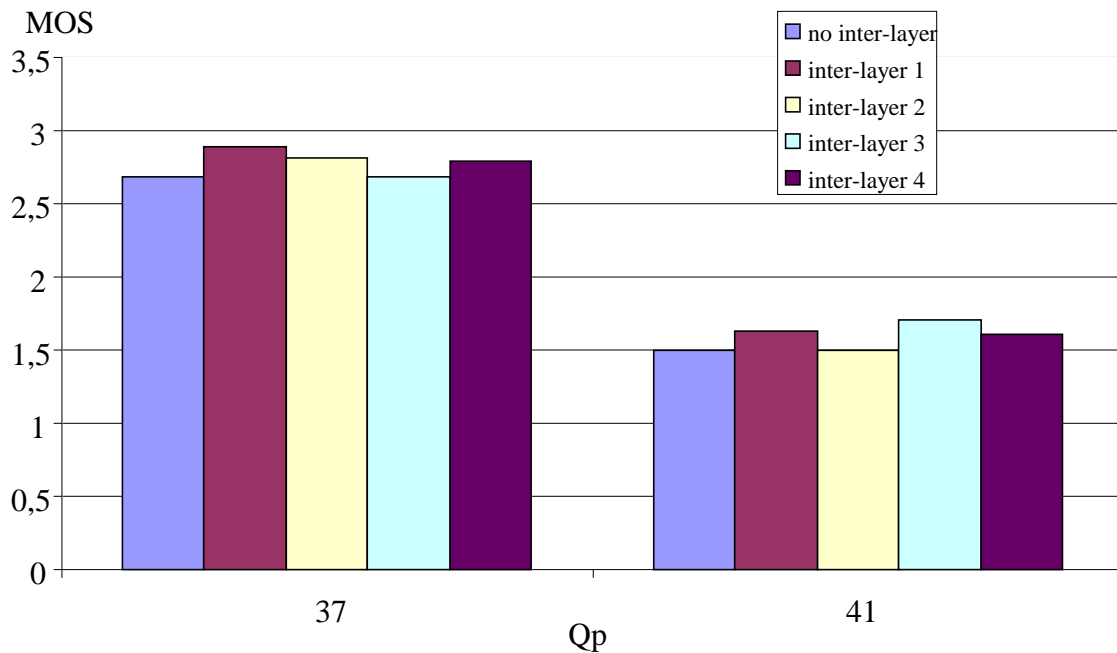


Fig. 7.21. The results of subjective assessment of the quality of *Football* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality evaluation performed for *Football* video sequence encoded with quantization parameter $Q_p=37$, the highest value of MOS (MOS=2.9) was achieved by the scalable video codec using IILP technique variant 1. The scalable codec not using inter-layer motion vector prediction achieved the value of MOS equal to 2.7. The other video codecs that exploited IILP algorithms achieved the values of MOS equal to 2.8, 2.7 and 2.8 (variants 2, 3 and 4, respectively). In the tests performed for a video sequence encoded with the value of quantization parameter $Q_p=41$, the highest value of MOS (MOS=1.7) was achieved for a scalable codec using the IILP motion vector prediction variant 3. The codec that did not use IILP technique and the codec using IILP variant 2 achieved the lowest values of MOS equal to 1.5, while the codecs using IILP variant 1 and 4 achieved the value of MOS equal to 1.6.

7.6. IILP prediction of motion vectors – conclusions

In all experiments, inter-layer techniques of motion vectors prediction outperformed standard, intra-layer only motion vector prediction. At the same bitrate, the quality of video sequence increased in all experimental cases, as presented with R-D curves in Fig. 7.8 – Fig. 7.13. The increase of overall PSNR is up to 0.2 dB (*Football* sequence, Fig. 7.9).

Unexpectedly, the maximum PSNR improvement has been reached in sequences with rough motion vector field, i.e. sequences containing fast and complex motion (*Football*) or sequences containing rapid global illumination changes (*Crew*). In such sequences, inter-layer similarities between motion vector fields from high-resolution layer and low-resolution layer are lower, as it has been proven in Chapter 5. However, obviously also intra-layer correlation of the neighboring motion vectors is lower in the video sequences with rough motion vector field. Since median-based spatial prediction is less efficient, as it has been proven in Chapter 4, the additional motion vectors from the base layer improve the overall efficiency of the motion vector compression.

The proposed techniques of extended inter-layer motion vector prediction have significantly reduced motion vectors prediction residuals. The decrease of average absolute value of components of motion vector residual varies from 10% (*City*) to 28.7% (*Football*).

On the other hand, the differences between proposed variants of IILP prediction are minor and the modifications have a negligible impact on achieved bitrates and the values of PSNR. However, in almost all cases, the best prediction of motion vector component gave the “inter-layer 1” technique. This is confirmed by the results of subjective quality evaluation (the highest MOS in 3 of 4 cases, Fig. 7.20 and 7.21) and the highest value of $\Delta_{res,mv}$ parameter (Tab. 7.9-7.14) achieved for this variant. The “inter-layer 1” technique exploits a motion vector of the directly co-located block in the low-resolution layer as many times as many “missing” motion vectors appear in the neighborhood of the currently coded motion vector from the high-resolution layer.

The complexity of proposed algorithm is relatively small. The only additional operations at encoder’s and decoder’s site is storing of low-resolution motion vector field and simple rescaling of the motion vector field. These operations have a negligible impact on overall complexity of the codec.

The presented solution gives a noticeable gain in the compression efficiency of the scalable codec, as compared to independent representation of motion vectors in each layer. The improvement is achieved without extra costs neither in complexity, nor in major changes in other modules of a video codec.

As proved herein, the presented solution decreases the average motion vector prediction residual and improves the overall compression efficiency. In the following sections, another approach to inter-layer motion vector prediction is presented that was developed by MPEG after the author's proposal. These two approaches: the author's IILP solution and the one used in a scalable codec developed by MPEG are compared with each other in Section 7.8.

7.7. Inter-layer prediction with explicit signaling of prediction mode – MPEG approach

7.7.1. MPEG proposal of advanced scalable video coding - SVC

In 2003, MPEG committee started its activity in order to establish a new standard of Scalable Video Coding (SVC) [MP-03-25, MP03-93]. 21 teams from all around the world answered for a Call for Proposal with their proposals of scalable video codecs. Subjective tests were conducted in order to choose the most efficient technique of scalable video representation.

Very good subjective results (see Fig. 6.1 in Chapter 6) achieved the scalable codec provided by Heinrich Hertz Institute (HHI) called “Scalable Extension of AVC/H.264” [MP04-69, Bar04]. The HHI original approach used an open-loop subband coding in order to exploit temporal dependencies between pictures. Temporal scalability was achieved with Motion-Compensated Temporal Filtering (MCTF) and spatial scalability was achieved with layered multiresolution coding [Sch04]. The codec used most of the AVC/H.264 tools and the AVC/H.264 bitstream syntax in order to represent multiresolution video.

In the original version, the codec described in the document “Scalable Extension of AVC/H.264” did not support inter-layer motion vectors prediction. However, the algorithm was later modified during MPEG meetings and many tools were adopted over the time. Among other things – there has been added the technique of the inter-resolution motion vector prediction.

In 2006 works on SVC are about to be finished. Some of the originally contributed features were removed from the standard draft (for example MCTF). Finally, Scalable Video Coding has become an annex to the AVC/H.264 recommendation.

7.7.2. Inter-layer motion vector prediction in SVC

In the draft of SVC algorithm, two new macroblock prediction modes are specified, which allows for inter-layer inheritance of motion information. Inter-layer prediction of motion vector is signaled by flags which are present in a bitstream before a syntax element describing regular macroblock mode.

Syntax elements `base_mode_flag` and `base_mode_refinement_flag` allow for inheriting from the low-resolution layer the following data:

- partitioning of a macroblock,
- the indices of a reference frame,
- motion vectors.

Possible modes of inter-layer prediction of motion vectors are described in Tab. 7.15.

Tab. 7.15. Prediction of the motion vectors in SVC codec..

syntax element			motion vectors encoding
base_mode_flag	base_mode_refinement_flag	motion_prediction_flag_IX[i]	
“true”	“false”	-	direct re-use of motion vectors from low-resolution layer in whole macroblock
“true”	“true”	-	re-use of motion vectors from low-resolution layer + ¼-pel refinement in whole macroblock
“false”	-	“false”	standard AVC/H.264 predictive encoding of motion vectors
“false”	-	“true”	direct re-use of motion vectors from low-resolution layer in <i>i</i> -th partition of a macroblock

The combination of the values of the syntax elements specified in the table signals the mode of inter-layer motion vectors prediction. Therefore, inter-layer prediction mode is explicitly signaled in the bitstream.

In the basic inter-layer mode (`base_mode_flag` = “true”, `base_mode_refinement_flag` = “false”), no further motion data are sent for the current macroblock. Therefore, motion

vectors from the low-resolution layer are interpolated and directly used in motion-compensated prediction in high-resolution layer.

Moreover, in $\frac{1}{4}$ -pel refinement mode (`base_mode_flag = "true"`, `base_mode_refinement_flag = "true"`), motion vectors from the low-resolution layer are further refined using $\frac{1}{4}$ -pel correction values, which are encoded in a bitstream.

Additionally, another bitstream flag (`motion_prediction_flag_1X`) allows for switching on inter-layer prediction of the motion vectors for selected partitions of the regularly encoded macroblock in high-resolution layer.

7.8. Comparison of inter-layer motion prediction techniques in SVC codec

Explicit signaling of the inter-layer motion prediction in SVC codec is an alternative technique to the author's IILP algorithm presented in Section 7.3. The main difference between presented methods is a way of indication of the inter-layer prediction mode.

In the author's contribution, inter-layer motion vector prediction is not signaled explicitly in a bitstream, but it is rather deducted from the context of a coded motion vector. In a method incorporated into SVC codec, all inter-layer motion vector prediction modes are explicitly signaled in a bitstream. Such an approach requires a lot of extra information to be encoded, but allows for flexible control of the motion vector prediction mode.

In this section, a comparison of both algorithms is performed in terms of coding efficiency, motion vector prediction efficiency and complexity of the algorithms.

7.8.1. Comparison of inter-layer motion prediction techniques in SVC codec – experimental results

The results of comparison of IILP technique against the technique applied in scalable codec developed by MPEG are presented in the following sections. Comparisons have been performed for the CIF sequences *Bus*, *Foreman*, *Football* and *Mobile* using modified SVC scalable codec (version 4.2). The coder was set for producing two spatial layers, both temporally scalable using B-frames. The base layer represented a video

sequence with a half of the spatial resolution of the video sequence that was represented in the enhancement layer.

For each sequence the average motion vector prediction error has been measured as well as the overall bitrate for various techniques of inter-layer motion vector prediction. The subjective evaluations of the quality of the encoded video sequences *Bus* and *Football* were performed using the SSMM technique described in Section 1.5. Time-complexity has been measured for each of the inter-layer motion vector prediction method.

The following parameters have been set in the configuration file of the encoder:

- period between I frames: 96,
- group of pictures: I-B-P-B-P,
- number of reference frames: 3,
- entropy coding: CABAC,
- range of motion estimation +/- 96 samples (full-pel units),
- adaptive inter-layer prediction,
- range of low-resolution bitrate: 70 kbps – 280 kbps.
- range of high-resolution bitrate: 200 kbps – 1000 kbps.

In all diagrams, the bitrate of high-resolution layer is denoted as B_E , while the overall bitrate is denoted as B . Variants of modified SVC codec are denoted using the following symbols:

- **SVC** denotes the standard technique of motion vector coding used in SVC codec (see Section 7.7.2),
- **IILP** denotes the algorithm of motion vector prediction developed by the author of this dissertation (see Section 7.3.1),
- **SVC+IILP** denotes joint usage of both techniques (standard technique from SVC and IILP technique),
- **no inter-layer** denotes that no inter-layer motion vector prediction was performed.

7.8.1.1. Bitrate and distortion

In this section, comparison of bitrate and distortion is presented for various techniques of inter-layer prediction of motion vectors. Obtained bitrate and the values of PSNR in CIF sequences for different quantization parameter Q_P are shown in Tab. 7.16

– Tab. 7.19. Rate-distortion curves for all test sequences are depicted in Fig. 7.22 – Fig. 7.25.

Tab. 7.16. Bitrate (kbps) and PSNR (dB) in *Bus* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.

(kbps)/(dB)		Q_P=31	Q_P=33	Q_P=35	Q_P=37	Q_P=38
base layer		239.5/33.48	184.5/31.93	142.3/30.50	107.3/29.02	95.5/28.43
enhancement layer, prediction type:	SVC + IILP	765.4/33.80	580.7/32.35	443.6/31.00	330.6/29.57	293.9/28.99
	IILP	783.5/33.82	598.3/32.39	460.6/31.05	345.4/29.63	309.0/29.07
	SVC	764.4/33.81	578.6/32.34	441.5/31.00	327.9/29.56	291.5/29.00
	no inter-layer	792.8/33.82	607.1/32.39	467.9/31.06	352.9/29.64	315.6/29.09

Tab. 7.17. Bitrate (kbps) and PSNR (dB) in *Football* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.

(kbps)/(dB)		Q_P=33	Q_P=35	Q_P=37	Q_P=39	Q_P=41
base layer		279.6/33.02	219.4/31.74	167.0/30.51	129.3/29.26	99.3/28.12
enhancement layer, prediction type:	SVC + IILP	605.6/33.86	477.7/32.64	362.4/31.37	275.9/30.18	213.3/29.15
	IILP	637.0/33.95	505.1/32.77	387.2/31.53	300.9/30.40	237.6/29.38
	SVC	603.7/33.86	475.9/32.64	361.7/31.36	275.5/30.21	213.3/29.14
	no inter-layer	652.4/33.97	518.5/32.79	400.0/31.56	312.2/30.46	248.0/29.44

Tab. 7.18. Bitrate (kbps) and PSNR (dB) in *Foreman* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.

(kbps)/(dB)		Q_P=25	Q_P=27	Q_P=29	Q_P=31	Q_P=33
base layer		211.4/39.84	163.4/38.43	127.8/37.10	99.0/35.65	78.4/34.28
enhancement layer, prediction type:	SVC + IILP	618.4/39.73	444.4/38.53	334.2/37.37	248.2/36.08	188.3/34.87
	IILP	644.6/39.78	467.4/38.58	353.7/37.44	267.2/36.19	203.4/34.99
	SVC	614.3/39.74	441.5/38.53	330.4/37.39	245.4/36.09	187.2/34.88
	no inter-layer	645.3/39.78	469.4/38.59	354.6/37.44	268.4/36.19	205.8/34.99

Tab. 7.19. Bitrate (kbps) and PSNR (dB) in *Mobile* (352×288, IBPBP) sequence using various techniques of inter-layer prediction of motion vector in SVC video codec.

(kbps)/(dB)		Q _p =32	Q _p =33	Q _p =35	Q _p =37	Q _p =39
base layer		202.4/31.68	169.9/30.71	127.2/29.13	91.7/27.57	69.6/26.11
enhancement layer, prediction type:	SVC + IILP	963.6/32.14	792.6/31.26	560.8/29.81	375.8/28.27	264.2/26.84
	IILP	967.9/32.14	795.2/31.25	561.8/29.82	376.9/28.29	264.3/26.87
	SVC	961.6/32.15	790.6/31.26	559.3/29.82	373.5/28.27	261.8/26.85
	no inter-layer	970.1/32.14	797.9/31.25	563.4/29.81	376.8/28.29	264.4/26.87

In all video sequences the bitrates achieved when inter-layer prediction was used (cases denoted as SVC, IILP and SVC+IILP) were lower than the bitrate achieved for the codec with no inter-layer prediction of motion vector.

Since in some cases the obtained values of PSNR were different for different methods of motion vector prediction, the R-D curves presented in the following figures allow for comparison of the efficiency of given techniques.

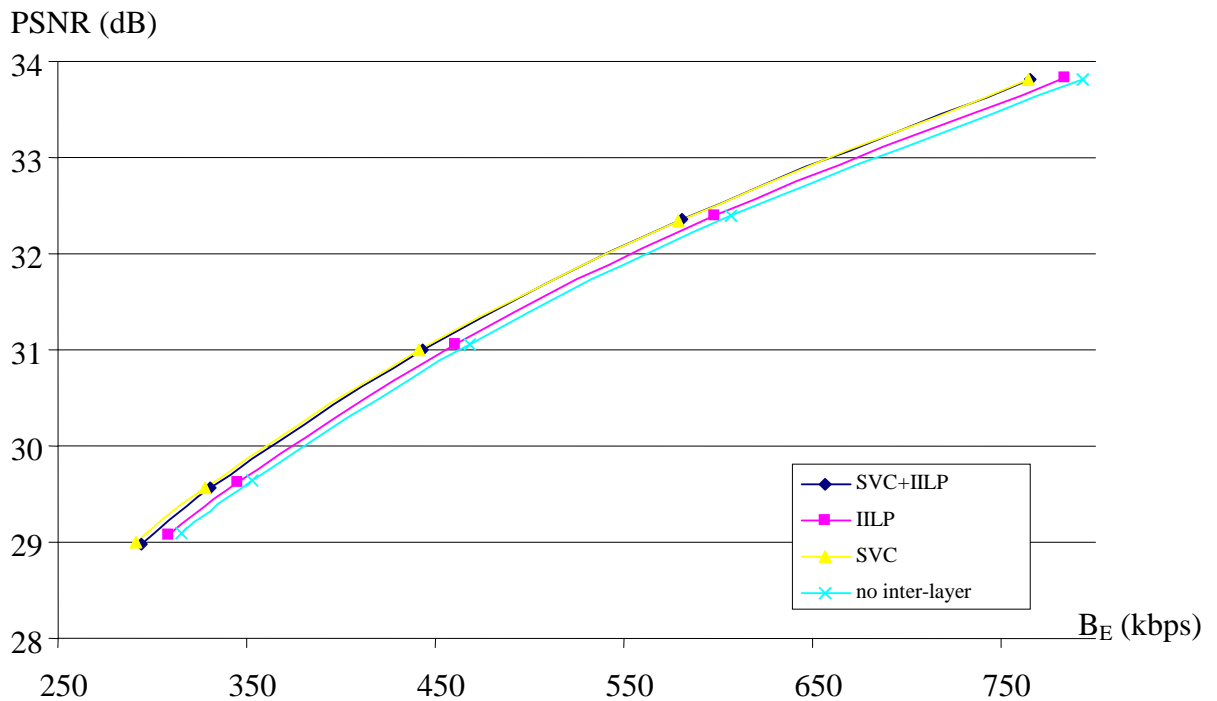


Fig. 7.22. R-D curves for various techniques of inter-layer prediction of motion vectors in *Bus* sequence (352×288); SVC video codec.

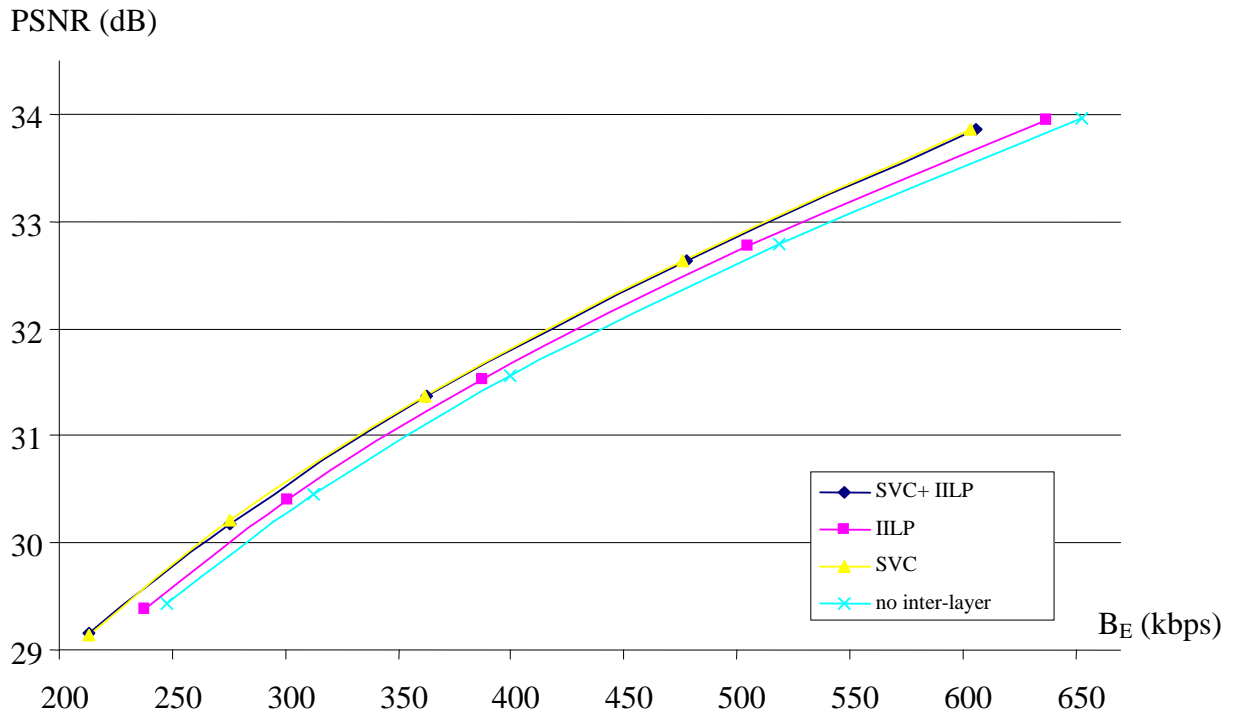


Fig. 7.23. R-D curves for various techniques of inter-layer prediction of motion vectors in *Football* sequence (352×288); SVC video codec.

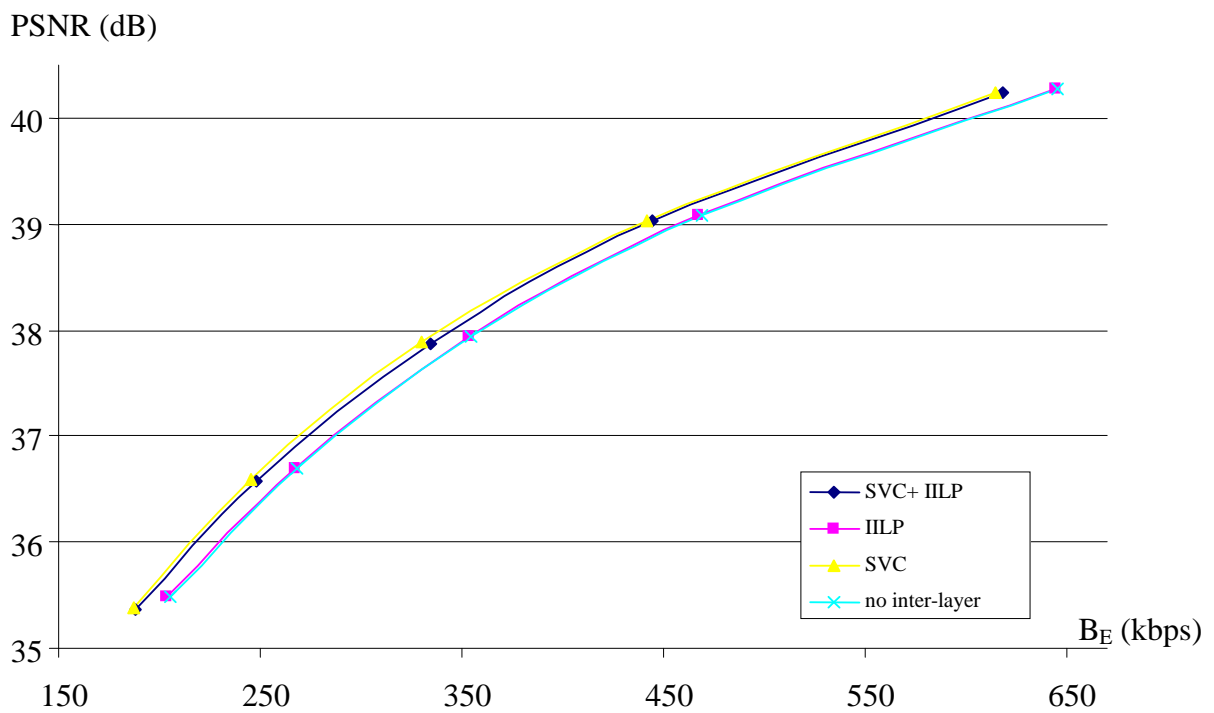


Fig. 7.24. R-D curves for various techniques of inter-layer prediction of motion vectors in *Foreman* sequence (352×288); SVC video codec.

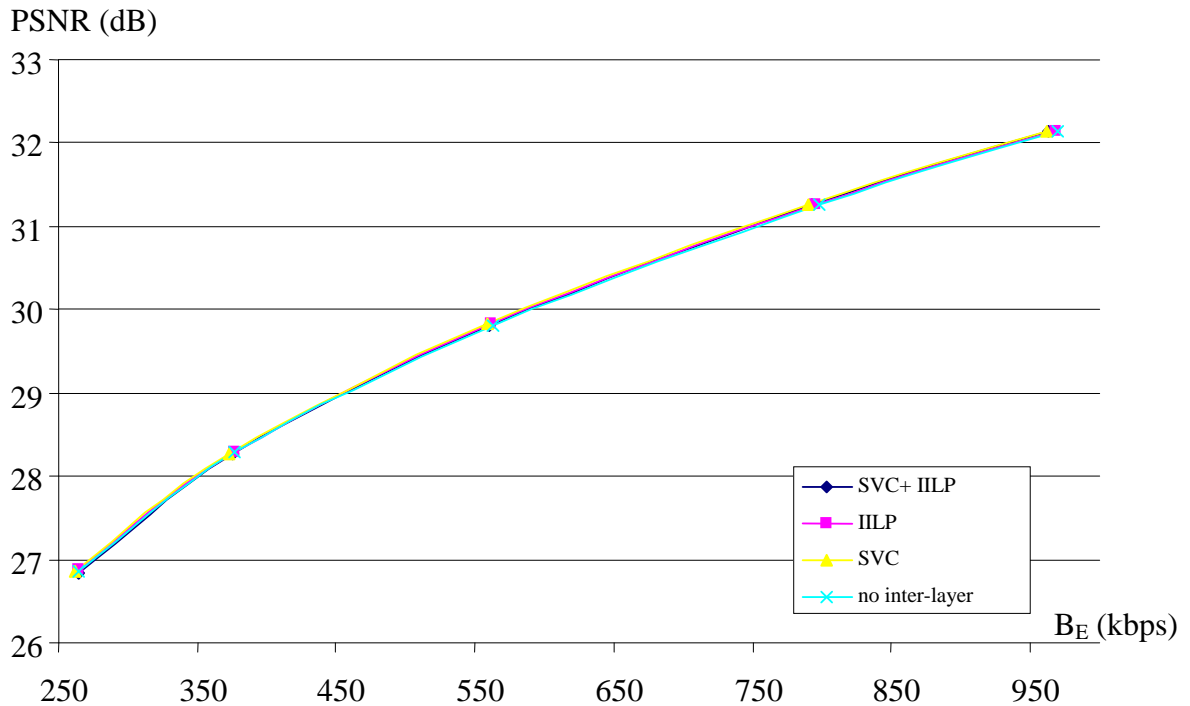


Fig. 7.25. R-D curves for various techniques of inter-layer prediction of motion vectors in *Mobile* sequence (352×288); SVC video codec.

The overall coding efficiency is illustrated by the R-D curves in Fig. 7.22-7.25. In all cases, the SVC codec with explicitly signaled inter-layer prediction of motion vectors achieved the best compression efficiency. Unfortunately, the IILP technique applied together with the original method implemented in SVC did not improve further the efficiency of the codec. However, IILP applied alone reduces the bitrate as compared with the solution without inter-layer prediction of motion vectors.

The maximum gain of using the inter-layer prediction of the motion vectors was achieved in the sequences *Bus* and *Football* (Fig 7.22 and 7.23). On the other hand, in *Mobile* video sequence, all techniques of motion vectors representation in SVC gives comparable results (Fig.7.25).

7.8.1.2. Average absolute values of motion vector prediction residuals

The impact of the applied method of inter-layer motion vectors prediction on motion vector prediction residuals is presented in this section. The average absolute value of

motion vector prediction residuals for various techniques of inter-layer motion prediction in SVC codec are depicted in Fig. 7.26 – Fig. 7.29.

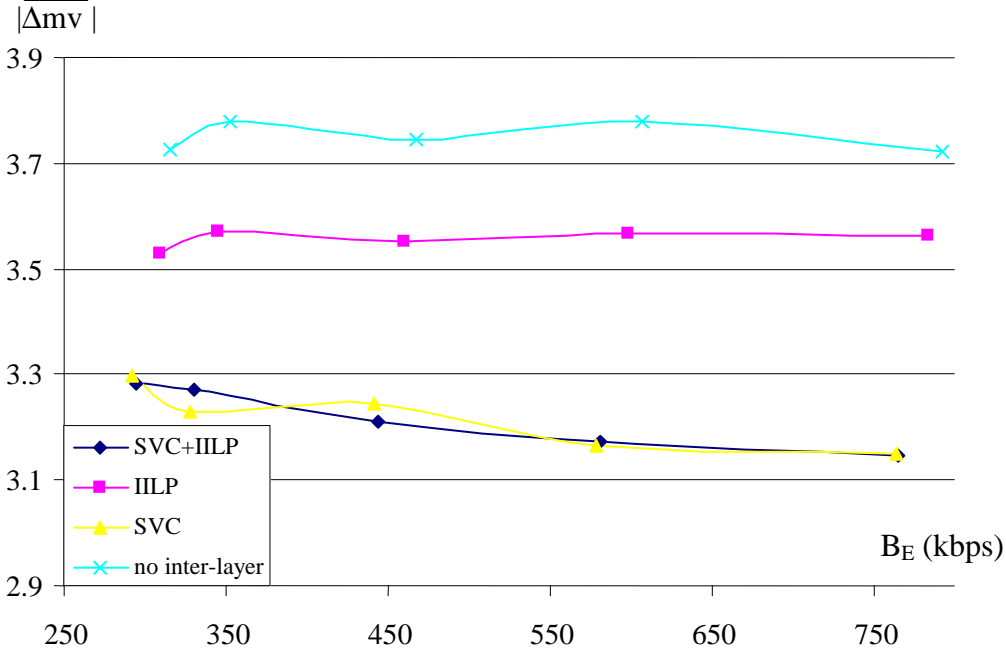


Fig. 7.26. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Bus* sequence (352×288); SVC video codec.

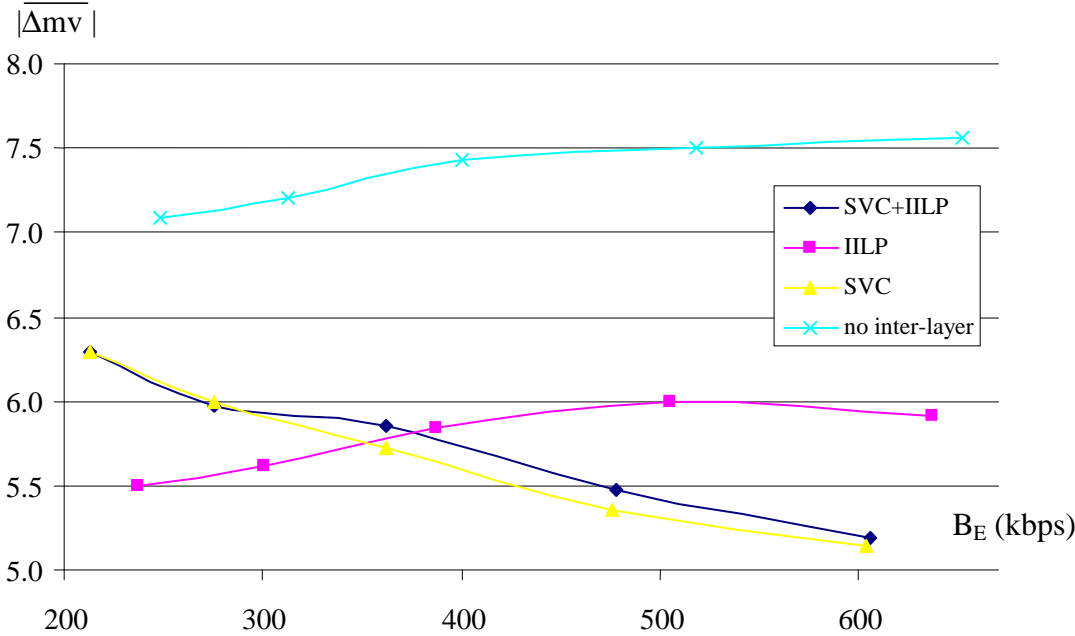


Fig. 7.27. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Football* sequence (352×288); SVC video codec.

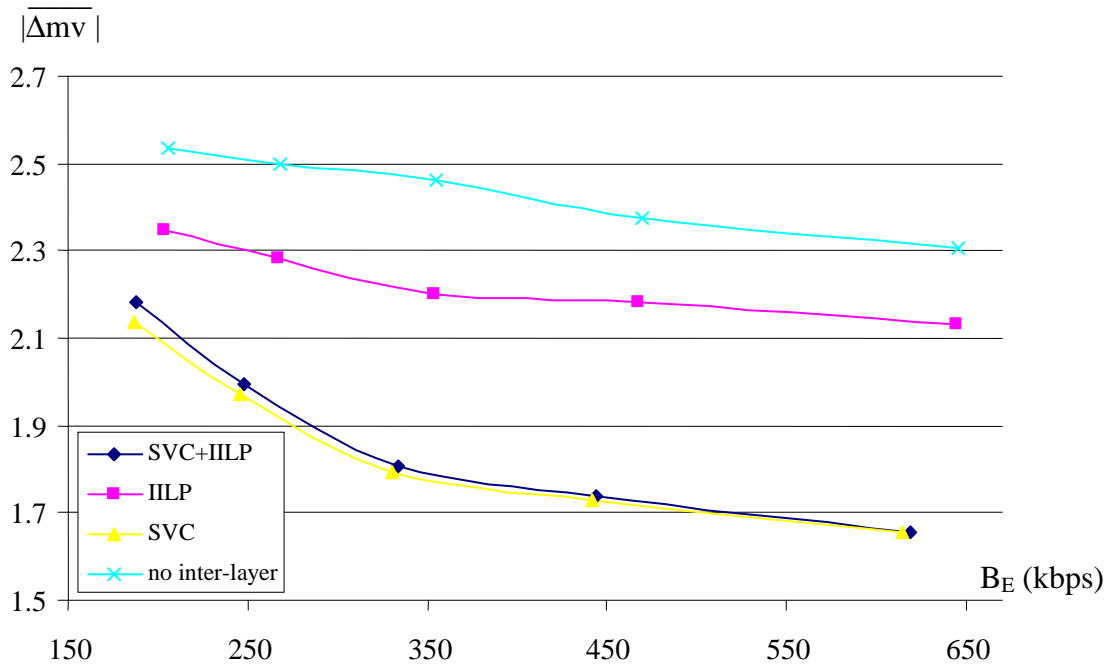


Fig. 7.28. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Foreman* sequence (352×288); SVC video codec.

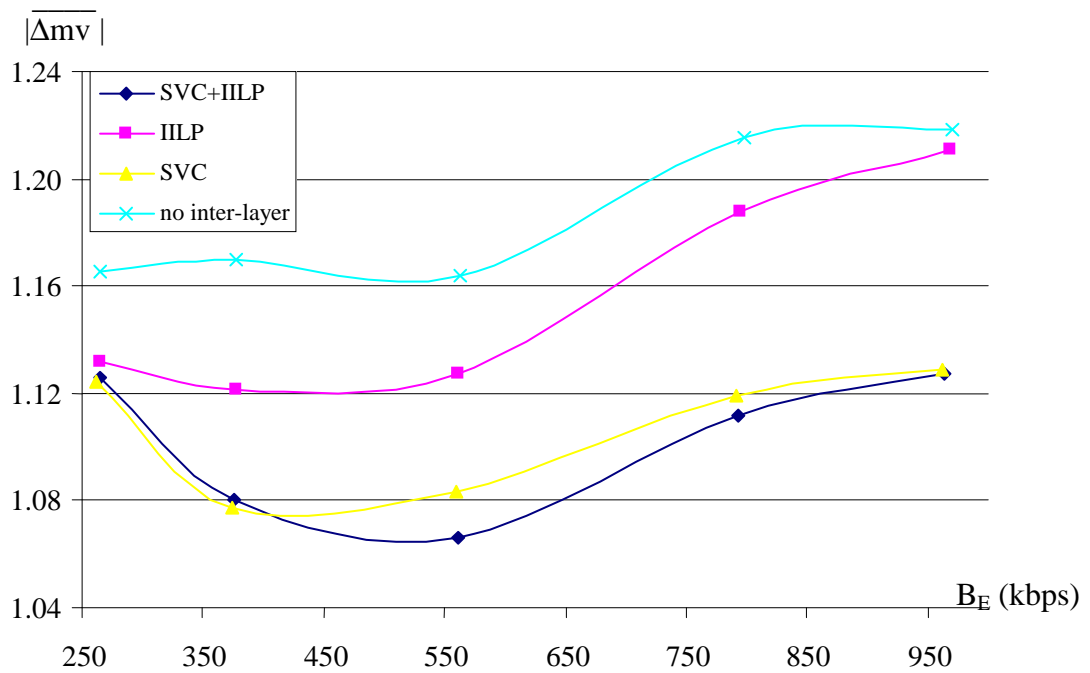


Fig. 7.29. Average absolute values of components of motion vector residual for various techniques of inter-layer prediction of motion vectors, *Mobile* sequence (352×288); SVC video codec.

In all cases, the average motion vector residual decreased when any of the inter-layer motion vector prediction algorithms was applied. The minimum value of the average

motion vector residual was achieved using the original SVC algorithm (*Foreman, Football*) or using SVC algorithm together with IILP technique (*Bus, Mobile*).

7.8.1.3. Subjective evaluation of the quality

The efficiency of the proposed techniques of motion vector coding in SVC codec was performed using Single Stimulus Multimedia (SSMM) method of subjective assessment of quality [Bar04]. The SSMM technique was described more precisely in Section 1.5. Because of economical reasons, only two video sequences were used (*Bus* and *Football*). Each sequence was encoded with two various values of quantization parameter Q_p . The results of the subjective assessments are depicted in Fig. 7.30 and Fig. 7.31.

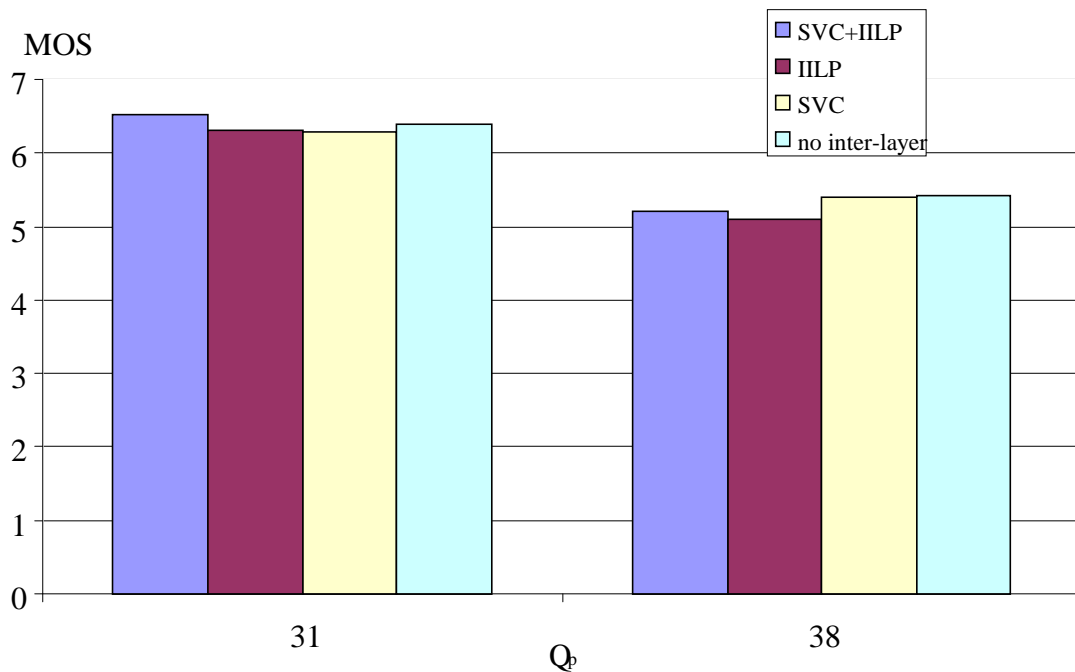


Fig. 7.30. The results of subjective assessment of the quality of *Bus* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality tests performed for *Bus* video sequence encoded with the value of quantization parameter $Q_p=31$, the highest value of Mean Opinion Score (MOS=6.5) was achieved for the scalable codec that exploits jointly the original SVC technique of motion vector prediction and the author's IILP algorithm. Surprisingly, in this test, the SVC codec that did not exploit inter-layer representation of the motion

vectors achieved better result (MOS=6.4) than the codecs using inter-layer motion prediction techniques (SVC or IILP) implemented alone (MOS=6.3). In the tests performed for the same video sequence encoded with a higher value of quantization parameter $Q_p=38$, the highest values of MOS (MOS=5.4) were achieved for the scalable codec using the original SVC method of motion vector prediction and for the scalable codec that did not use inter-layer motion vector prediction at all. Codecs that use IILP algorithm achieved the values of MOS equal to 5.2 and 5.1 (SVC+IILP and IILP, respectively).

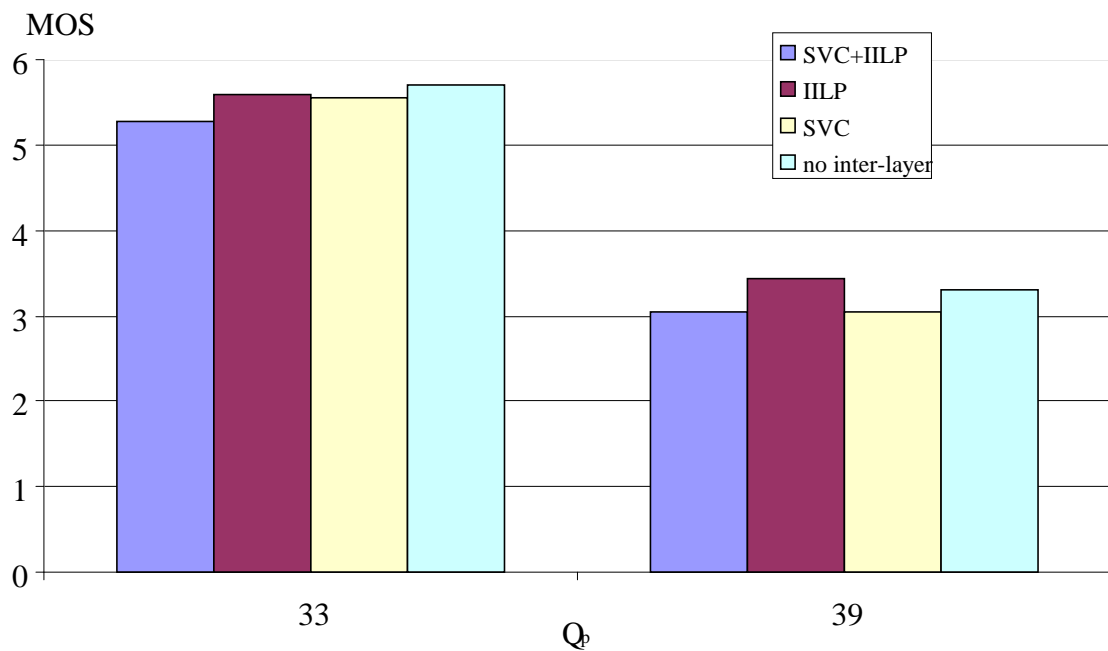


Fig. 7.31. The results of subjective assessment of the quality of *Football* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality evaluation performed for *Football* video sequence encoded with quantization parameter $Q_p=33$, the highest value of MOS (MOS=5.7) was achieved by the scalable video codec that did not use inter-layer prediction of motion vectors. A slightly lower value of MOS (MOS=5.6) was achieved by the video codec that exploited the author's IILP technique. The remaining video codecs achieved the values of MOS equal to 5.3 (SVC+IILP algorithms) and 5.55 (original SVC technique). In the tests performed for the video sequence encoded with the value of quantization parameter $Q_p=39$, the highest value of MOS (MOS=3.45) was achieved for the scalable codec using the algorithm of IILP motion vector prediction. Other codecs achieved the values

of MOS equal to 3.05 (SVC+IILP and SVC versions) and 3.3 (the codec without inter-layer prediction of motion vectors).

7.8.1.4. Complexity estimation

There has been made an attempt of estimation and comparison of the complexity of competitive methods of inter-layer motion vectors prediction. In Tab. 7.20 and Tab. 7.21 the execution time of encoding and decoding is given for different inter-layer motion vector coding techniques. Benchmarks of the encoder were performed by encoding of 100 frames of the sequence *Bus* and *City* with the constant quantization parameter $Q_P=32$. The value of Q_P parameter does not change significantly the encoding time. Benchmarks of the decoder were performed by decoding two bitstreams that contain 150 frames of the sequence *Bus* with the overall bitrates of about 400 kbps and about 1000 kbps. For higher accuracy, encoder and decoder executables were benchmarked three times and the average execution times were calculated. All experiments were performed on a workstation with double Intel Xeon 3.6 GHz processors, 2GB of RAM and with Windows XP Professional installed.

Relative values of execution time have been calculated in reference to original SVC codec. The estimated values allow for easy comparison of the complexity of benchmarked codecs.

Tab. 7.20. Execution time of encoder for various techniques of inter-layer prediction of motion vectors in SVC video codec

	CIF (~700kbps)		4CIF (~1500kbps)	
	average encoding time of one frame t_E (s)	relative to original SVC $\frac{t_E}{t_{E_{SVC}}} \cdot 100$ (%)	average encoding time of one frame t_E (s)	relative to original SVC $\frac{t_E}{t_{E_{SVC}}} \cdot 100$ (%)
SVC + IILP	3.749	98.5	11.104	99.4
IILP	2.230	58.6	7.039	63.0
SVC ($t_{E_{SVC}}$)	3.804	100.0	11.17	100.0
no inter-layer	2.542	66.8	7.206	64.5

IILP technique significantly reduces the time needed to perform the encoding operation both for CIF video sequences and 4CIF video sequences. Moreover, the encoder using IILP algorithm performs even faster than that without inter-layer motion vector prediction at all. The explanation is that IILP algorithm allow for fast finding of sub-optimal motion vector predicted from the base layer that provide very good motion-compensated prediction in the enhancement layer, hence the motion estimation does not need to be performed in some cases.

Tab. 7.21. Execution time of decoder for various techniques of inter-layer prediction of motion vectors in SVC video codec

	~400kbps (CIF)		~1000kbps (CIF)	
	average decoding time of one frame	relative to original SVC	average decoding time of one frame	relative to original SVC
	t_D (ms)	$\frac{t_D}{t_{D_{SVC}}} \cdot 100$ (%)	t_D (ms)	$\frac{t_D}{t_{D_{SVC}}} \cdot 100$ (%)
SVC + IILP	98.116	95.5	105.553	99.0
IILP	92.416	90.0	90.066	84.5
SVC ($t_{D_{SVC}}$)	102.718	100.0	106.6	100.0
no inter-layer	87.158	84.9	90.491	84.9

The decoding time (Tab 7.21) is also shorter for the decoder using IILP technique as compared with the standard SVC decoder. However, in this case, the bitstreams that did not contain inter-layer motion vector dependencies were decoded faster (for 400kbps bitstream) or almost as fast as (for 1000kbps bitstream) the bitstream encoded using IILP prediction.

7.8.2. Comparison of inter-layer motion prediction techniques in SVC codec – conclusions

The comparison of four techniques of motion vectors encoding has been performed using the following modifications of the SVC reference codec:

- SVC codec without the inter-layer motion prediction (referred to as **no inter-layer**),
- SVC codec with explicitly signaled inter-layer motion information prediction as defined in a draft of SVC (referred to as **SVC**),
- SVC codec with the author's IILP proposal, described in Section 7.3.3 (referred to as **IILP**),
- SVC codec with jointly used: SVC and IILP technique of inter-layer motion prediction (referred to as **SVC+IILP**).

The standard inter-layer algorithm of motion prediction implemented in SVC results in increase of the PSNR parameter for luminance up to 0.3 dB as compared with the SVC codec without inter-layer motion prediction. A significant gain is achieved in *Bus*, *Football* and *Foreman* sequences for the given configuration of the encoder. However, PSNR in *Mobile* sequence remains almost constant for all tested codecs, regardless of the technique of motion vector representation.

In *Bus* and *Football* sequences, IILP technique provides better compression efficiency as compared with the encoder that does not exploit inter-layer correlations; however, SVC method of inter-layer motion prediction gives even better results (Fig. 7.22 and Fig. 7.23). Joint usage of IILP technique with the existing SVC method of inter-layer motion prediction does not improve coding efficiency (Fig. 7.22-7.25).

Subjective comparison of the efficiency of proposed solutions of motion vector coding (Fig. 7.30 and Fig. 7.31) depicts that in 2 of 4 cases inter-layer technique of motion vector representation gives higher quality of encoded video sequence than the algorithm that does not exploit inter-layer dependencies between motion vectors. In one case, the quality of the video sequences was judged the same for the codec that utilized inter-layer prediction of motion vector as for the codec that did not use such a prediction. In one case, the quality of the video sequence that had been encoded without inter-layer prediction of motion vectors, was judged to be higher than the quality of the video sequences encoded using inter-layer prediction of motion vectors.

Using of the IILP together with the original SVC method, in *Mobile* and *Bus* decreases motion vector prediction residual (Fig. 7.26, Fig. 7.29). In *Foreman* sequence original SVC inter-layer motion prediction provides the best results (Fig. 7.28), while in *Football* sequence the best results give the proposed technique, implemented alone (Fig. 7.27).

On the other hand, the complexity of the author's proposal is significantly lower than the complexity of the standard solution used in SVC codec (herein we consider the complexity as the time needed to perform encoding and decoding operations). Modified SVC encoder with IILP algorithm enabled performed about 40% faster on average than the original SVC encoder (Tab. 7.20). Surprisingly, the author's proposal of motion vector coding used together with the standard SVC technique also speeds-up the encoder. The encoder with IILP performs even faster than the encoder without any inter-layer motion prediction. The speed-up in the encoder is achieved mainly because of simplified decision process during R-D coding.

The differences in execution time of decoders are somewhat lower; however, the bitstream with a bitrate of 400 kbps is decoded 10% faster and the bitstream with a bitrate of 1000 kbps is decoded 15% faster when IILP method is used as compared to the standard SVC solution (Tab. 7.21). The speed-up in the decoder is achieved because of simplified bitstream parsing process and simplified motion vectors decoding.

The comparison of the author's IILP technique and the method of inter-layer motion vector representation used in SVC proves that both techniques reduce the bitrate for a given quality of decoded video sequence. In all experiments, SVC method insignificantly outperformed the solution presented in this thesis in a sense of overall coding efficiency. However, in some cases, the lowest motion vector residuals were achieved when IILP proposal of inter-layer motion prediction was used.

Since side information in SVC is coded very efficiently using CABAC, the decrease of the average motion vector residual when IILP was used did not result in the decrease of overall bitrate. The contextual model of entropy coder was not optimized for the author's original technique.

However, the IILP method of inter-layer motion information prediction significantly outperforms the SVC technique in a sense of time complexity at encoder side. As shown in Tab. 7.20 the codec with implemented IILP operates significantly faster than the standard SVC codec – both at the side of the encoder and at the side of the decoder.

7.9. Summary

In this chapter, there has been presented an original IILP technique of inter-layer prediction of motion vectors. The technique was first applied and tested in a scalable

video codec based on AVC/H.264 technology. This codec was developed at Poznań University of Technology.

The author's proposal has significantly reduced motion vector prediction residuals and has improved overall compression efficiency of the codec. A scalable video coder that exploits the proposed method, achieved one of the best results in a comparison of scalable codecs conducted by MPEG committee in 2004 [Bar04].

The performance of IILP technique was also experimentally tested in SVC video codec. SVC is the codec developed by MPEG in order to establish a new scalable video coding standard. The algorithm employed in SVC utilizes another technique of inter-layer motion prediction, which was proposed after the author's contribution with a description of IILP technique. In Section 7.8, this author's approach was experimentally compared against the technique used in SVC codec.

The experiments proved that both techniques of inter-layer motion vectors coding improve compression efficiency in SVC codec. In the experiments, the original SVC codec achieved better results. However, the complexity of SVC solution is much higher as compared with IILP approach; it has been depicted in Section 7.8.1.3.

In the following chapter, a new technique of joint multiresolution coding of motion vectors in temporally scalable codec is introduced. This author's original proposal is presented and experimentally tested in SVC video codec.

Chapter 8.

Joint multiresolution coding of motion vectors in temporally scalable codec

8.1. Temporal scalability

Temporal scalability allows for representation of video sequence with various temporal resolutions using a single bitstream. Decoding of a whole bitstream enables to achieve full frame-rate of the video sequence, while decoding of the part of a bitstream enables to achieve reduced frame-rate of the video sequence.

In classic hybrid video coding, temporal scalability is achieved using B-frames dropping. Dropped B-frames are not used as reference frames for motion-compensated prediction of other frames [ISO94]. The idea is depicted in Fig. 8.1.

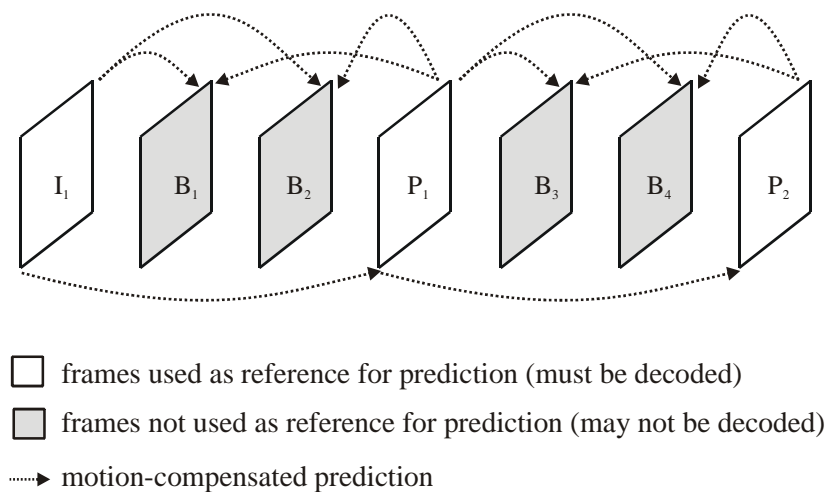


Fig. 8.1. Using of B-frames in order to achieve temporal scalability.

Bidirectionally encoded frames can be dropped during decoding of a bitstream, because they are not used for prediction of any other frames. Since specific part of a bitstream, which contains the data of B-frames, does not have to be decoded at all, this technique introduces temporal scalability of a bitstream.

The hierarchically organized Group of Pictures (GOP) allows for many levels of temporal scalability, as depicted in Fig. 8.2. The number of levels depends on the number of consecutive B-frames in GOP.

Dropping of frames marked as B₁, B₃, B₄ and B₆ in Fig. 8.2 causes the decrease of temporal resolution by a factor of 2. Further dropping of frames marked as B₂ and B₅ causes the decrease of temporal resolution of given video sequence to $\frac{1}{4}$ of the original frame-rate.

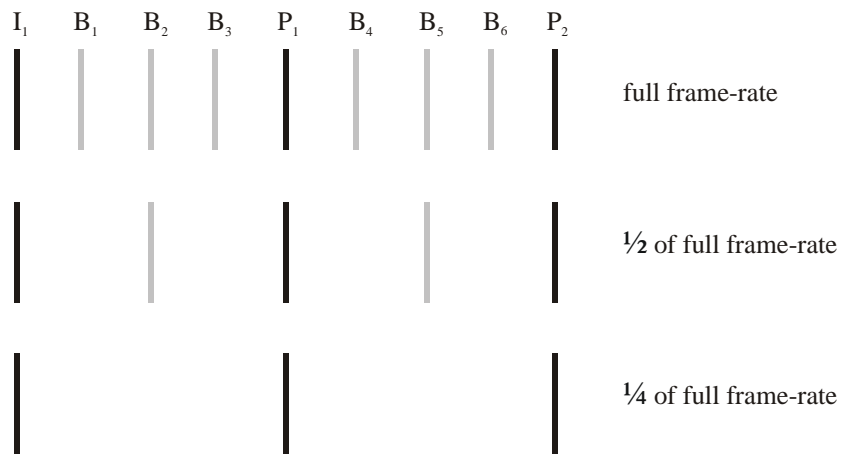


Fig. 8.2. The structure of GOP that allows for 3 levels of temporal scalability.

The technique of B-frames dropping used in order to achieve temporal scalability is simple and gives subjectively good results in the output video. This technique was adopted in most of scalable profiles of hybrid video coders (e.g. MPEG-2, H.263, MPEG-4, SVC [ISO94, ITU05, ISO98, JVT06-02]). Dropping of some arbitrary chosen frames from the original video sequence introduces temporal aliasing [Dom99], although in most cases this aliasing is hardly visible in decoded video sequence.

8.2. Original proposal of joint multiresolution coding of motion vectors in B-frames

In bidirectionally coded frames all kinds of motion-compensated prediction can be exploited: forward prediction, backward prediction and bidirectional prediction. Therefore each coded block can be predicted using one or two motion vectors. However, as concluded in Chapter 5, there is very high similarity between estimated motion vectors from low-resolution video sequence and estimated motion vectors from high-resolution video sequence, especially in B-frames.

On the other hand, the author's experiments from Chapter 4 proved that usually, motion vector prediction error is significantly lower in B-frames than in P-frames [Lan06]. Other author's experiments, not presented in this dissertation, showed, that in SVC video codec, inter-layer motion prediction mode in B-frames is chosen very often.

Summarizing, during the designing stage of the technique of joint multiresolution motion representation in B-frames, the following circumstances were taken into consideration:

- high correlation exists between motion vectors from low-resolution layer and high-resolution layer,
- motion vector prediction errors are very low in B-frames,
- inter-layer motion vector prediction mode is chosen very often in B-frames in SVC video codec.

Originally, inter-layer motion prediction mode is signaled in SVC bitstream using at least two syntax elements for each macroblock, as described in Section 7.7.2. Such an approach generates quite large, constant sub-bitstream of additional data that has to be sent to the decoder (the bitrate of this sub-bitstream is about 24 kbps for CIF video sequence and about 95 kbps for 4CIF video sequence).

The original author's proposal presented in this chapter introduces fixed inter-layer motion prediction in B-frames. This inter-layer motion information prediction mode is not explicitly signaled in a bitstream. Instead, inter-layer motion inference is made always when motion-compensated prediction is used in a low-resolution base layer. The possible motion predictions in modified SVC codec are showed in Fig. 8.3.

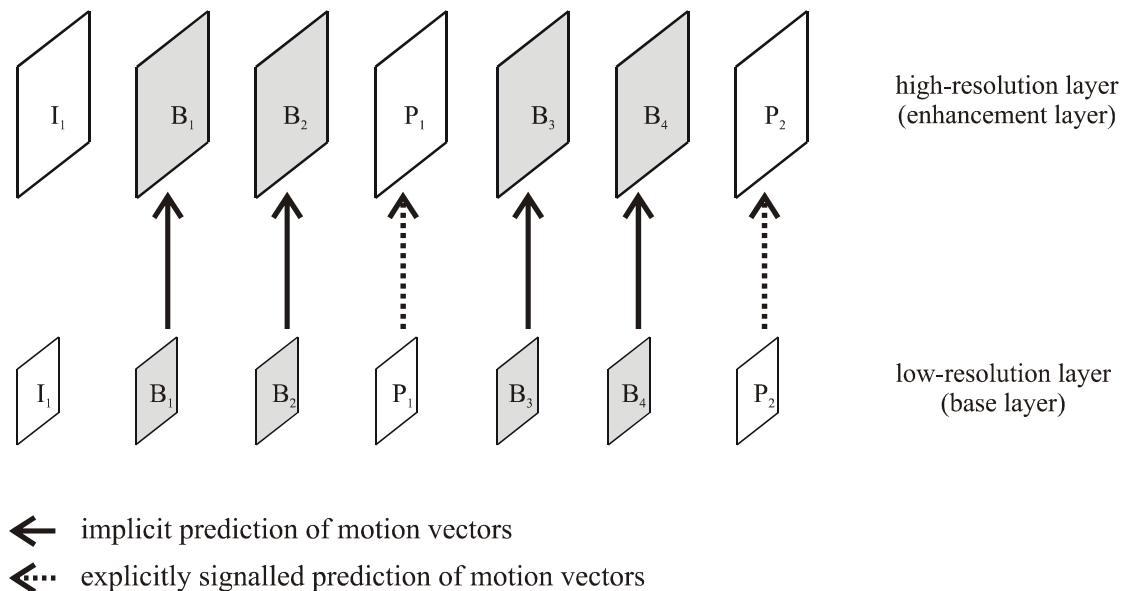


Fig. 8.3. Motion vectors prediction used in modified joint multiresolution coding of motion vectors in B-frames.

Since the rescaled motion vectors from the base layer have lower accuracy than the motion vectors estimated in the enhancement layer, the extension of given algorithm has been proposed as well. The modified algorithm assumes sending $\frac{1}{4}$ -pel correction value for each motion vector interpolated from the low-resolution layer in order to increase the accuracy of motion vectors and refine motion vector field. Consequently two new methods of motion vectors coding in B-frames have been proposed and experimentally tested:

- implicit prediction of the motion data from low-resolution layer (referred to as **INTER-B#1**),
- implicit prediction of the motion data from low-resolution layer and $\frac{1}{4}$ -pel motion vectors refinement (referred to as **INTER-B#2**).

8.3. Modification of SVC codec

The proposed algorithms have been implemented in version 4.2 of the SVC reference software. The algorithm of B-frames coding has been modified in such a way that interpolated motion vectors from low-resolution layer are used in order to perform motion-compensated prediction in B-frames. The type of motion-compensated prediction (forward, backward or bi-directional) and partitioning of a macroblock in the enhancement layer are derived from the data of the co-located macroblock from the base layer. Additionally, refining $\frac{1}{4}$ -pel motion estimation is performed for the latter proposed algorithm (INTER-B#2).

When the co-located macroblock from the base layer is not coded using motion-compensated prediction (intra coded macroblock), then corresponding macroblock in high-resolution layer is coded using regular SVC syntax and semantics.

The syntax of a bitstream has been modified: syntax elements that were originally used in order to signal inter-layer motion prediction are no longer present in a bitstream, according to Tab. 8.1.

Tab. 8.1. Multiresolution coding of motion vectors in B-frames, techniques and bitstream syntax.

	available modes of motion vectors encoding	syntax elements present in a bitstream
standard coding (SVC)	– standard intra-layer prediction – inter-layer prediction – inter-layer prediction + 1/4-pel refinement	base_mode_flag base_mode_refinement_flag
proposal #1 (INTER-B#1)	– inter-layer prediction	–
proposal #2 (INTER-B#2)	– inter-layer prediction + 1/4-pel refinement	–

In intra coded frames (I-frames) and in frames coded using unidirectional prediction from the past only (P-frames), the compression efficiency, as well as bitstream syntax and semantics are exactly the same as in standard SVC video codec. Therefore the performance of coding of I-frames and P-frames is not affected by the originally proposed techniques.

8.4. Joint multiresolution coding of motion vectors in B-frames – experimental results

The comparison of the efficiency of proposed algorithms of joint multiresolution motion vectors coding in B-frames have been performed for the CIF sequences *Bus*, *Foreman*, *Football* and *Mobile* using modified SVC scalable codec (version 4.2). The coder was set for producing two spatial layers and two temporal layers. The temporal scalability was achieved using dropping of B-frames. For each video sequence the achieved bitrate was measured for different techniques of joint multiresolution coding of motion vectors in B-frames. The bitrates of motion vector residuals sub-bitstream, transform coefficients sub-bitstream and control data sub-bitstream were measured as well. The subjective evaluations of the quality of the encoded video sequences *Bus* and *Mobile* were performed using the SSMM technique described in Section 1.5. The complexity has been estimated for each of the proposed method of motion vectors representation in B-frames.

The following parameters have been set in the configuration file of the encoder:

- period between I frames: 96,
- group of pictures: I-B-P-B-P,
- number of reference frames: 3,
- entropy coding: CABAC,
- range of motion estimation +/- 96 samples (full-pel units),
- adaptive inter-layer prediction,
- range of low-resolution bitrate: 70 kbps – 280 kbps.
- range of high-resolution bitrate: 200 kbps – 1000 kbps.

In all diagrams, the bitrate of high-resolution layer is denoted as B_E , while the overall bitrate is denoted as B . The variants of modified SVC codec are denoted using the symbols given in Tab. 8.1.

8.4.1. Bitrate and distortion

The achieved bitrate and the values of PSNR in tested sequences for different quantization parameter Q_P are shown in Tab. 8.2 – Tab. 8.5. Rate-distortion curves for all test sequences are depicted in Fig. 8.4 – Fig. 8.7. The proposed algorithms change the performance of coding of B-frames only (temporal enhancement layer). Therefore, in Fig. 8.8 – Fig. 8.11 the values of PSNR are shown in consecutive B-frames for all test sequences, for the given value of quantization parameter Q_P . In Tab. 8.6 the average decreases of PSNR Δ_{PSNR} in B-frames are given for all test sequences.

Tab. 8.2. Bitrate (kbps) and PSNR (dB) in *Bus* (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.

(kbps)/(dB)		$Q_P=31$	$Q_P=33$	$Q_P=35$	$Q_P=37$	$Q_P=38$
base layer		239.5/33.48	184.5/31.93	142.3/30.50	107.3/29.02	95.5/28.43
enhancement layer	SVC	764.4/33.81	578.6/32.34	441.5/31.00	327.9/29.56	291.5/29.00
	INTER-B#1	804.9/33.35	605.9/31.85	452.7/30.46	329.8/28.99	289.2/28.41
	INTER-B#2	784.7/33.58	593.1/32.10	449.8/30.73	331.7/29.26	291.5/28.68

Tab. 8.3. Bitrate (kbps) and PSNR (dB) in *Football* (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.

(kbps)/(dB)		Q_p=33	Q_p=35	Q_p=37	Q_p=39	Q_p=41
base layer		279.6/33.02	219.4/31.74	167.0/30.51	129.3/29.26	99.3/28.12
enhancement layer	SVC	603.7/33.86	475.9/32.64	361.7/31.36	275.5/30.21	213.3/29.14
	INTER-B#1	615.8/33.34	473.8/32.06	352.1/30.80	262.1/29.65	200.2/28.56
	INTER-B#2	608.4/33.48	472.8/32.22	354.5/30.95	267.6/29.80	203.2/28.71

Tab. 8.4. Bitrate (kbps) and PSNR (dB) in *Foreman* (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.

(kbps)/(dB)		Q_p=25	Q_p=27	Q_p=29	Q_p=31	Q_p=33
base layer		211.4/39.84	163.4/38.43	127.8/37.10	99.0/35.65	78.4/34.28
enhancement layer	SVC	614.3/39.74	441.5/38.53	330.4/37.39	245.4/36.09	187.2/34.88
	INTER-B#1	645.5/39.24	450.4/38.02	331.0/36.88	237.5/35.62	189.1/34.60
	INTER-B#2	636.5/39.44	451.4/38.23	336.3/37.09	245.2/35.83	185.7/34.65

Tab. 8.5. Bitrate (kbps) and PSNR (dB) in *Mobile* (352×288, IBPBP) sequence using joint multiresolution coding of motion vectors in B-frames in SVC video codec.

(kbps)/(dB)		Q_p=32	Q_p=33	Q_p=35	Q_p=37	Q_p=39
base layer		202.4/31.68	169.9/30.71	127.2/29.13	91.7/27.57	69.6/26.11
enhancement layer	SVC	961.6/32.15	790.6/31.26	559.3/29.82	373.5/28.27	261.8/26.85
	INTER-B#1	1012.2/31.77	833.7/30.82	586.5/29.29	383.5/27.72	266.6/26.29
	INTER-B#2	978.9/31.97	806.0/31.06	568.9/29.61	379.2/28.07	266.6/26.66

In all experiments the bitrate achieved by modified codecs increased. Since motion vectors in the enhancement layer are not estimated for B-frames, higher prediction errors need to be compensated by the additional transform coefficients that are transmitted in a bitstream.

The R-D curves in Fig. 8.4 – 8.7 allow for the comparison of average compression efficiency.

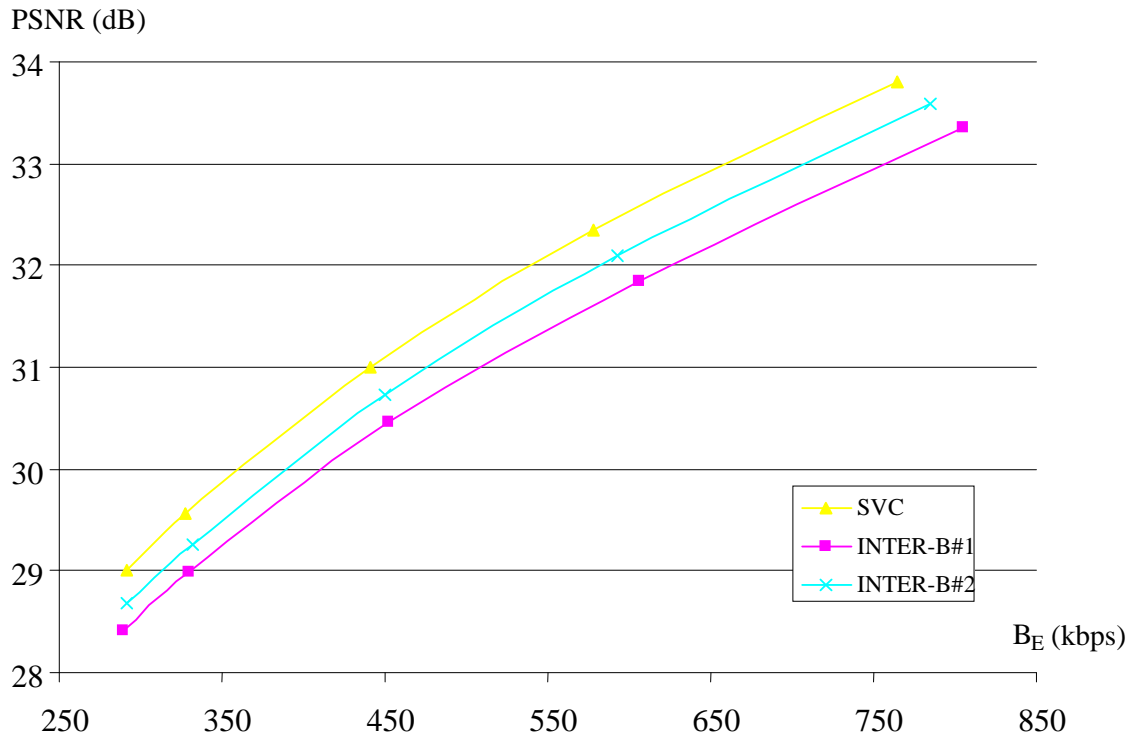


Fig. 8.4. R-D curves for various techniques of motion vectors coding in B-frames in *Bus* sequence (352×288); SVC video codec.

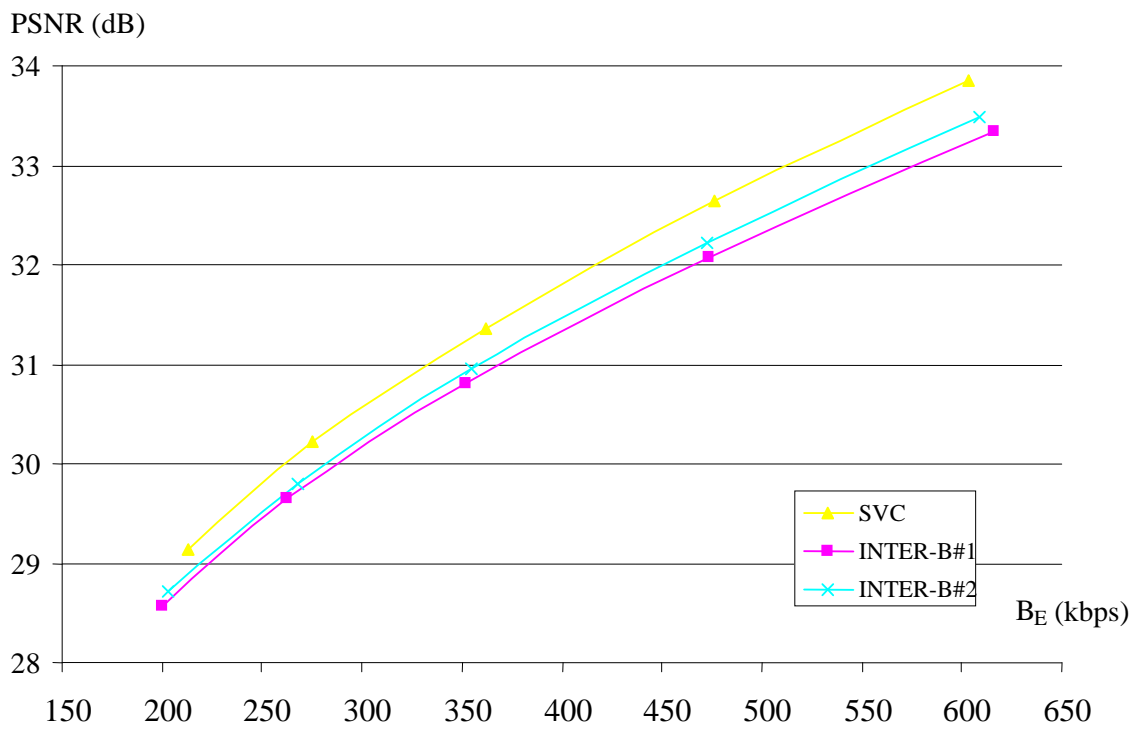


Fig. 8.5. R-D curves for various techniques of motion vectors coding in B-frames in *Football* sequence (352×288); SVC video codec.

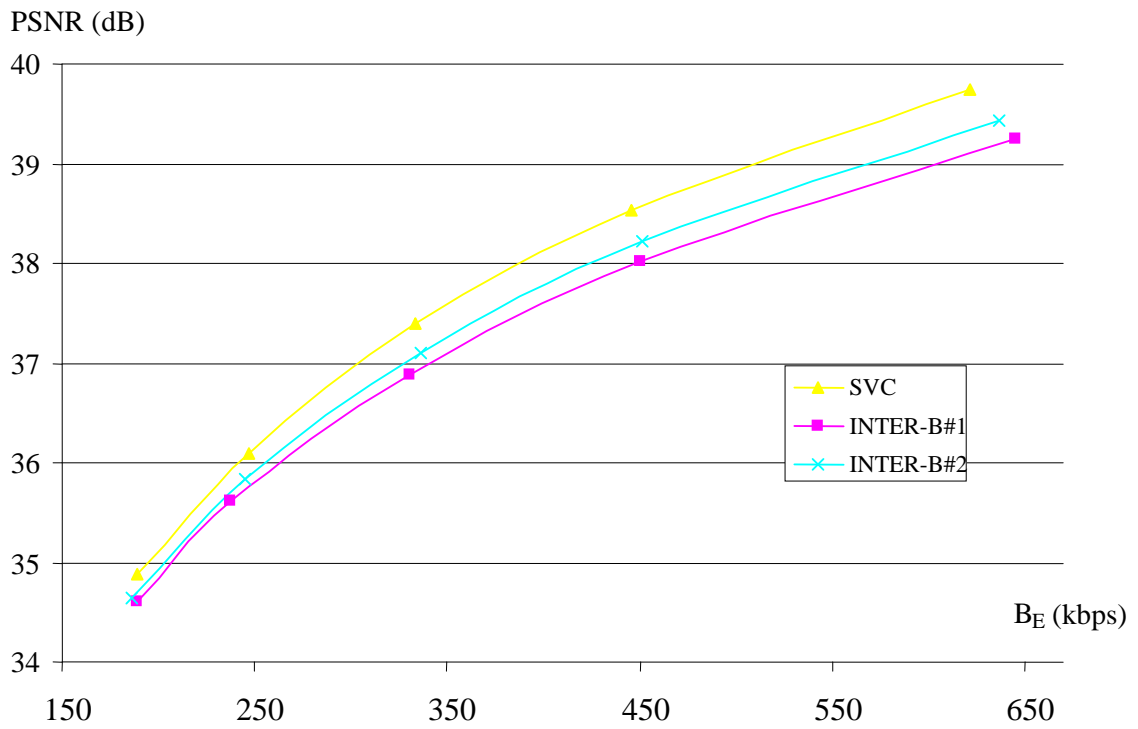


Fig. 8.6. R-D curves for various techniques of motion vectors coding in B-frames in *Foreman* sequence (352x288); SVC video codec.

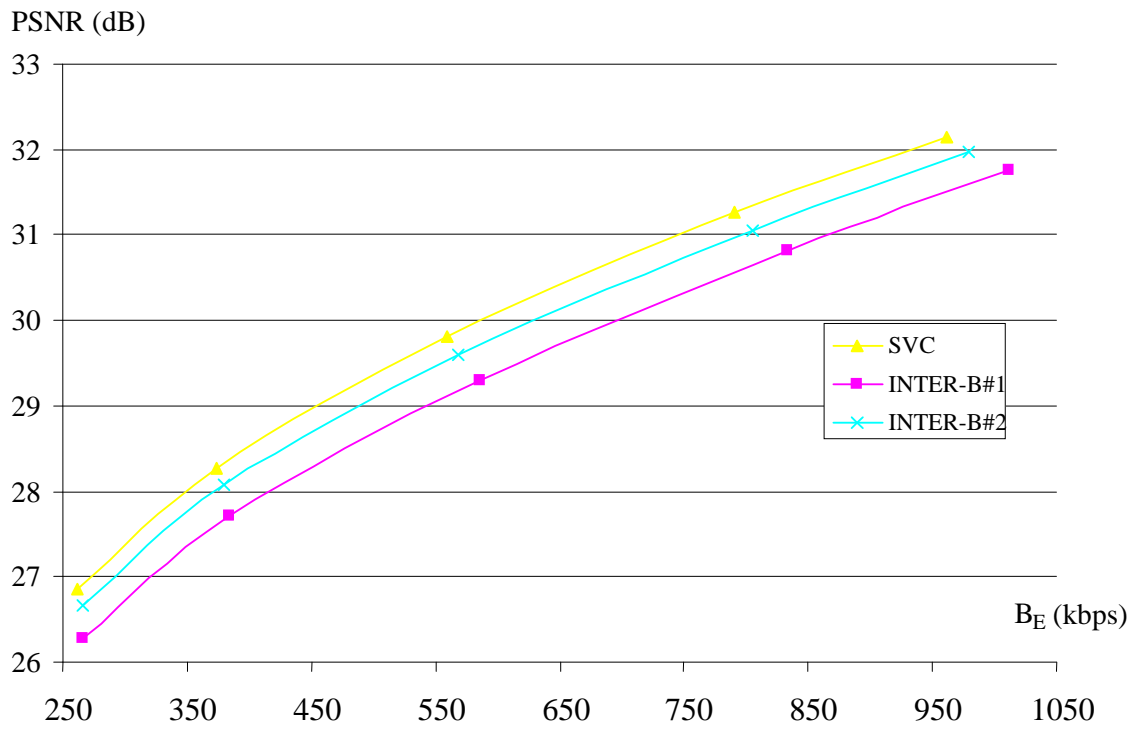


Fig. 8.7. R-D curves for various techniques of motion vectors coding in B-frames in *Mobile* sequence (352x288); SVC video codec.

The decreases of average PSNR were observed in all tested video sequences. Since the PSNR in I-frames and P-frames does not depend on the technique of motion vector coding in B-frames, in the figures below, the PSNR in consecutive B-frames is compared for different techniques of motion vectors coding.

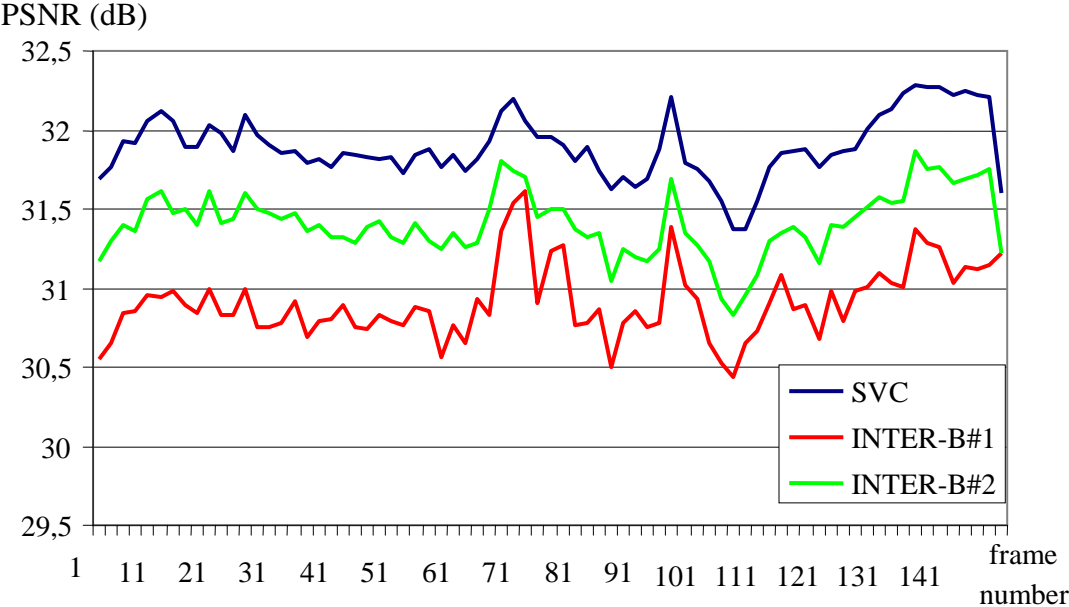


Fig. 8.8. The values of PSNR in B-frames for various techniques of motion vectors coding in *Bus* sequence (352×288); SVC video codec, $Q_P=33$.

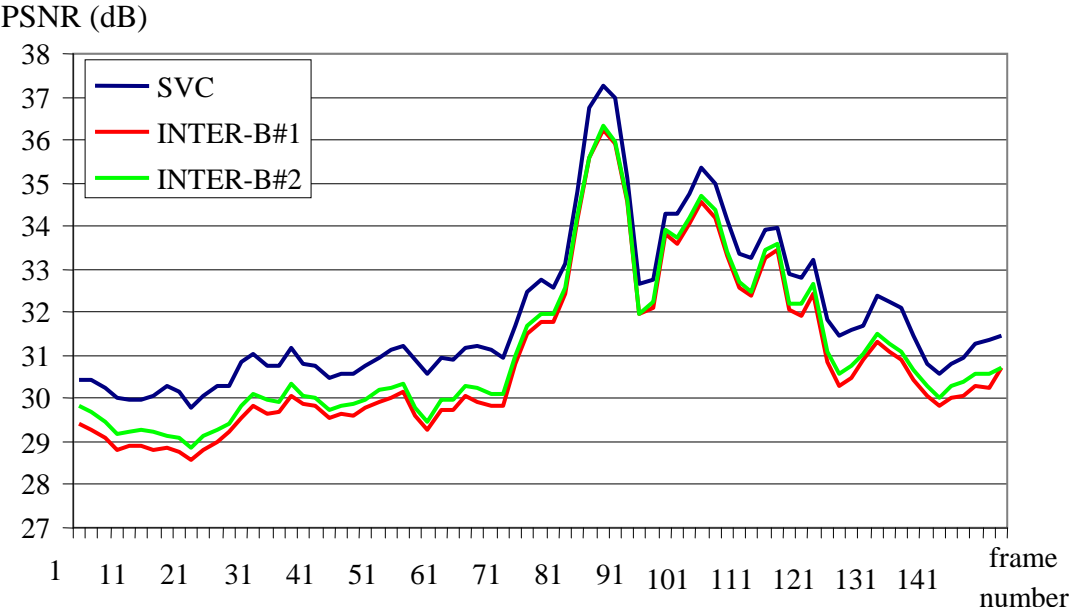


Fig. 8.9. The values of PSNR in B-frames for various techniques of motion vectors coding in *Football* sequence (352×288); SVC video codec, $Q_P=35$.

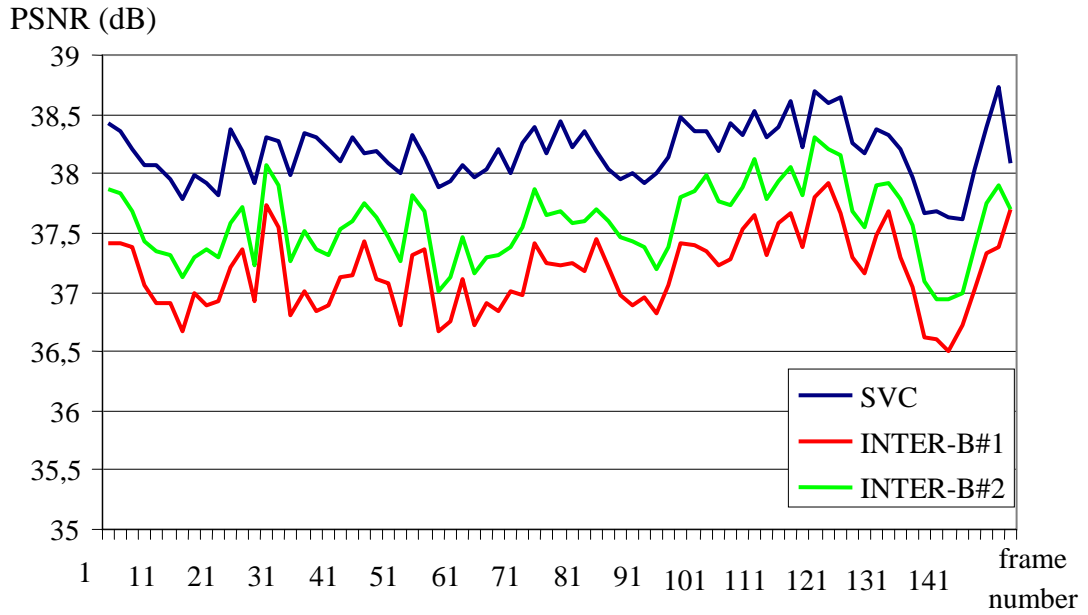


Fig. 8.10. The values of PSNR in B-frames for various techniques of motion vectors coding in *Foreman* sequence (352×288); SVC video codec, $Q_P=27$.

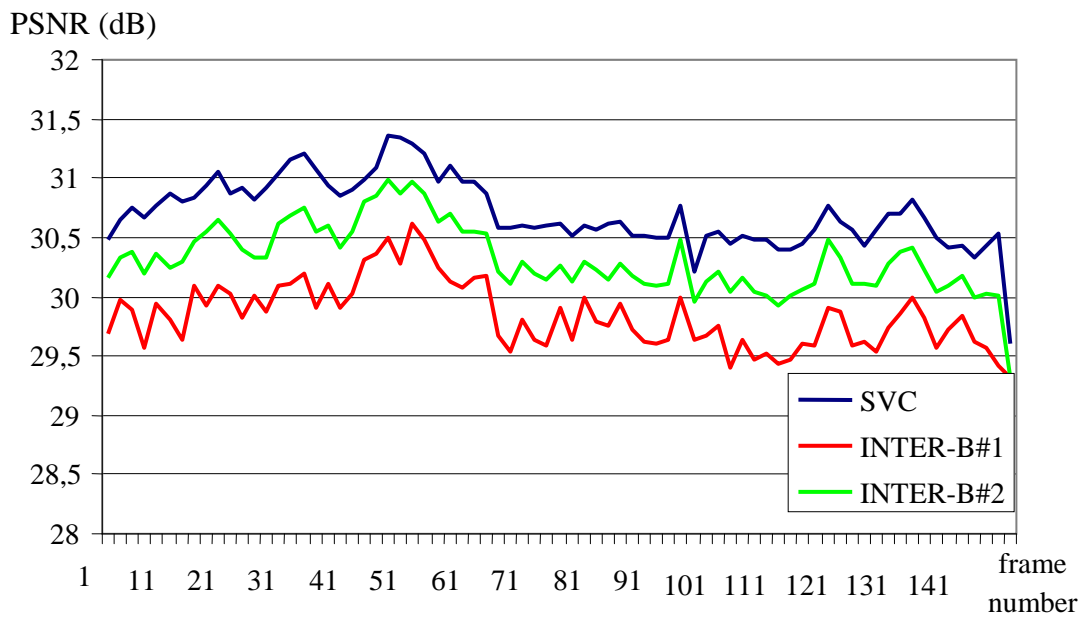


Fig. 8.11. The values of PSNR in B-frames for various techniques of motion vectors coding in *Mobile* sequence (352×288); SVC video codec, $Q_P=33$.

The decrease of PSNR in B-frames depends on the content of a video sequence. However, inside the given sequence it is almost constant. The average values of the decrease of PSNR in B-frames have been estimated and are presented in Tab. 8.6.

Tab. 8.6. Average decrease of PSNR in B-frames as compared with SVC codec for various techniques of motion vectors coding.

	Average decrease of PSNR in B-frames, Δ_{PSNR} (dB)			
	<i>Bus</i>	<i>Football</i>	<i>Foreman</i>	<i>Mobile</i>
INTER-B#1	0.985	0.987	1.007	0.866
INTER-B#2	0.487	0.771	0.597	0.390

The average decrease of PSNR in the temporal enhancement layer varies from 0.39dB to 1.007dB. The decrease of B-frames quality was definitely lower when the additional stage of $\frac{1}{4}$ -pel motion vectors refinement was used (INTER-B#2).

8.4.2. The bitrate of motion vector residuals sub-bitstream

In Fig. 8.12 – Fig. 8.15 the portion of the bitrate of motion vector residuals sub-bitstream p_{mv} in the overall bitrate is shown for various methods of motion vectors coding in B-frames.

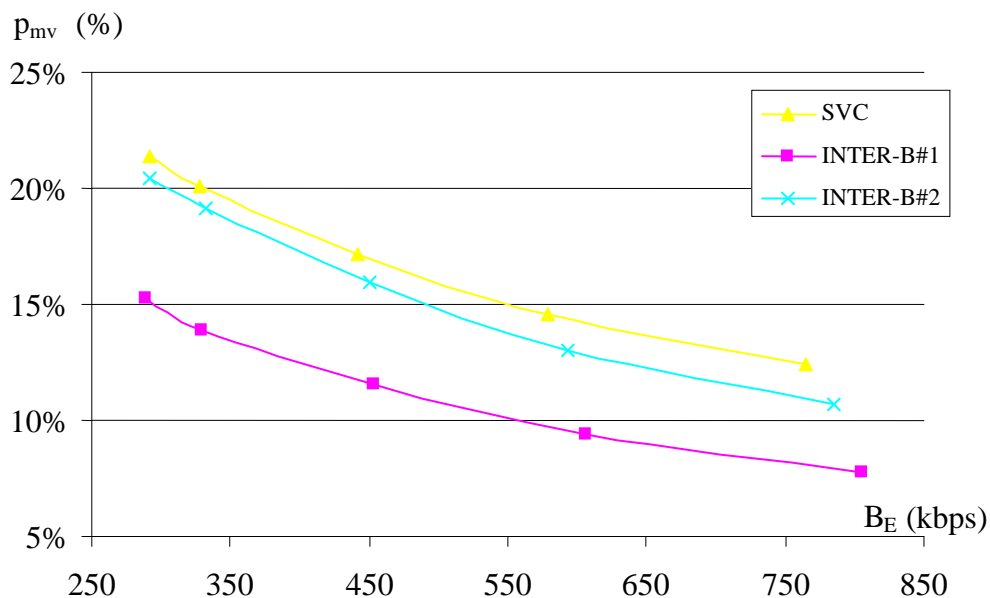


Fig. 8.12. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Bus* sequence (352×288); SVC video codec.

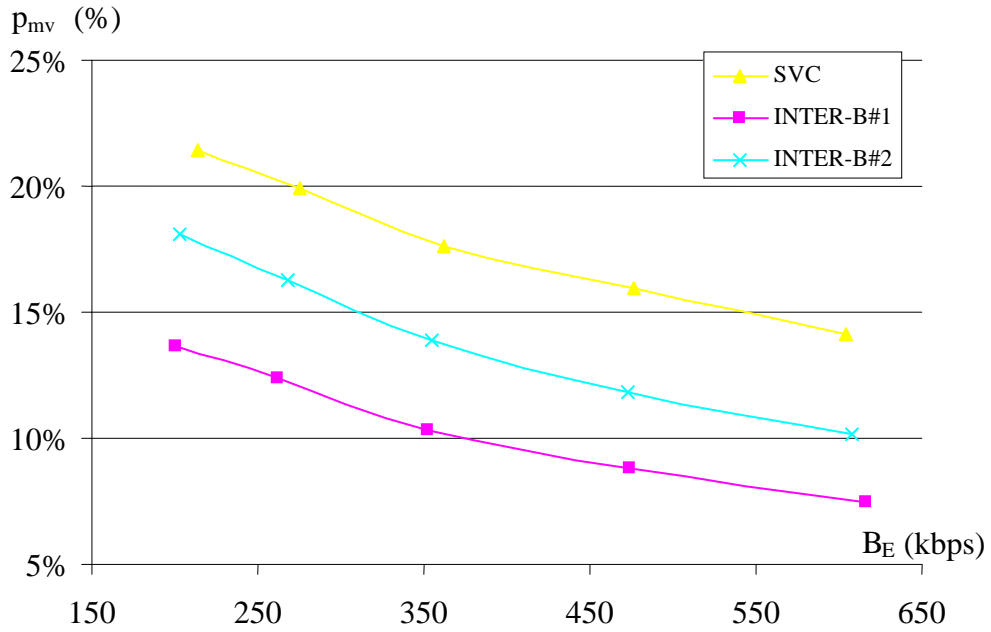


Fig. 8.13. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Football* sequence (352×288); SVC video codec.

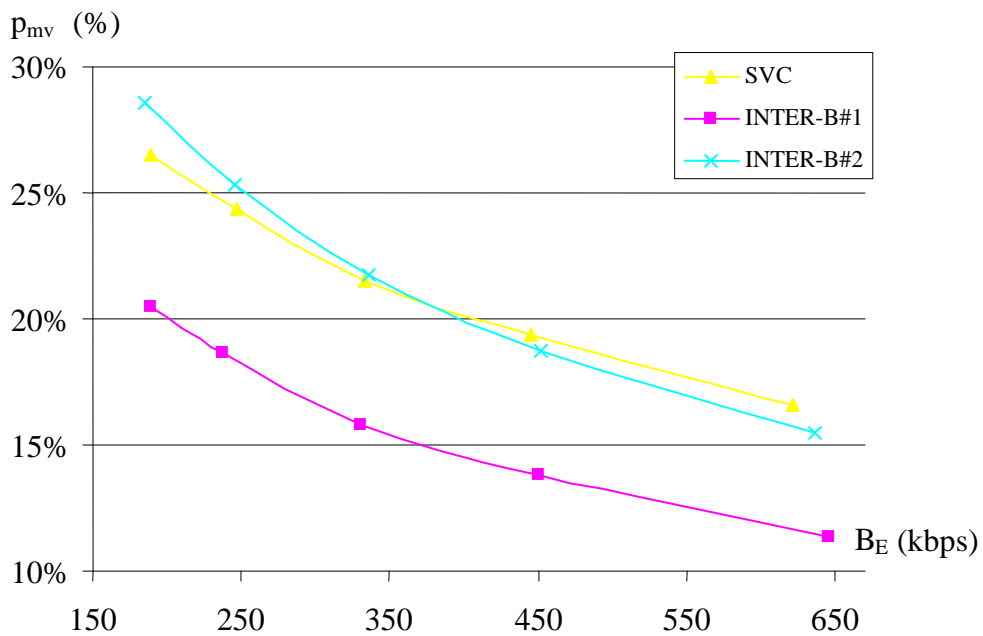


Fig. 8.14. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Foreman* sequence (352×288); SVC video codec.

Since no motion vector residuals were transmitted in B-frames when INTER-B#1 algorithm of motion vectors representation was used, the percentage of the motion vector residuals sub-bitstream in the overall bitstream was the lowest for this algorithm. In *Bus* and *Football* sequences, also INTER-B#2 techniques caused the decrease of percentage of motion vector residuals bitrate.

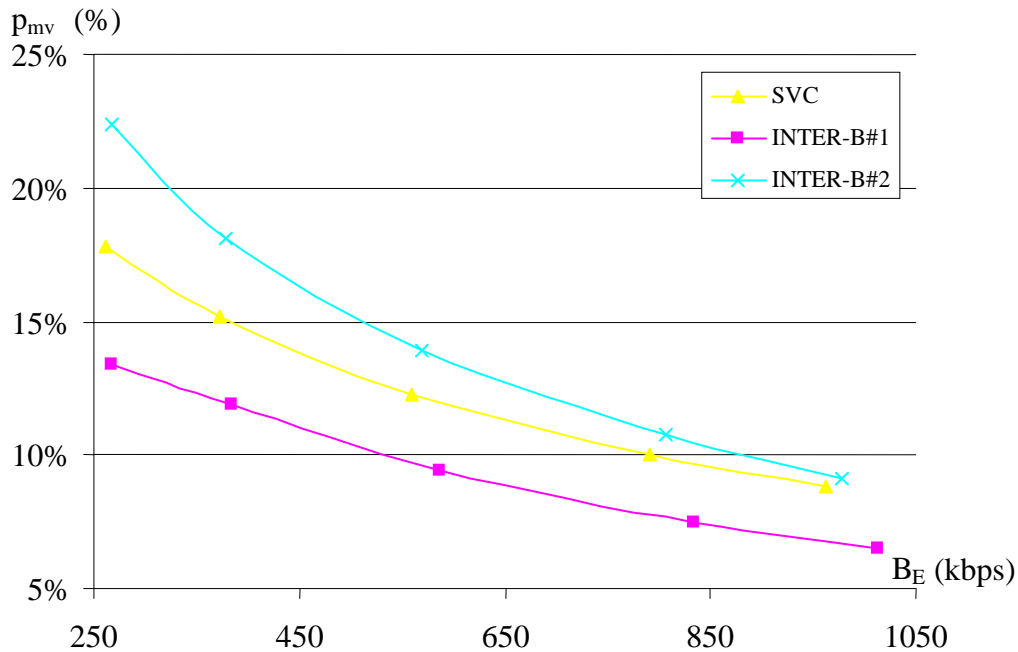


Fig. 8.15. The percentage of the bitrate of motion vector residuals sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Mobile* sequence (352×288); SVC video codec.

Surprisingly, in *Mobile* video sequence, the percentage of motion vector residuals sub-bitstream in the overall bitstream was the highest when INTER-B#2 algorithm was used. It means that many $\frac{1}{4}$ -pel refinements of motion vectors were present in a bitstream. This phenomenon is explained by the achieved percentage of transform coefficients bitrate in the overall bitrate for *Mobile* sequence, which is depicted in the following section.

8.4.3. The bitrate of transform coefficients sub-bitstream

In Fig. 8.16 – Fig. 8.19 the percentage of the bitrate of transform coefficients sub-bitstream p_{coeff} in the overall bitrate is shown for various methods of motion vectors coding in B-frames.

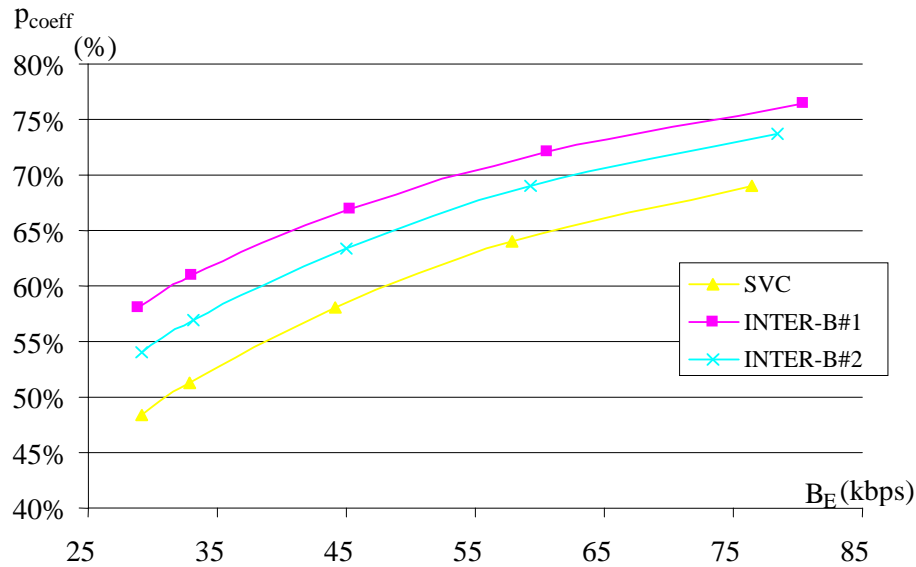


Fig. 8.16. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Bus* sequence (352×288); SVC video codec.

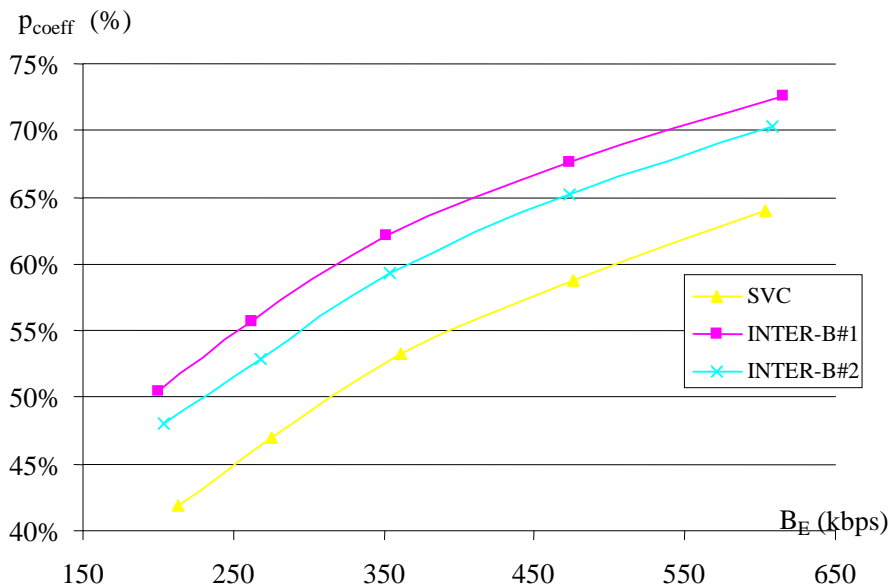


Fig. 8.17. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Football* sequence (352×288); SVC video codec.

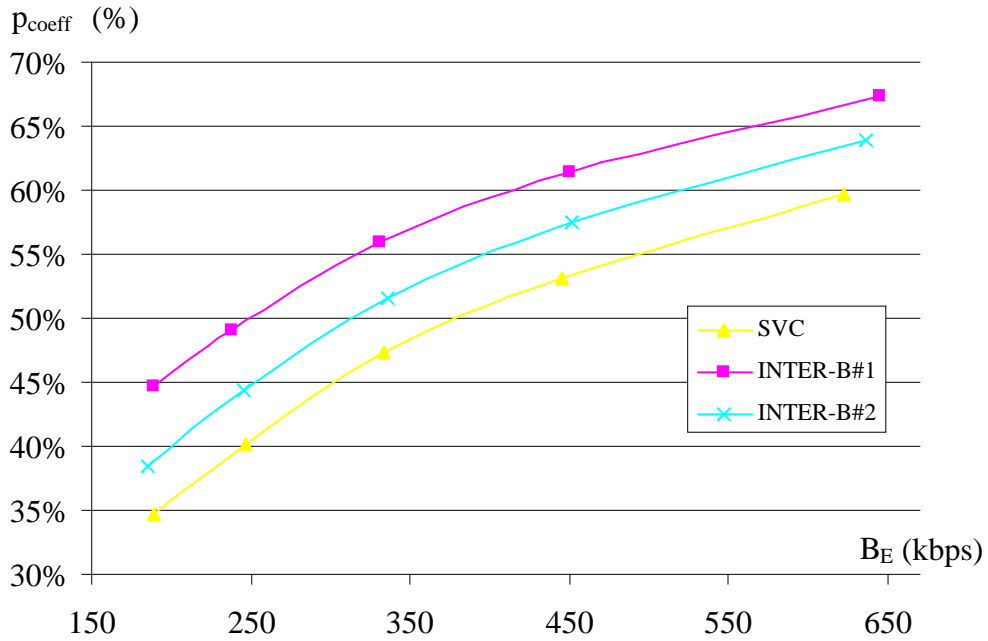


Fig. 8.18. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Foreman* sequence (352×288); SVC video codec.

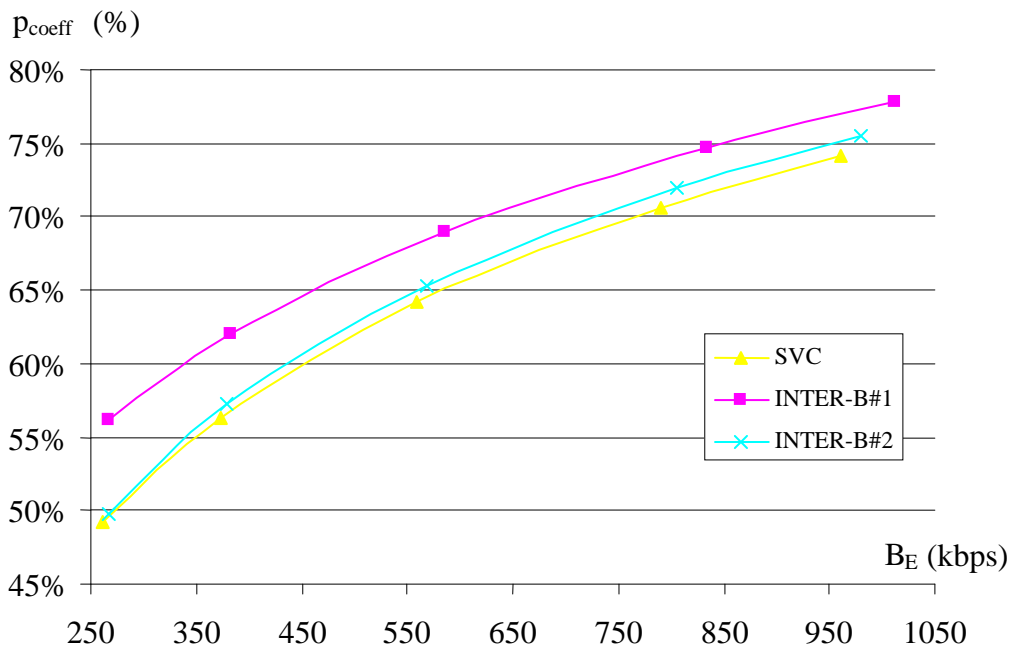


Fig. 8.19. The percentage of the bitrate of transform coefficients sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Mobile* sequence (352×288); SVC video codec.

In all cases, the percentage of transform coefficients sub-bitstream in the overall bitstream increases significantly when no motion vector residuals are transmitted in a bitstream (INTER-B#1 technique). Since the interpolated motion vectors from the base layer are less accurate, the encoder needs to send more transform coefficients in order to compensate less accurate prediction. However, when additional $\frac{1}{4}$ -pel refinements of the motion vectors were send (INTER-B#2 technique), the percentage of the transform coefficients sub-bitstream decreases.

In *Mobile* video sequence, the percentage of transform coefficients in the overall bitstream is almost the same for INTER-B#2 algorithm as for original SVC algorithm of motion vectors coding in B-frames. This is because the motion in *Mobile* sequence is rather slow and smooth. In this video sequence motion fields estimated for low resolution video and high resolution video are very similar (see Fig. 5.12, Fig. 5.24, Tab. 5.4 and Tab. 5.10). Therefore, interpolated motion vectors from the base layer that are refined with the $\frac{1}{4}$ -pel residual values allow for efficient motion-compensated prediction in the enhancement layer. It explains the phenomenon of the increase of the percentage of motion vector residuals in a bitstream (Fig. 8.15). Moreover, the decrease of average PSNR in B-frames as compared with standard SVC codec was also the lowest in *Mobile* video sequence (0.39dB in Tab. 8.6).

8.4.4. The bitrate of control data sub-bitstream

In SVC codec a significant part of a bitstream consist of control data. In this experiment, all syntax elements present in a bitstream except the transform coefficient residuals and motion vector residuals were considered to be the control data. In Fig. 8.20 – Fig. 8.23 the percentage of the bitrate of control data sub-bitstream $p_{control}$ in the overall bitrate is shown for various methods of motion vectors coding in B-frames.

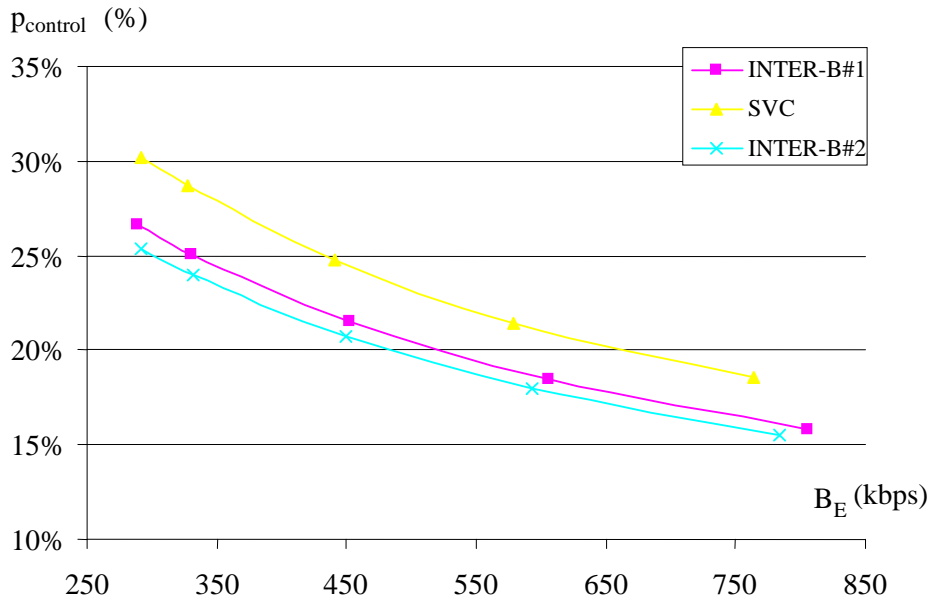


Fig. 8.20. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Bus* sequence (352×288); SVC video codec

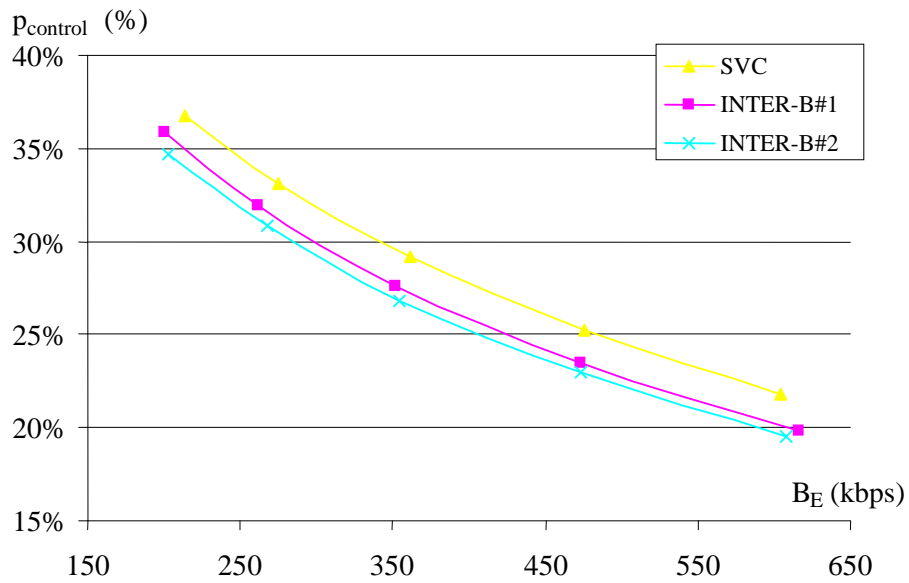


Fig. 8.21. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Football* sequence (352×288); SVC video codec.

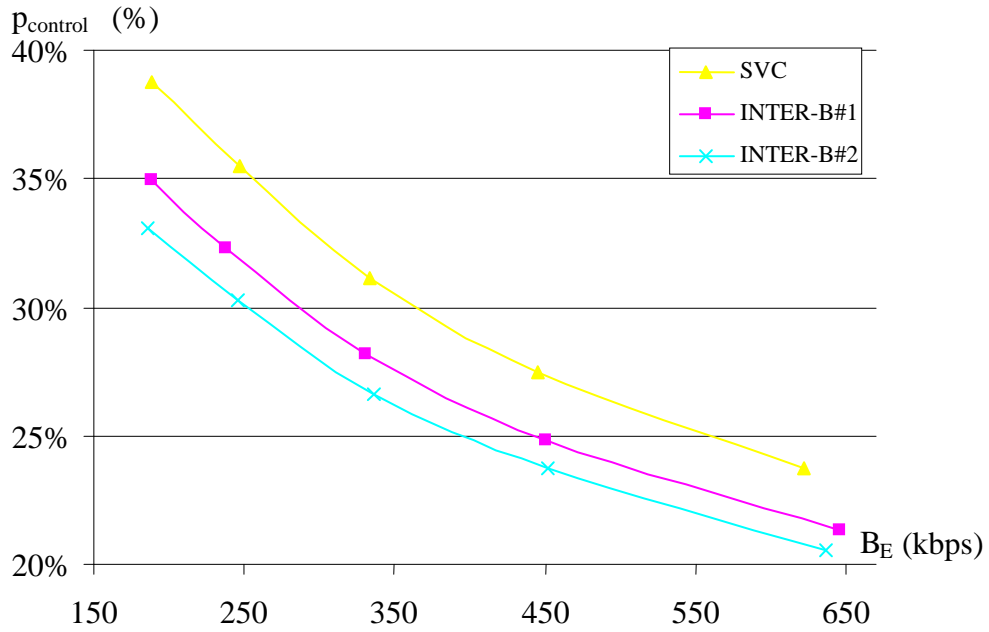


Fig. 8.22. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Foreman* sequence (352×288); SVC video codec.

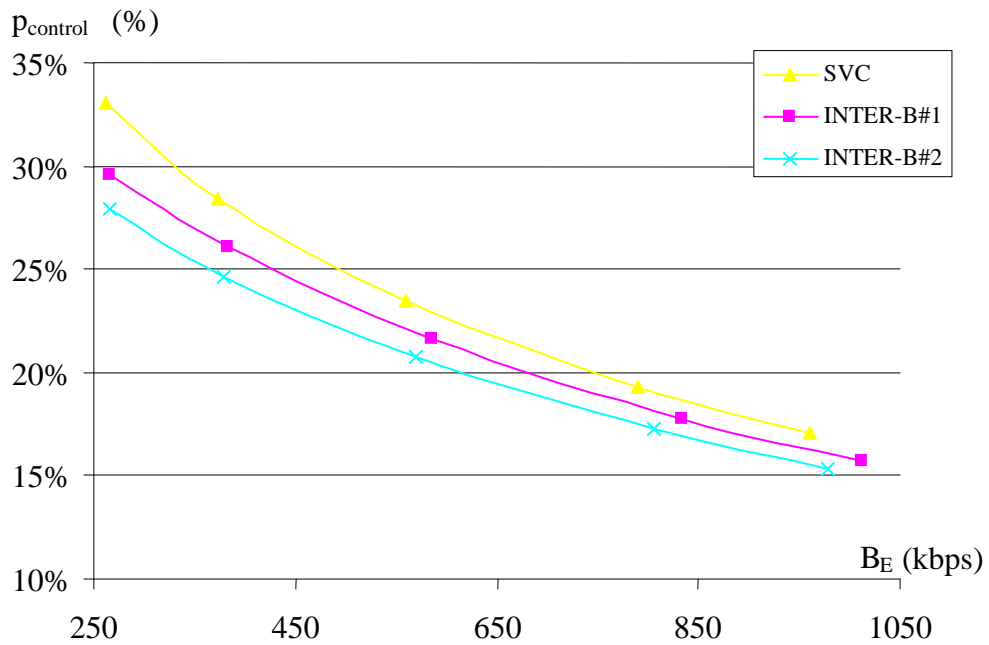


Fig. 8.23. The percentage of the bitrate of control data sub-bitstream in the overall bitrate for various techniques of motion vectors coding in B-frames in *Mobile* sequence (352×288); SVC video codec.

When joint multiresolution coding of the motion vectors in B-frames were applied, the percentage of control data sub-bitstream in the overall bitstream decreased at the cost of the increase of transform coefficients sub-bitstream percentage.

Interestingly, the bitrate of control data sub-bitstream in original SVC codec is in most cases higher than the bitrate of motion vector residuals sub-bitstream.

8.4.5. Subjective evaluation of the quality

In order to verify the results obtained in Section 8.4.1, the subjective quality evaluation of the encoded video sequences was performed for the proposed methods of motion vector representation in B-frames of the SVC codec. The Single Stimulus Multimedia (SSMM) method of subjective assessment of quality was applied [Bar04]. Because of economical reasons, only two video sequences were evaluated (*Bus* and *Mobile*). Each sequence was encoded with two different values of quantization parameter Q_p . The results of the subjective assessments are depicted in Fig. 8.24 and Fig. 8.25.

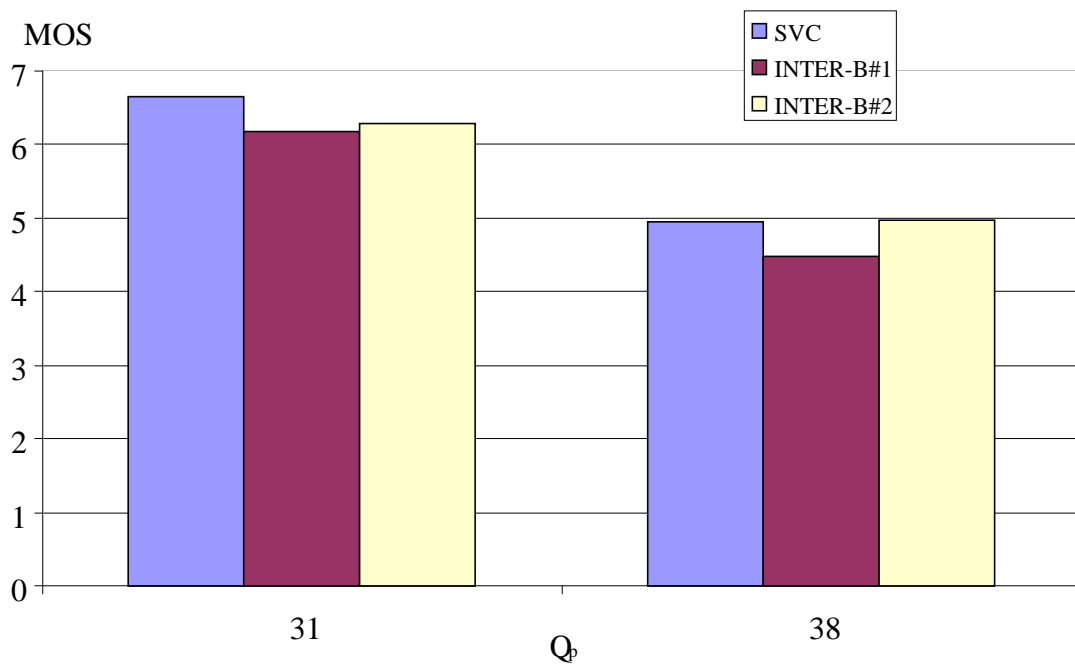


Fig. 8.24. The results of subjective assessment of the quality of *Bus* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality tests performed for *Bus* video sequence encoded with the value of quantization parameter $Q_p=31$, the highest value of Mean Opinion Score (MOS=6.65) was achieved for the original SVC scalable codec without any modifications. When the author's techniques of joint multiresolution coding of the motion vectors in B-frames were applied, the value of MOS was equal to 6.2 (INTER-B#1 algorithm) and equal to 6.3 (INTER-B#2 algorithm). In the tests performed for video sequence encoded with higher value of quantization parameter $Q_p=38$, Mean Opinion Score values were almost identical for the original SVC codec (MOS=4.94) and for the INTER-B#2 variant of joint coding of the motion vectors in B-frames (MOS=4.97).

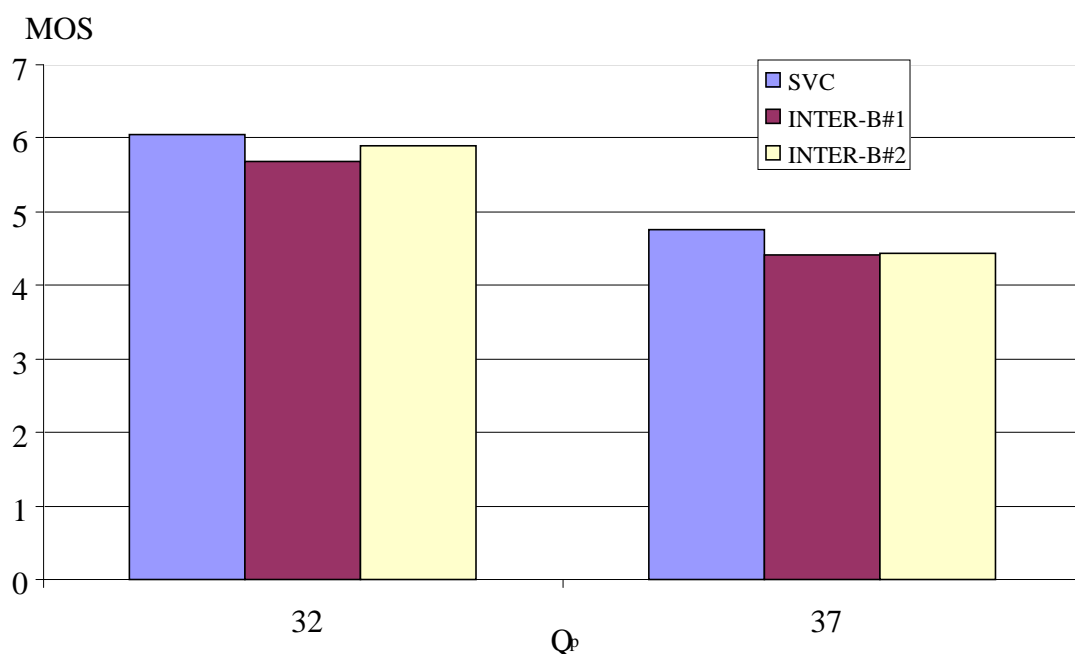


Fig. 8.25. The results of subjective assessment of the quality of *Mobile* sequence for various techniques of inter-layer prediction of motion vectors, enhancement layer (352×288). Results are given as mean opinion score.

In the subjective quality evaluation performed for *Mobile* video sequence, the highest values of MOS (MOS=6.05 for $Q_p=32$ and MOS=4.75 for $Q_p=37$) were achieved for the original SVC video codec. The codec using INTER-B#1 technique achieved the values of MOS equal to 5.7 and 4.4, and the codec using INTER-B#2 technique of joint multiresolution motion vector coding achieved the values of MOS equal to 5.9 and 4.45, respectively.

8.4.6. Complexity estimation

There has been made a comparison of complexity of the proposed solutions with the original SVC technique of motion vectors coding in B-frames. In Tab. 8.7 and Tab. 8.8 the execution time of encoding and decoding is given for various techniques of motion vector representation. Benchmarks of the encoder were performed by encoding of 100 frames of the sequences *Bus* and *City* with the constant quantization parameter $Q_P=32$. Benchmarks of the decoder were performed by decoding the bitstreams that contain 150 frames of the sequence *Bus* with the overall bitrates of about 400 kbps and about 1000 kbps. For higher accuracy, encoder and decoder executables were benchmarked three times and average execution times were calculated. All experiments were performed on a workstation with double Intel Xeon 3.6 GHz processors, 2GB of RAM and with Windows XP Professional installed.

Relative values of execution time have been calculated in reference to original SVC software.

Tab. 8.7. Execution time of encoder for various techniques of inter-layer prediction of motion vectors in SVC video codec

	CIF (~700kbps)		4CIF (~1500kbps)	
	average encoding time of one frame t_E (s)	relative to original SVC $\frac{t_E}{t_{E_{SVC}}} \cdot 100$ (%)	average encoding time of one frame t_E (s)	relative to original SVC $\frac{t_E}{t_{E_{SVC}}} \cdot 100$ (%)
SVC ($t_{E_{SVC}}$)	3.804	100	11.17	100
INTER-B#1	1.516	39.8	3.986	35.7
INTER-B#2	1.569	41.3	4.004	35.9

The proposed algorithms of motion vectors representation in B-frames reduce the time needed to perform the encoding operation. The difference in encoding time is huge: the encoder performs more than two times faster (for CIF sequence) and almost three times faster (for 4CIF sequence) when joint multiresolution coding of motion vectors is used in B-frames.

When either of the proposed algorithms is used, encoder does not need to perform motion estimation in B-frames for most of the macroblocks, since motion vectors are predicted from the the base layer. On the other hand, the R-D decision process for B-frames in the enhancement layer is much less complex: when macroblock in the base layer is coded using motion-compensated prediction, then prediction mode of the macroblock in enhancement layer is the same as the prediction mode of the macroblock in the base layer.

Tab. 8.8. Execution time of decoder for various techniques of inter-layer prediction of motion vectors in SVC video codec

	~400 kbps (CIF)		~1000kbps (CIF)	
	average decoding time of one frame t_D (ms)	relative to original SVC $\frac{t_D}{t_{D_{SVC}}} \cdot 100$ (%)	average decoding time of one frame t_D (ms)	relative to original SVC $\frac{t_D}{t_{D_{SVC}}} \cdot 100$ (%)
	SVC ($t_{D_{SVC}}$)	102.718	100	106.6
INTER-B#1	90.858	88.5	98.933	92.8
INTER-B#2	90.924	88.5	99.342	93.2

The decoding time is also shorter when joint multiresolution motion vector coding is used in the enhancement layer (Tab. 8.8). The decoder performed 11.5% faster for CIF video sequences and about 7% faster for 4CIF video sequence. The gain is achieved from simplified parsing of the bitstream for B-frames.

8.5. Conclusions

There has been proposed an original technique of joint multiresolution representation of motion vectors in temporally scalable video codec. The following two variants of motion vectors representation in B-frames have been implemented and experimentally tested:

- motion information prediction from the low-resolution layer without signaling of a macroblock prediction mode in a bitstream (referred to as **INTER-B#1**),

- motion information prediction from the low-resolution layer without signaling of a macroblock prediction mode in a bitstream and additional refinement of obtained motion vectors to $\frac{1}{4}$ -pixel accuracy (referred to as **INTER-B#2**).

In the proposed algorithms motion vectors in B-frames in the enhancement layer are not estimated. Instead, in order to perform motion-compensated prediction, interpolated motion vectors from the base layer are used.

There has been performed the comparison of the proposed algorithms against the original SVC technique of motion vectors coding in B-frames.

Since motion vectors are not estimated in B-frames in the enhancement layer, the average PSNR decreased in all cases when the author's techniques were used. When INTER-B#1 method of motion vectors encoding in B-frames was applied, the decrease of PSNR in B-frames varied from 0.29 dB in *Mobile* sequence (Fig. 8.11) up to 1.47 dB in *Football* and *Foreman* video sequences (Fig. 8.9 and Fig. 8.10 respectively). The average decrease of PSNR in B-frames was relatively large in all test sequences when INTER-B#1 method is used (0.87-1.01 dB, see Tab. 8.6).

When the additional $\frac{1}{4}$ -pel motion estimation was performed in order to increase the accuracy of the motion vectors in the enhancement layer (algorithm INTER-B#2), the decrease of PSNR in B-frames varied from 0.18 dB in *Mobile* sequence (Fig. 8.11) up to 1.15dB in *Football* video sequences (Fig. 8.9). The average decrease of PSNR in B-frames was the highest in *Football* video sequence (0.77dB). Since motion within *Football* sequence is fast and complex, the prediction of the motion vectors from low-resolution layer was not efficient. On the other hand, the lowest decrease of PSNR in B-frames appears in *Mobile* video sequence (0.39dB). Since motion in *Mobile* sequence is slow and smooth, the inter-layer motion prediction was very efficient.

These results are consistent with the results obtained in Chapter 5, where the highest correlation of the motion vectors estimated for different spatial resolutions were observed in video sequences with slow and smooth motion. On the other hand, in the case of fast and rough motion the correlation of motion vectors is lower, thus the inter-layer motion prediction is less efficient.

R-D curves from Fig. 8.4 – Fig. 8.7 show, that INTER-B#2 method introduces significantly lower degradation of average video quality than INTER-B#1 technique, as compared to the original SVC algorithm. The subjective comparisons of the efficiency of proposed algorithms (Fig. 8.24 and Fig. 8.25) confirm the results obtained using PSNR measure.

Importantly, the proposed algorithms do not affect the efficiency of encoding of I-frames and P-frames. The distortions in these frames are constant regardless used algorithm (INTER-B#1, INTER-B#2 or SVC).

A huge advantage of proposed techniques is a significant decrease of complexity of the codec: the encoder performs 60%-65% faster when INTER-B#1 method is used and 58%-64% faster when INTER-B#2 method is used (Tab. 8.8). Also the decoder performs 7%-11% faster when INTER-B#1 method is used and 6%-11% faster when INTER-B#2 method is used (Tab. 8.9).

The differences in execution time of the encoder and the decoder are enormous as compared with SVC codec. It is very important that there is no loss of the quality in I-frames and P-frames. The decrease of quality appears only in B-frames, which are not used as reference frames. Moreover, B-frames are used in SVC codec in order to achieve temporal scalability, as described in Section 8.1. Thus, when B-frames are dropped in spatially enhancement layer, they do not impact the quality of remaining frames.

Since the efficiency of encoding of I-frames and P-frames is not affected and the encoding time for B-frames is significantly reduced, the author's proposal of joint multiresolution coding of motion vectors in B-frames allows for achieving complexity scalability. Low-complex profile of B-frames coding can be used in order to encode a video sequence with doubled temporal resolution.

There have been also examined the impact of the proposed techniques of B-frames coding on the content of the bitstream. Joint multiresolution coding of motion vectors in B-frames results in a decrease of a control data percentage in the overall bitstream, as depicted in Fig. 8.20 – Fig. 8.23. The decrease varies from 2% (Fig. 8.21, *Football*, INTER-B#2) to 5% (Fig. 8.22, *Foreman*, INTER-B#1).

On the other hand, the portion of motion vector residuals in a bitstream is the highest when standard SVC coding of B-frames is applied (Fig. 8.12 – Fig. 8.14), except *Mobile* sequence, for which, the highest percentage of motion vector residuals is achieved using INTER-B#2 algorithm.

The decrease of motion vector residuals sub-bitstream is balanced by the increase of transform coefficients sub-bitstream. In all cases the highest percentage of transform coefficients sub-bitstream in the overall bitstream was achieved when INTER-B#1 algorithm was used: lower efficiency of the motion-compensated prediction needed to be compensated by additional transform coefficients transmitted in a bitstream.

Summarizing, the largest part of a bitstream makes up transform coefficients (35%-78%), motion vector residuals make up 7%-28% and control data make up 15%-36% of the overall bitstream.

8.6. Summary

The proposed algorithms of joint multiresolution motion vectors coding in bi-directionally, non-referenced pictures allow for simple and very fast encoding of B-frames in the enhancement layer. The encoder does not need to perform motion estimation and mode decision for each macroblock:

1. When a co-located macroblock from the base layer is coded using motion compensated-prediction, then their motion vectors are rescaled and re-use in the enhancement layer. Additionally, $\frac{1}{4}$ -pel refinement is applied to interpolated motion vectors when the second proposed algorithm (INTER-B#2) is used.
2. When a co-located macroblock from the base layer is coded using intra prediction, then intra prediction is applied in the enhancement layer as well.

The decrease of quality in B-frames is acceptable, especially when INTER-B#2 algorithm is used. The results of experimental tests proved that refining the accuracy of motion vectors to $\frac{1}{4}$ -pel is not complex and can be performed very fast, using existing tools available in standard SVC encoder. Therefore, the proposed solution provides complexity scalability for SVC codec. Joint multiresolution motion vectors coding in B-frames allows for an easy increase of the temporal resolution of the transmitted video sequence in the enhancement layer with almost no impact on the encoder's complexity.

Chapter 9.

Results and conclusions

9.1. Results

The author focused his efforts on developing new, efficient methods of representation of motion vectors in scalable video coding with spatial and temporal scalability. The goal was to find either algorithms that result in better compression efficiency or algorithms with lower complexity and comparable compression efficiency to the ones offered by existing techniques of motion vectors coding in applications to multiresolution video representation.

- At the beginning of this dissertation, a problem of multiresolution representation of motion vectors in video compression has been formulated. The problem regards motion vector estimation and motion vector coding in a scalable video codec.

In order to develop an efficient solution for motion representation in scalable video coding, series of preliminary studies and experiments have been performed, including:

- Review of techniques of multiresolution video representation – scalable video coding (Chapter 2);
- Review of algorithms used in motion-compensated prediction (Chapter 3);
- Studies of existing techniques of motion vectors representation in non-scalable codecs (Chapter 3);
- Experimental investigations of the efficiency of the most advanced techniques of motion vector representation in non-scalable codecs (Chapter 4);
- Studies and investigations of the efficiency of entropy coding of motion vector residuals using Exp-Golomb codes and CABAC (Chapter 4);
- Studies and experiments in order to improve motion vector prediction efficiency by employing methods used in signal processing (vector median filtering – Chapter 4);
- Experimental investigations of multiresolution motion vector fields with a special focus on the correlation between motion vectors estimated independently for various spatial resolutions of the same video sequence (Chapter 5).

The considered techniques of motion vectors coding, based on component-wise prediction using spatially adjacent blocks, proved to be very efficient. However, high correlation between motion vectors estimated for low-resolution video and motion vectors estimated for high-resolution video, which has been revealed in Chapter 5,

encouraged the author to exploit this correlation in order to improve further the compression efficiency.

In the first proposal of multiresolution motion vector representation, described in Chapter 6, further increase of the correlation between motion vectors from a low-resolution layer (the base layer) and high-resolution layer (the enhancement layer) has been achieved by motion estimation based on optical flow technique. Obtained motion vectors were coded as follows:

- in the base layer, motion vectors were spatially predicted according to the AVC/H.264 algorithm,
- in the enhancement layer, motion vectors were jointly coded together with motion vectors from the base layer, using differential scheme with median prediction of residual values.

Unfortunately, the first proposal has not improved the overall coding efficiency. On the one hand, worse motion-compensated prediction of samples increased prediction residuals, on the other hand the proposed method of joint motion vector encoding in enhancement layer did not outperform the original AVC/H.264 prediction scheme.

Consequently, a new method of multiresolution motion vectors coding, called Implicit Inter-Layer Prediction (IILP), has been developed by the author of this dissertation. During designing of the IILP technique the following assumptions were made:

- the performance of a motion-compensated prediction should not be affected, thus no modification into motion estimation algorithm should be introduced,
- existing implicit correlations between motion vectors from the base layer and motion vectors from the enhancement layer should be exploited,
- the efficiency of current technique of motion vector coding should not be affected, thus new algorithm should extend the existing technique (median, component-wise prediction),

IILP method of inter-layer motion vectors prediction, presented in Chapter 7, fulfils the above mentioned expectations. In this proposal, motion vectors from the base layer are exploited when the actual context for the currently coded motion vector in the enhancement layer is not satisfactory in the sense of motion vector prediction. In these cases, the co-located motion vector from the base layer is used.

The IILP algorithm developed by the author, improves the efficiency of motion vectors prediction up to 30%. A decrease of the average motion vector prediction residual causes an improvement of a total compression efficiency. Objective distortions measure – PSNR – gains up to 0.2 dB for a given bitrate. Also values of objective quality measure – MOS – were the highest when IILP algorithm was used in a scalable video codec.

It is worth noticing that IILP technique was one of the first successful algorithms of inter-layer motion vector prediction, presented for scalable video codec based on Advanced Video Coding technology proposed for standardization.

The proposed method has been experimentally compared with another technique of inter-layer prediction of motion vectors, which was used in SVC codec – the scalable video codec developed by MPEG committee. The compression efficiency achieved by the encoder that used IILP approach was a little worse than compression efficiency of the original SVC algorithm. However, the author's approach is significantly less complex and performs about 40% faster than the standard SVC solution.

Another original technique of multiresolution motion vector representation has been presented in Chapter 8. The technique exploits similarities between motion vector fields estimated for low-resolution video and high-resolution video in temporally scalable codec. The presented modifications are based on high correlations between co-located motion vectors in the video frames coded with bidirectional motion-compensated prediction (B-frames). The author modified the syntax and the semantics of SVC bitstream for B-frames in order to achieve very fast and still efficient method of temporal scalability.

The new proposed algorithm of joint multiresolution coding of motion vectors in B-frames speeds up SVC encoder up to 65% and speeds up SVC decoder up to 12% with the reported decrease of PSNR in B-frames of 0.39dB – 0.77dB, dependent on the content of a video sequence. The quality of I-frames and P-frames remains unchanged. The proposal introduces the complexity scalability feature into SVC codec.

Summarizing, the following main original results have been achieved and described in this thesis:

- Estimation of the efficiency of existing techniques of non-scalable motion vectors coding (Section 4.2);
- Comparison of the efficiency and complexity of component-wise motion vector prediction versus vector median motion vector prediction (Section 4.3);

- Comparison of the efficiency of entropy coding using Exp-Golomb codes versus arithmetic coding using CABAC algorithm (Section 4.4);
- Estimation of correlation between multiresolution motion vector fields, obtained by independent motion estimation (Section 5.4);
- Development of joint motion estimation for various spatial resolutions in scalable codec using optical flow technique (Section 6.3.1);
- Development of joint, differential coding of motion vectors in the enhancement layer of scalable video codec using motion vectors from the base layer (Section 6.3.2);

The most important achievements of the author that have been presented in this dissertation are:

- Development of IILP technique – an original, very fast method of inter-layer motion vector prediction developed for AVC-based scalable video codec (Section 7.3);
- The experimental comparison of the proposed IILP technique against the technique originally used in SVC codec (Section 7.8);
- Proposal of the original algorithm of motion vectors representation in a temporally scalable codec; the proposed solution is extremely fast as compared with existing techniques (Section 8.2).

It has been proven that the proposed inter-layer motion vector prediction improves the efficiency of motion vectors coding in a scalable video codec. Exploiting motion vectors from a low-resolution layer while encoding motion vectors in high-resolution layer decreases average motion vector prediction residual and yields in the overall compression efficiency. Algorithms proposed in this dissertation are very fast and can be successfully applied in real-time scalable video codecs.

9.2. Conclusions

The thesis of this dissertation states that there exist correlations between motion vectors estimated for different resolutions of the same video sequence. Exploiting these correlations should allow for improving the performance of motion vectors coding and also improving the overall compression efficiency of a scalable codec.

Preliminary studies proved that the existing techniques of motion vector representation in advanced, non-scalable hybrid codecs are very efficient. Experimental results lead to the following conclusions:

- The complex motion model used in advanced video coding results in heterogeneous motion vector field that has to be compressed in order to transmit to a decoder;
- The median component-wise prediction minimizes motion vector prediction residual in most cases; a lot of the prediction residuals (38%-80%) have the values of zero;
- The median prediction performs extremely efficiently in video sequences containing slow and moderately slow motion;
- There is no significant difference in efficiency of the component-wise median prediction and the vector median prediction of motion vectors;
- The context-based arithmetic entropy coding of motion vector residuals is definitely more efficient than Exp-Golomb coding in video sequences with fast and complex motion;
- There is implicit correlation between motion vectors estimated in the low-resolution version of a video sequence and motion vectors estimated from the high-resolution version of the same video sequence, even when independent motion estimations are performed.

Very efficient techniques of representation of motion vectors in hybrid video coders have been developed over the years. As a result, it is not easy to further improve motion vector compression in scalable video codecs. However, several techniques were proposed by the author, and they have been presented in this dissertation.

The performance of the firstly proposed joint motion estimation and joint coding of motion vectors in layered scalable codec was not satisfactory. However, another technique developed by the author – IILP, which involves implicit inter-layer motion prediction, brought improvements in a motion vector coding and overall compression efficiency. Joint motion vector coding in B-frames in a temporally scalable codec significantly simplify encoding process and speeds up the codec with a minor impact on the compression efficiency.

The following conclusions were drawn:

- In a scalable video coder, the best compression efficiency is achieved when independent rate-distortion optimization is used in each layer. Independent R-D optimization forces independent motion estimation for each spatial resolution;
- Interpolated motion vectors from the base layer have lower accuracy, thus their usefulness in the enhancement layer is limited. However, $\frac{1}{4}$ -pel refinement of interpolated motion vectors significantly improves the efficiency of motion-compensated prediction;
- Since median-based motion prediction produces very low residuals, the existing intra-layer techniques should be replaced only in some specific cases, when achieved results are not satisfactory;
- In the cases when some of adjacent motion vectors are not available, inter-layer motion vectors prediction leads to good results. It significantly decreases motion vector prediction residual and improves overall coding efficiency.

A comparison of the author's IILP method with the standard technique, in which inter-layer motion prediction mode is explicitly signaled in a bitstream (Section 7.8) leads to a conclusion that the best results are obtained with a more parameterized motion model. Therefore, in modern video coders, control data (including motion parameters) constitutes a significant part of a bitstream. An important issue is to find an appropriate trade-off between transmitted parameters of a video model and the actual residual data like motion vectors and transform coefficients.

SVC video coding standard is about to be established. Its performance is going to be comparable with state-of-the-art AVC/H.264 codec without scalability. The results presented in this dissertation were contributed to MPEG during the process of standardization. The author hopes that his work was helpful and useful.

Bibliography

Author's contributions

- [Bła04a] Błaszak Ł., Domański M., Lange R., Maćkowiak S., "Scalability of Modified AVC/H.264 Video Codecs", 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, Lisboa, Portugal, 2004.
- [Bła04b] Błaszak Ł., Domański M., Lange R., Łuczak A., "Scalable AVC Codec", ISO/IEC JTC1/SC29/WG11 MPEG04 Doc. M10626, Munich, March 2004.
- [Bła04c] Błaszak Ł., Domański M., Lange R., "Modified AVC Codec with SNR Scalability Based on Macroblock Hierarchy", Proc. of 11th International Workshop on Systems, Signals and Image Processing, IWSSIP'2004, Poznań, September 2004.
- [Bła04d] Błaszak Ł., Domański M., Lange R., Łuczak A., "Response to SVC CE2 Tasks: Testing of SNR Scalability Technology", ISO/IEC JTC1/SC29/WG11 Doc. MPEG04/M11101, Redmond, July 2004.
- [Bła04e] Błaszak Ł., Domański M., Lange R., "SVC CE6 Results at Poznań University of Technology", ISO/IEC JTC1/SC29/WG11 Doc. MPEG04/M11418, Palma, October 2004.
- [Bła04f] Błaszak Ł., Lange R., „Skalowalny koder AVC z ulepszoną predykcją wektorów ruchu”, Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2004, Warsaw, June 2004.
- [Dom04] Domański M., Błaszak Ł., Lange R., Łuczak A., "Propozycje skalowalnego kodera obrazu MPEG", Przegląd Telekomunikacyjny, Vol. LXXIII, No.7, pp. 303-306, July 2004.
- [Lan03] Lange R., Domański M., "Motion Vector Encoding in AVC Spatially-scalable Video Coders", Recent Trends in Multimedia Information Processing, pp. 111-114, Czech Technical University in Prague, Prague, 2003.
- [Lan03a] Lange R., „Estymacja i kodowanie wektorów ruchu w skalowalnych koderach wizyjnych”, Master of Science thesis, Poznań University of Technology, Poznań, 2003.
- [Lan04] Lange R., Domański M., "Enhanced Inter-layer Prediction of Motion Vectors in AVC-based Scalable Video Coder", 11th International Workshop on Systems, Signals And Image Processing, IWSSIP'2004, Poznań, September 2004.

- [Lan04a] Lange R., Domański M., "Representation of Motion Vectors in Modified Advanced Video Coding with Spatial Scalability", Proc. of 12th European Signal Processing Conference EUSIPCO, Vienna, Austria, September 2004.
- [Lan04b] Lange R., Błaszak Ł., Domański M.: "Simple AVC-Based Codecs with Spatial Scalability", Proc. of IEEE International Conference on Image Processing ICIP 2004, Singapore, October 2004.
- [Lan04c] Lange R., Domański M., "Response to SVC CE2 Tasks: Inter-layer Motion Information Prediction", ISO/IEC JTC1/SC29/WG11 Doc. MPEG04/M11101, Redmond, July 2004.
- [Lan05] Lange R., Łuczak A., „Zaawansowane narzędzia kompresji sekwencji wizyjnych w koderze standardu AVC/H.264”, Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2005, Kraków, June, 2005.
- [Lan06] Lange R., "Extended Inter-layer Motion Vectors Prediction in Scalable Video Coding – Case Study and Improvement Proposal", ISO/IEC JTC1/SC29/WG11 MPEG06 Doc. M12822, Bangkok, January, 2006.
- [Lan06a] Lange R., Domański M., "Investigation on Vector Median Prediction of Motion Vectors in SVC", ITU-T SG16 Q.6 Doc. JVT-S062, Geneva, April 2006.
- [Lan06b] Lange R., Domański M., "Extended Inter-layer Motion Vectors Prediction in JSVM – Case Study and Experimental Results", ITU-T SG16 Q.6 Doc. JVT-S063, Geneva, April 2006.
- [Lan06c] Lange R., Domański M., „Skalowalne kodowanie sekwencji wizyjnych w zastosowaniach dla bezprzewodowych i heterogenicznych sieci telekomunikacyjnych– prace badawcze w ramach projektu polsko-flandryjskiego", Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2006, Poznań June 2006.
- [Lan06d] Lange R., "Wielorozdzielczościowe kodowanie wektorów ruchu z predykcją przestrzenną i międzywarstową", Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT 2006, Poznań, June 2006.
- [Lan06e] Lange R., "Eksperymentalne porównanie wariantów kodowania wektorów ruchu w zaawansowanym koderze wizyjnym", V Sympozjum Naukowe Techniki Przetwarzania Obrazu, Serock, November 2006.

Other references used in the dissertation

- [Abh03] Abhayaratne C.G.K., Heijmans H.J.A.M., “A Novel Morphological Subband Decomposition Scheme for 2D+t Wavelet Video Coding”, Proc. of the 3rd International Symposium on Image and Signal Processing and Analysis, ISPA 2003, Vol. 1, pp.:239 – 244, September 2003.
- [Ahm74] Ahmed N., Natarajan T., Rao K.R., “On Image Processing and a Discrete Cosine Transform”, IEEE Transactions on Computers, Vol. C-23, No. 1, pp.90-93, January 1974.
- [Arg91] Argenti F., Barni M., Cappellini V., Mecocci A., “Vector Median Deblurring Filter for Colour Image Restoration”, Electronics Letters, Vol. 27, Issue 21, pp. 1899 – 1900, October 1991.
- [Ast90] Astola J., Haavisto P., Neuvo Y., “Vector Median Filters”, Proc. IEEE, Vol. 78, No. 4, pp. 678-689, April 1990.
- [AVS06] Audio Video Coding Standard Working Group of China, The Standards of People’s Republic of China GB/T 20090.2-2006, “Information Technology - Advanced Coding of Audio and Video - Part 2:Video”, 2006.
- [Bar04] Baronchini V., Oelbaum T., „Subjective Test Results for the CfP on Scalable Video Coding Technology“, ISO/IEC JTC1/SC29/WG11 MPEG 04 Doc. M10737, Munich, Germany, March 2004.
- [Bar04a] Barbarien J., Munteanu A., Verdicchio F., Andreopoulos Y., Cornelis J., Schelkens P., "Scalable Motion Vector Coding", Proc. of IEEE International Conference on Image Processing ICIP 2004, Singapore, October 2004.
- [Bar94] Barron J.L., Fleet D.J., Beauchemin S.S., "Performance of optical flow techniques", International Journal of Computer Vision Vol. 12, Issue 1, pp. 43-77, February 1994.
- [Bar97] Bardos A.J., Sangwine S.J., “Selective Vector Median Filtering of Colour Images”, Sixth International Conference on Image Processing and Its Applications, Vol. 2, pp. 708 – 711, July 1997.
- [Bla98] Blasiak D., Chan W.-Y., “Efficient Wavelet Coding of Motion Compensated Prediction Residuals”, Proc. International Conference on Image Processing, ICIP’98, Vol. 2, pp. 287-290, Chicago, 1998.
- [Bła03] Błaszak Ł., Domański M., Maćkowiak S., “Spatio-temporal Scalability in AVC Codecs”, ISO/IEC/SC29/WG11 Doc. M9469, March 2003.
- [Bła06] Błaszak Ł., “Advanced Scalable Hybrid Video Coding”, Doctoral Thesis, Poznań University of Technology, Poznań 2006.

- [Bur83] Burt P., Adelson E., "The Laplacian Pyramid as a Compact Image Code", IEEE Transactions on Communications, Vol. 31, Issue 4, pp.: 532 – 540, April 1983.
- [Can71] Candy J. C., Franke M. A., Haskell B. G., Mounts F. W., "Transmitting Television as Clusters of Frame-to-Frame Differences", Bell System Technical Journal, Vol. 50, No. 6, pp. 1889-1919, July-August 1971.
- [CCI84] "Description of two of the Video Codecs used in the United States with Primary Rate Transmission Systems", Contribution #4, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, November 1984.
- [CCI87] "Hybrid Coding at 64Kbits/s and its Compatibility with 384Kbits/s". Contribution #234, Specialist Group on Coding for Visual Telephony, Study Group XV, CCITT, May 1987.
- [Ced06] "Memories of VideoDisc - Who's Who in VideoDisc", <http://www.cedmagic.com/mem/whos-who/kell-ray.html>, August 2006.
- [Cha90] Chan M.H., Yu Y.B., Constantinides A.G., "Variable Size Block Matching Motion Compensation with Applications to Video Coding", Proc. IEE Communications, Speech and Vision, Vol. 137, Issue 4, pp. 205-212, August 1990.
- [Che01] Chen M.-J., Chu M.-C., Lo S.-Y., "Motion vector composition algorithm for spatial scalability in compressed video", IEEE Transactions on Consumer Electronics, Vol. 47, Issue 3, pp.:319 – 325, August 2001.
- [Che02] Cheng H.-Y., Tourapis A.M., Topiwala P., "Fast Motion Estimation within the JVT codec", ITU-T SG16 Q.6, Doc. JVT-E023, Geneva, October, 2002.
- [Che05] Chen Y.-H., Chen C.-Y., Chen L.-G., "Architecture of Global Motion Compensation for MPEG-4 Advanced Simple Profile", IEEE International Symposium on Circuits and Systems, ISCAS 2005, Vol. 2, pp. 1798 - 1801, May 2005.
- [Che93] Chen C. T., "Video Compression: Standards and Applications", Visual Communication and Image Representation, No. 2, pp. 103-111, June 1993.
- [Cho05] Cho-Chun Ch., Wen-Liang H., Zuowei S., Tao X., "Advanced motion compensation techniques for blocking artifacts reduction in 3-D video coding systems", Proc. of International Conference on Image Processing ICIP05, Vol. 3, pp. 209-212, September 2005.

- [Cho99] Choi S.-J., Woods J.W., "Motion Compensated 3-D Subband Coding of Video", IEEE Transactions on Image Processing, Vol. 8, pp. 155-167, February 1999.
- [Chu94] Chung-Lin H., Chao-Yuen H., "A new motion compensation method for image sequence coding using hierarchical grid interpolation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, No. 1, pp. 42-52, February 1994.
- [Con97] Conklin G. J., Hemami S. S., "Multi-resolution Motion Estimation", Proc. International Conference on Acoustics, Speech and Signal Processing, Vol. 4, pp.: 2873-2877, Munich, Germany, April 1997.
- [Dho05] Dhondt Y., Lambert P., Notebaert S., Van de Walle R., "Flexible Macroblock Ordering as a Content Adaptation Tool in H.264/AVC", Proc. of the SPIE Optics East conference. 2005. pp. 9, Boston, October, 2005.
- [Di03] Di W., Wen G., Mingzeng H., Zhenzhou J., "An Exp-Golomb Encoder and Decoder Architecture for JVT/AVS", Proc. 5th International Conference on ASIC, Vol. 2, pp.:910-913, October 2003.
- [Dom00] Domański M., Łuczak A., Maćkowiak S., "Spatio-Temporal Scalability for MPEG", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10 No. 7, pp. 1088-1093, October 2000.
- [Dom03] Domański M., Błaszak Ł., Maćkowiak S., "AVC Video Coders with Spatial and Temporal Scalability," Picture Coding Symposium PCS'03, Saint Malo, France, April 2003.
- [Dom05] Domański M., Bartkowiak M., "Multimedia - Przełom Technologiczny", Przegląd Telekomunikacyjny, Vol. LXXVIII, No. 6, pp.209-210, June 2005.
- [Dom98] Domański M., "Zaawansowane Techniki Kompresji Obrazów i Sekwencji Wizyjnych", Wydawnictwo Politechniki Poznańskiej, Poznań 1998.
- [Dom99] Domański M., Łuczak A., Maćkowiak S., Świerczyński R., Benzler U., "Spatio-Temporal Scalable Video Codecs with MPEG-Compatible Base Layer", Proc. of Picture Coding Symposium, pp.45-48, Portland 1999.
- [Dzi05] Dziecielewski T., Grajek T., Marek J., „Eksperymentalna Analiza Częstotliwości Wyboru Trybów Zaawansowanego Kodowania Wizyjnego”, Proc. X Poznańskie Warsztaty Telekomunikacyjne, pp. 115-120, Poznań 2005.
- [Eno65] Enomoto H., Shibata K., "Features of Hadamard Transformed Television Signal", National Conference of The Institute for Engineers in Communications and Electronics, Japan, Paper 881, 1965.

- [Eri85] Ericsson S., "Fixed and Adaptive Predictors for Hybrid Predictive/Transform Coding", IEEE Transactions on Communications Vol. COM-33, No. 12, pp.1291-1302, December 1985.
- [Fan04] Fan L., Ma S., Wu F., "Overview of AVS video standard", Trans. IEEE International Conference on Multimedia and Expo ICME '04, Vol. 1, pp. 423-426, June 2004.
- [Fli03] Flierl M, Girod B., "Video Coding with Motion-compensated Lifted Wavelet Transforms," Proc. of Picture Coding Symposium, PCS03, pp. 59-62, Saint-Malo, France, April 2003.
- [Fli04] Flierl M., Girod B., "Video Coding With Superimposed Motion-Compensated Signals: Applications to H.264 and Beyond", Kluwer Academic Publishers, Dordrecht 2004.
- [Gal01] Gallant M., Kossentini F., "Rate-distortion Optimized Layered Coding with Unequal Error Protection for Robust Internet Video", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, Issue 3, pp. 357 - 372, March 2001 .
- [Gha99] Ghanbari M., "Video Coding - an Introduction to Standard Codecs", The Institution of Electrical Engineers, Stevenage 1999.
- [Gir87] Girod B., "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences", IEEE Journal on Selected Areas in Communications, Vol. 5, Issue 7, pp. 1140-1154, August, 1987.
- [Gir93] Girod B., "Motion-compensating Prediction with Fractional-pel Accuracy", IEEE Transactions on Communications, Vol. 41, Issue 4, pp.604 – 612, April 1993.
- [Gir95] Girod B., Horn U., Belzer B., "Scalable Video Coding with Multiscale Motion Compensation and Unequal Error Protection," Proc. International Symposium on Multimedia Communications and Video Coding, New York, pp. 475–482, October 1995.
- [Got93] Gothe, M., Vaisey, J., "Improving Motion Compensation Using Multiple Temporal Frames", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Vol. 1, pp.:157 - 160, 19-21 May 1993.
- [Gu99] Gu W.-C., Lin D. W., "Joint Rate-Distortion Coding of Multiple Videos," IEEE Trans. on Consumer Electronics, Vol. 45, no. 1, pp. 159-164, February 1999.
- [He01] He Y., Yan R., Wu F., Li S., "H.26L-based Fine Granularity Scalable Video Coding", ISO/IEC JTC1/SC29/WG11 Doc. MPEG01/M7788, Pataya, December 2001.

- [Hel96] Held G., "Data and Image Compression", John Wiley & Sons Ltd., Chichester, 1996.
- [Hid89] Hidaka T., "Description of the Proposing Algorithm and its Score for Moving Image (A Part of the Proposal Package)", ISO/IEC JTC1/SC2/WG8, Doc. MPEG 89/188, October 1989.
- [Hoa02] Hoang D. T., Vitter J. S., "Efficient Algorithms for MPEG Video Compression", John Wiley & Sons Ltd, New York 2002.
- [Hor81] Horn B.K.P., Schunck B., "Determining Optical Flow", Artificial Intelligence, Vol. 17, pp. 185-203, 1981.
- [Ill97] Illgner K., Mueller F., "Spatially Scalable Video Compression Employing Resolution Pyramids", IEEE Journal on Selected Areas In Communications, Vol. 15, No. 9, pp. 1688-1703, December 1997.
- [Ish97] Ishwar P., Moulin P., "Switched control grid interpolation for motion compensated video coding", Proc. of International Conference on Image Processing ICIP97, Vol. 3, pp. 650-653, October 1997.
- [ISO06] ISO/IEC 14496-10, "Generic Coding of Audio-Visual Objects, Part 10: Advanced Video Coding", March 2006.
- [ISO06a] „H.264/AVC Software Coordination”, <http://iphome.hhi.de/suehring/tml/>, August 2006.
- [ISO06b] „SVC Reference Software”, <http://www.ient.rwth-aachen.de/cgi-bin/viewvcext/>, August 2006.
- [ISO93] ISO/IEC 11172-2, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1,5 Mbit/s, Part 2: Video" (MPEG-1), November 1993.
- [ISO94] ISO/IEC 13818-2/ITU-T Rec. H.262, "Generic Coding of Moving Pictures and Associated Audio, Part 2: Video" (MPEG-2), November 1994.
- [ISO98] ISO/IEC 14496-2, "Generic Coding of Audio-Visual Objects, Part 2: Visual" (MPEG-4 Visual), December 1998.
- [ITU03] ITU-R Rec. BT.500-11, "Methodology for the Subjective Assessment of the Quality of Television Pictures", 2003.
- [ITU05] ITU-T Rec. H.263, "Video Coding for Low Bit Rate Communication", August 2005.
- [ITU94] ITU-R Rec. BT.500-6, "Methodology for the Subjective Assessment of the Quality of Television Pictures", 1994.

- [Jai81] Jain J.R., Jain A.K., "Displacement Measurement and Its Application in Interframe Image Coding", IEEE Transactions on Communications, Vol. 29, No. 12, pp. 1799-1808, December 1981.
- [Ji04] Ji X., Xu J., Zhao D., Wu F., "Architectures of Incorporating MPEG-4 AVC into Three-dimensional Subband Video Coding", Picture Coding Symposium, PCS04, San Francisco, USA, 2004.
- [JVT06-02] "Joint Scalable Video Model JSVM-6", ITU-T VCEG SG16 Q.6, Doc. JVT-S202, Geneva, Switzerland, March-April, 2006.
- [Kar04] Karim H.A., Bister M., Siddiqi M.U., "Multiresolution Motion Estimation for Low-rate Video Frame Interpolation", EURASIP Journal on Applied Signal Processing, Vol. 2004:11, pp. 1708–1720, Hindawi Publishing Corporation, New York, 2004.
- [Kim99] Kim S.D., Ra J.B., "An Efficient Motion Vector Coding Scheme Based on Minimum Bitrate Prediction", IEEE Transactions on Image Processing, Vol. 8, Issue 8, pp. 1117 – 1120, August 1999.
- [Kog81] Koga T., Iinuma K., Hirano A., Iijima A., Ishiguro T, "Motion-Compensated Interframe Coding for Video Conferencing", Proc. of National Telecommunications Conference, vol. 4, pp. G5.3.1-G5.3.5, 1981.
- [Kri97a] Krishnamurthy R.: "Compactly-encoded Optical Flow Fields for Motion-compensated Video Compression and Processing", doctoral thesis, Rensselaer Polytechnic Institute Troy, NY, 1997.
- [Kri97b] Krishnamurthy R., Moulin P., Woods J.W., "Compactly-encoded Optical Flow Fields for Bidirectional Prediction and Video Coding", Proc. SPIE, Visual Communications and Image Processing '97, Vol. 3024, pp. 1164-1173, January 1997.
- [Lai02] Lainema J., Karczewicz M., "Skip Mode Motion Compensation", ITU-T VCEG SG16 Q.6, Doc. JVT-C027, Fairfax, May 2002.
- [Lam93] Lam W. M., Rebman A.M., Liu B., "Recovery of Lost or Erroneously Received Motion Vectors", Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing 1993, ICASSP-93, Vol. 5, pp. 417-420, April 1993.
- [Lee95a] Lee Y. Y., Woods J.W., "Motion Vector Quantization for Video Coding", IEEE Transactions on Image Processing, Vol. 4, Issue 3, pp.:378 – 382, March 1995.
- [Lee95b] Lee Y. Y., Woods J. W., "Macro Motion Vector Quantization," Proc. SPIE Visual Communications and Image Processing '95, Vol. 2501, pt.2, pp. 778-86, 1995.

- [Li94] Li R., Zeng B., Liou M.L., "A New Three Step Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits, Systems and Video Technology, Vol. 4, pp. 438-442, August 1994.
- [Lig89] Ligtenberg A., "No Bells", ISO/IEC JTC1/SC2/WG8, Doc. MPEG 89/183, October 1989.
- [Lim71] Limb J. O., Pease R. F. W., "A Simple Interframe Coder for Video Telephony", Bell System Technical Journal, Vol. 50, pp. 1877-1888, July-August 1971.
- [Lin97] Lin S., Shi Y.Q., Zhang Y.-Q., "An Optical Flow Based Motion Compensation Algorithm for Very Low Bit-rate Video Coding", IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-97., Vol. 4, pp.: 2869-2872, April 1997.
- [Mac02] Maćkowiak S. , "Scalable Coding of Digital Video", Doctoral Thesis, Poznań University of Technology, Poznań 2002.
- [Mac03] Maćkowiak S. , "Multi-loop Scalable MPEG-2 Video Coders", Lecture Notes in Computer Science 2756, (ed. by N. Petkov and M. Westenberg), pp. 262-269, Springer, Berlin / Heidelberg 2003.
- [Mar01] Marpe D., Blättermann G., Wiegand T., "Adaptive Codes for H.26L", ITU-T Video Coding Experts Group, Doc. VCEG-L13, Eibsee, January 2001.
- [Mar01a] Marpe D., Blättermann G., Heising G., Wiegand T., "Further Results for CABAC Entropy Coding Scheme", ITU-T Video Coding Experts Group, Doc. VCEG-M59, Austin, April, 2001.
- [Mar03] Marpe D., Schwarz H., Wiegand T., "Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, Issue: 7, pp.: 620- 636, July 2003.
- [Moo06] Moon Y.H., Ki D.W., Kim J.H., "A Hybrid Motion Vector Coding Scheme Based on an Estimation of the Locality for Motion Vector Difference", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, Issue 6, pp. 781-785, June 2006.
- [Mou69] Mounts F.W., "A Video Encoding System with Conditional Picture-Element Replenishment", Bell System Technical Journal paper, pp. 2545-2554, September 1969.
- [MP02-20] "ISO/IEC 14496-10 AVC, Text of Final Committee Draft of Joint Video Specification", ISO/IEC JTC1/SC29/WG11 Doc. N4920, July 2002.

- [MP02-35] “AHG on Scalable Video Coding”, ISO/IEC JTC1/SC29/WG11, Doc. N4935, July 2002.
- [MP03-25] "Requirements and Applications for Scalable Video Coding", ISO/IEC JTC1/SC29/WG11 Doc. N6025, October 2003.
- [MP03-93] "Call for Proposals on Scalable Video Coding Technology", ISO/IEC JTC1/SC29/WG11 Doc. N6193, December 2003.
- [MP04-20] “Scalable Video Model 2.0”, ISO/IEC JTC1/SC29/WG11, Doc. N6520, July 2004.
- [MP04-69] "Registered Responses to the Call for Proposals on Scalable Video Coding", ISO/IEC JTC1/SC29/WG11 Doc. M10569, Munich, March 2004.
- [MP04-72] “Scalable Video Model V 1.0”, ISO/IEC JTC1/SC29/WG11, Doc. N6372, March 2004.
- [Muk85] Mukawa, N., Kuroda, H., “Uncovered Background Prediction in Interframe Coding”, IEEE Transactions on Communications, Vol. 33, Issue 11, pp. 1227 – 1231, November 1985.
- [Nak91] Nakaya Y., Harashima H., "An Iterative Motion Estimation Method Using Triangular Patches for Motion Compensation," Proc. SPIE Visual Communications and Image Processing, Vol. 1605, pp. 546--557, November 1991
- [Nav94] Naveen T., Woods J.W., “Motion Compensated Multiresolution Transmission of High Definition Video”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 4, Issue 1, pp. 29 - 41, February 1994.
- [Nav95] Naveen T., Bosveld F., Woods J.W., Lagendijk R.L., “Rate Constrained Multiresolution Transmission of Video”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, Issue 3, pp. 193 - 206, June 1995.
- [Net79] Netravali A. N., Robbins J. D., “Motion Compensated Television Coding: Part-I”, Bell System Technical Journal, Vol. 58, pp. 631-670, March 1979.
- [Nin82] Ninomiya Y., Ohtsuka Y., “A Motion-Compensated Interframe Coding Scheme for Television Pictures”, IEEE Transactions on Communications, Vol. COM-30, No. 1, pp. 201-211, January 1982.
- [Nog92] Nogaki S., Ohta M., “An Overlapped Block Motion Compensation for High Quality Motion Picture Coding”, Proc. IEEE International Symposium on Circuits and Systems, ISCAS '92, Vol. 1, pp. 184-187, May 1992.

- [Oel04] Oelbaum T., Baroncini V., Tan T.K., Fenimore C., “Subjective Quality Assessment of The Emerging AVC/H.264 Video Coding Standard”, International Broadcasting Convention, IBC,Amsterdam 2004.
- [Ohm01] Ohm J.-R., van der Schaar M., “Scalable Video Coding”, Tutorial Material, SM.T3, IEEE International Conference on Image Processing, ICIP’2001, Thessaloniki, Greece, 2001.
- [Ohm02] Ohm J.-R., “Complexity and Delay Analysis of MCTF Interframe Wavelet Structures,” ISO/IEC JTC1/WG11 Doc. M8520, Klagenfurt, Austria, July 2002.
- [Ohm02a] Ohm J.-R., Ebrahimi T., “Report of Ad hoc Group on Exploration of Interframe Wavelet Technology in Video”, ISO/IEC JTC1/SC29/WG11 Doc. MPEG02/M8359, Fairfax, May 2002.
- [Ohm04] Ohm J.-R., “Multimedia Communication Technology”, Springer-Verlag, Berlin, 2004.
- [Ohm05] Ohm J.-R., “Advances in Scalable Video Coding”, Proc. of the IEEE, Vol. 93, Issue 1, pp. 42 – 56, January 2005.
- [Ohm92] Ohm J.-R., “Temporal Domain Sub-band Video Coding with Motion Compensation”, Proc. International Conference on Acoustic, Speech and Signal Processing, Vol. 3, pp. 229-232, San Francisco, USA, March 1992.
- [Ohm93] Ohm J.-R., “Advanced Packet-Video Coding Based on Layered VQ and SBC Techniques”, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 3, No. 3, pp. 208-221, June 1993.
- [Oja03] Ojansivu V., Silven O., Huotari R., "A Technique for Digital Video Quality Evaluation", Proc. IEEE International Conference on Image Processing ICIP'03, Vol. 3, pp. 181-184, Barcelona, Spain, 2003.
- [Ort98] Ortega A., Ramchandran K., “Rate-Distortion Methods for Image and Video Compression: An Overview”, IEEE Signal Processing Magazine, Vol. 15, Issue 6, pp. 23-50, November 1998.
- [Pas06] Pastrana-Vidal R., Gicquel J.-Ch., “Automatic Quality Assessment of Video Fluidity Impairments Using A No-Reference Metric”, Proc. of International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-06, Scottsdale 2006.
- [Per02] Pereira F., Ebrahimi T., „The MPEG-4 Book”, Prentice Hall, Upper Saddle River, 2002.
- [Pod95] Podilchuk C., Jayant N., Farvardin N., "Three Dimensional Subband Coding of Video," IEEE Trans. on Image Processing, Vol. 4, No. 2, pp. 125-139, February 1995.

- [Pur94] Puri A., Yan, L., Haskell B.G., "Temporal Resolution Scalable Video Coding", Proc. IEEE International Conference on Image Processing, ICIP'1994, Vol. 2, Pages: 947 - 951, November 1994.
- [Ram94] Ramchandran K., Ortega A., Vetterli M., "Bit Allocation for Dependent Quantization with Applications to Multiresolution and MPEG Video Coders," IEEE Transactions on Image Processing, Vol. 3, no. 5, pp. 533-545, September 1994.
- [Rea02] Reader C., „History of Video Compression”, ITU-T VCEG SG16 Q.6, Doc. JVT-D068, Klagenfurt, July 2002.
- [Reg97] Regunathan S.L., Rose K., "Motion Vector Quantization in a Rate-Distortion Framework", Proc. International Conference on Image Processing, ICIP'97, Vol. 2, pp.:21 – 24, October 1997.
- [Rib03] Ribas-Corbera J., "Windows Media 9 Series – A Platform to Deliver Compressed Audio and Video for Internet and Broadcast Applications", EBU Technical Review, January 2003.
- [Rib99] Ribas-Corbera J., "Core Experiment on Motion Compensated Prediction: Accuracy and Interpolation", ITU-T VCEG SG16 Q.15, Doc. Q15-I-59, Red Bank, October 1999.
- [Ric02] Richardson I.E.G., "Video Codec Design", John Wiley & Sons Ltd., Chichester 2002.
- [Ric02a] Richardson I.E.G., "H.264/MPEG-4 Part 10 White Paper, Variable Length Coding", www.vcodex.com, October 2002.
- [Ric03] Richardson I.E.G., "H.264 and MPEG-4 Video Compression, Video Coding for Next-generation Multimedia", John Wiley & Sons Ltd., Chichester 2003.
- [Roc69] Rocca F., "Television Bandwidth Compression Utilizing Frame-to-Frame Correlation and Movement Compensation", Symposium on Picture Bandwidth Compression, Massachusetts Institute of Technology, Eds. Huang T.S., Tretiak O.J., Gordon and Breach, 1972.
- [Ros01] Rose K., Begunathan S., "Toward Optimality In Scalable Predictive Coding", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, pp. 965-976, July 2001.
- [Sad02] Sadka A.H., "Compressed Video Communications", John Wiley & Sons Ltd., Chichester, 2002.
- [Sal98] Salomon D., „Data Compression”, Springer Verlag, New York, 1998.

- [Say00] Sayood K., „Introduction to Data Compression”, Second Edition, Morgan Kaufmann Publishers, San Fransisco, 2000.
- [Sch04] Schwarz H., Marpe D., Wiegand T., "Scalable Extension of H.264/AVC", ISO/IEC JTC1/SC29/WG11 MPEG 04 Doc. M10569, Munich, Germany, March 2004.
- [Sch95] Schafer R., Sikora T., "Digital Video Coding Standards and Their Role in Video Communications", Proc. of IEEE, Vol. 83, No. 6, pp. 907-924, June 1995.
- [Shi00] Shi Y.Q., Sun H., "Image and Video Compression for Multimedia Engineering", CRC Press, Boca Raton 2000.
- [Ska93] Skarbek W., "Metody Reprezentacji Obrazów Cyfrowych", Akademska Oficyna Wydawnicza PLJ, Warszawa 1993.
- [Ska98] Skarbek W. (ed), "Multimedia. Algorytmy i Standardy Kompresji", Akademska Oficyna Wydawnicza PLJ, Warszawa 1998.
- [Ska99] Skarbek W. (ed), "Multimedia. Sprzęt i Oprogramowanie", Akademska Oficyna Wydawnicza PLJ, Warszawa 1999.
- [Smo04] Smolić A., Vatis Y., Schwarz H., Wiegand T., "Improved H.264/AVC Coding Using Long-Term Global Motion Compensation", Proc. VCIP 2004, SPIE Visual Communications & Image Processing, San Jose, CA, USA, January 2004.
- [Smo99] Smolić A., Sikora T., Ohm J.-R., "Long-Term Global Motion Estimation and its Application for Sprite Coding, Content Description and Segmentation", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, No. 8, pp. 1227-1242, December 1999.
- [SMP05] SMPTE Technology Committee C24 on Video Compression Technology, "Proposed SMPTE Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process", SMPTE 421M, August 2005.
- [Son89] Sony, "Description of the Proposed Coding Algorithm", ISO/IEC JTC1/SC2/WG8, Doc. MPEG 89/194, October 1989.
- [Str96] Strachan D., "Video Compression", The Society of Motion Picture and Television Engineers Journal, Vol. 105, pp. 68-73, February 1996.
- [Sul03] Sullivan G., "Some Potential Enhancements of H.264/AVC", ITU-T VCEG SG16 Q.6, Doc. VCEG-T03, San Diego, September 2003.
- [Sul05] Sullivan G.J, Wiegand T., "Video compression – from concepts to the H.264/AVC standard", Proc. of the IEEE, Special Issue on Advances in Video Coding and Delivery, Vol. 93, No. 1, pp. 18-31, January 2005.

- [Sul91] Sullivan G.J., Baker R. L., "Motion Compensation for Video Compression Using Control Grid Interpolation", Proc. IEEE Conference on Acoustic, Speech, Signal Processing, ICASSP '91, pp. 2713-2716, Toronto, Canada, May 1991
- [Sul91a] Sullivan G.J., Baker R.L., "Rate-Distortion Optimized Motion Compensation for Video Compression Using Fixed or Variable Size Blocks", Global Telecommunications Conference, 1991, GLOBECOM '91, Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services, Vol. 1, pp. 85-90, December 1991.
- [Sul98] Sullivan G.J., Wiegand T., "Rate-Distortion Optimization for Video Compression", IEEE Signal Processing Magazine, Vol. 15, Issue 6, pp. 74-90, November 1998.
- [Sun01] Sun S., Lei S., "Global Motion Vector Coding (GMVC)", ITU-T VCEG SG16 Q.6, Doc. VCEG-O20, Pattaya, December 2001.
- [Suz02] Suzuki Y., "Adaptive Motion Vector Coding", ITU-T VCEG SG16 Q.6, Doc. JVT-B061, Geneva, January 2002.
- [Swe95] Sweldens W., "The Lifting Scheme: A New Philosophy In Biorthogonal Wavelet Constructions", Proc. of SPIE 2569, Wavelet Applications in Signal and Image Processing III, pp. 68-79, 1995.
- [Tau94] Taubman D., Zakhor A., "Multirate 3-D Subband Coding of Video", IEEE Trans. On Image Processing, Vol. 3, No. 5, pp. 572-588, September 1994.
- [Tek95] Tekalp A. M., "Digital Video Processing", Prentice Hall, Upper Saddle River, 1995.
- [Teu78] Teuhola J., "A Compression Method for Clustered Bit-Vectors", Information Processing Letters, Vol. 7, pp. 308-311, October 1978.
- [Tou01] Tourapis A. M., Xu J., Cheong H.Y., Liou M.L., Au O.C., "Temporal Interpolation of Video Sequences Using Zonal Based Algorithms" Proc. of the 2001 IEEE International Conference on Image Processing, ICIP'01, Vol. 3, pp. 895-898, Thessaloniki, Greece, October 2001.
- [Tou02] Tourapis A. M., Xu J., Wu F., Li S., "Motion Vector Prediction in Bidirectionally Predictive (B) frames with regards to Direct Mode", ITU-T VCEG SG16 Q.6, Doc. JVT-C127, Fairfax, May 2002.
- [Tou05] Tourapis A.M., Wu F., Li S., "Direct Mode Coding for Bipredictive Slices in the H.264 Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 1, pp. 119-126, January 2005.

- [Tou99] Tourapis A. M., Au O. C., Liou M. L., Shen G., "Status Report of Core Experiment on Fast Block-Matching Motion Estimation using Diamond Zonal Search with Embedded Radar", ISO/IEC JTC1/SC29/WG11 Doc.M4917, July 1999.
- [Wal87] Walker D. R., Rao K. R., "Motion-Compensated Coder", IEEE Trans. on COM , Vol. 35, no. 11, pp. 1171-1178, November 1987.
- [Wan02] Wang Y., Ostermann J., Zhang Y.-Q., "Video Processing and Communications", Prentice Hall, Upper Saddle River, 2002.
- [Wan98] Wang Y., Zhu Q. F., "Error Control and Concealment for Video Communication: A Review", Proc. of the IEEE, Vol. 86, No. 5, pp. 974-997, May 1998.
- [Web05] Webb R. C., "Tele-Visionaries: The People Behind the Invention of Television", Wiley-IEEE Press, Chichester, 2005.
- [Wei99] Wei-Cheng Gu; Lin, D.W., "Joint Rate-Distortion Coding of Multiple Videos", IEEE Transactions on Consumer Electronics, Vol. 45, Issue 1, pp. 159 - 164, February 1999.
- [Wen97] Wen G., Villasenor J. D., "A Class of Reversible Variable Length Codes for Robust Image and Video Coding," Proc. of the IEEE International Conference on Image Processing, Vol. 2, pp. 65-68, October 1997.
- [Wie03] Wiegand T., Sullivan G.J., Bjontegaard G., Luthura A., "Overview of the H264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, pp. 560-576, July 2003.
- [Wie04] Wien M., Rusert T., Hanke K., "RWTH Proposal for Scalable Video Coding Technology", ISO/IEC JTC1/SC29/WG11 MPEG04 Doc. M10569, Munich, March 2004.
- [Wie96] Wiegand T., Lightstone M., Mukherjee D., Campbell T. G., Mitra S. K., "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, Issue 2, pp. 182 - 190, April 1996.
- [Wie99a] Wiegand T., Xiaozheng Z., Girod B., "Long-term Memory Motion-Compensated Prediction", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 9, Issue 1, pp.:70-84, Feb. 1999.
- [Win05] Winkle S., "Digital Video Quality. Vision Models and Metrics.", John Wiley & Sons Ltd., Chichester, 2005.
- [Wit87] Witten I.H., Neal R.M., Cleary J.G., "Arithmetic Coding for Data Compression", Communications of the ACM, Vol.30, pp.520-540, 1987.

- [Won95] Wong C.K. , Au O.C., "Fast Motion Compensated Temporal Interpolation for Video", Proc. Of SPIE, Visual Communication and Image Processing, Vol. 2, pp.1108-1118, May 1995.
- [Woo02] Woods J. W., Chen P., "Improved MC-EZBC with Quarter-pixel Motion Vectors", ISO/IEC JTC1/SC29/WG11 Doc. MPEG02/M8366, Fairfax, VA, May 2002.
- [Xin99] Xin T., Heeger D., Van Den Branden Lambrecht C., Rogowitz Bernice E., Pappas Thrasylvoulos N., „Video quality evaluation using ST-CIELAB”, Proc. of SPIE Human Vision and Electronic Imaging, vol. 3644, pp. 185-196, San Jose 1999.
- [Xu02] Xu J., Xiong Z., Li S., Zhang Y.Q., “Memory Constrained 3D Wavelet Transform for Video Coding Without Boundary Effect”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, pp. 850-856, October, 2002.
- [Xu04] Xu J., Xiong R., Feng B., Sullivan G., Lee M.-C., Wu F., Li S., "3D Sub-band Video Coding using Barbell lifting", ISO/IEC JTC1/SC29/WG11 Doc. MPEG2004/M10569/S05, Munich, March 2004.
- [Yam89] Yamada Y., "Beach and Flower Garden Demonstration", ISO/IEC JTC1/SC2/WG8, Doc. MPEG 89/005, February 1989.
- [Zaf93] Zafar S., Zhang Y.-Q., Jabbari B., “Multiscale Video Representation Using Multiresolution Motion Compensation and Wavelet Decomposition”, IEEE Journal on Selected Areas in Communications, Vol. 11, Issue 1, pp.:24 – 35, January 1993.
- [Zhe03] Zheng J., Chau L.-P., "A motion vector recovery algorithm for digital video using Lagrange interpolation", IEEE Transactions on Broadcasting, Vol. 49, No. 4, pp. 383-389, December 2003.
- [Zhe05] Zheng J., "Efficient motion vector recovery algorithm for H.264 based on a polynomial model", IEEE Transactions on Multimedia, Vol. 7, No. 3, pp. 507-513, June 2005.
- [Zhe93] Zheng J., Valavaris K. P., Gaugh J. M., “Noise Removal from Color Images”, Journal of Intelligent Robotic Systems, Vol. 7, No. 3, pp. 257-285, June 1993.
- [Zil05] Ziliani F., Michelou J.-C., “Scalable Video Coding in Digital Video Security - White Paper”, VisioWave Sarl, Ecublens, Switzerland, 2005.

Annex – Test sequences

In this annex, the test sequences that were used in the experiments are presented. The short description of each video sequence is depicted together with the first video frame of the sequence. The sequences are presented in an alphabetical order.

Basket –

a fragment of basketball game. In the foreground there are basketball players, which are fighting for a ball. There is an audience in the background. The camera slowly moves with translational motion.



Bus –

the camera is panning towards bus, which is riding rapidly in the background. In the foreground there is a parking car and a fence.



Cheer –

dancing cheerleaders. Mostly slow, rotational motion in the sequence. Almost static background.



City –

a bird's eye view of the city, probably filmed from a helicopter. Camera is slowly stirring with translational motion.



Crew –

crew of space shuttle coming out of the building. Rapid illumination changes caused by cameras flashes. Camera is panning the walking crew.



Football –

a fragment of American football match. In the foreground, players are moving rapidly in all directions, in the background there is a complex texture: muddy grass. Extremely fast and complex motion with rapid camera panning.



Foreman –
speaking foreman with rich
facial expression. Head of
the person is moving
rapidly.



Fun –
slowly turning carousel.
Some parts of carousel
move upward and
downward alternately.
There is a riding bus present
in the background.



Ice –
a scene from ice rink.
People are skating around.
Motion is moderate,
however, locally fast.



Mobile –

a sequence with slow but very complex motion. A lot of different types of motion: rolling ball, moving toy and a calendar. Additionally, camera is slowly panning. Complex texture.



Stefan –

tennis player during his match. Fast and complex local motion (the player and the racquet), static background (audience).

