ITU-Telecommunications Standardization Sector Document Q15-A-18
STUDY GROUP 16 Filename: q15a45.ps
Video Coding Experts Group

Portland, 24-27 June 1997

Question: Q.15/16

Source: University of British Columbia
Department of Electrical Engineering

CANADA Guy Côté

Title: Computation constrained fast-search motion estimation algorithm for TMN-7

Efficient Motion Vector Estimation and Coding for H.263-Based Very Low Bit Rate Video Compression *

Guy Côté, Michael Gallant, and Faouzi Kossentini
Department of Electrical Engineering,
University of British Columbia,
Vancouver BC V6T 1Z4, Canada

Abstract

In this paper, we propose an efficient motion vector estimation and coding method for H.263-based low bit rate video compression. The method exploits structural constraints within the motion field. The same H.263 median predictor is used to localize motion estimation, which is directed to the best motion vectors by employing a computation-constrained layered search technique. Moreover, the localized motion field is encoded using semi-fixed-length codes. The resulting low bit rate video encoder yields essentially the same levels of rate-distortion performance and subjective quality achieved by Telenor's implementation of the ITU TMN5 model. However, our motion vector estimation and coding method provides for substantially higher encoding speed and channel error robustness.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Very low bit rate video compression techniques are becoming increasingly important due to present and forthcoming visual services, e.g. videophone over analog telephone lines or over wireless channels, and applications on the internet. This ever-expanding area has generated a lot of research [1, 2, 3, 4, 5] and increased standardization activities, especially those of the H.263 and MPEG-4 groups.

The H.263 video standard [6] was developed for videophone applications with a target bit rate of 64 kilobits per second (kbps) and below. It is based on techniques common to many current video coding standards. Motion-compensated prediction, usually based on simple block matching algorithms (BMAs), first removes temporal redundancies. DCT-based algorithms are then used for residual coding. However, the H.263 standard also allows for more advanced prediction techniques and more flexible macroblock coding modes than the earlier H.261 and MPEG standards. These new features address the fact that, in a H.263 framework, the number of bits representing the motion field is quite significant, as compared to the total number of bits required by the video coder [7].

H.263 encoders, e.g. Telenor's coder [8], which implement the ITU TMN5 model [9], employ a two-step motion vector estimation method. An integer-pel accuracy motion vector is first obtained by exhaustively searching a fixed window around the predicted motion vector and minimizing the Sum of Absolute Difference (SAD). Half-pel motion vector values are then determined by interpolating the previous reconstructed frame around the integer-pel motion vector. While this technique guarantees that the minimum SAD candidate is found within the given search constraints, the large number of required SAD computations makes the above approach very computationally intensive. This suggests speed improvements.

Another area of improvement is motion vector coding. As specified in the H.263 standard [6], motion vectors are encoded by applying variable length codes (VLCs), defined in the standard, to difference motion vectors, obtained by subtracting two-dimensional median predicted vectors from the actual motion vectors. The horizontal and vertical motion vector components are coded independently, using the same VLC table. In the presence of channel errors, such a variable-length coding technique can easily cause a loss of synchronization, a potentially serious problem in wireless communication applications.

We propose an efficient motion vector estimation and coding method that addresses the above problems. Motion estimation involves two-dimensional median prediction and computation-constrained layered searching, while motion vector coding involves employing semi-fixed-length codes to represent the prediction difference vectors. The explicit computation constraint placed on motion estimation will favor the candidate motion vectors closest in distance to the predicted motion vector. Such was taken into account while designing the semi-fixed-length codes. A H.263-based coder employing the proposed motion estimation and coding techniques is shown to achieve essentially the same level of performance compared to Telenor's implementation. This is expected since the motion vectors in a low-resolution video sequence used in a typical H.263 application are usually slowly varying and/or equal to *zero* vectors. The advantage of our method, however, is that the resulting coder is less computationally intensive than ITU TMN5 implementations and generates bit streams which are more robust to channel errors than those of H.263-compliant coders. In the next section, we discuss the proposed computation-constrained motion estimation technique. This is followed by a detailed description of our motion vector coding technique. Experimental results and conclusions are provided in the last two sections.

2 Computation-Constrained Motion Estimation

TMN5 specifies that motion estimation be performed via a full-search, which minimizes the SAD within the search area, yet is computationally intensive. To reduce the number of computations, many sub-optimal motion vector search algorithms have been developed [10]. Unfortunately, most of these algorithms fall prey to local minima, as the distortion function does not necessarily increase monotonically as we move away from the global minimum distortion candidate. More recently, several predictive motion estimation algorithms, such as the ones described in [7, 11], have emerged that achieve significantly better computation-performance tradeoffs. In [7, 11], previously coded motion vectors representing spatially or temporally neighboring macroblocks can be used to predict the current motion vectors. Localizing the search in this way reduces the possibility of getting trapped in local minima, because the predicted motion vector is usually closer than the vector $(0, 0)$ to the global minimum candidate.

Our proposed algorithm is based on the computation-constrained predictive motion estimation algorithm developed for fast MPEG-2 video encoding in [11]. The H.263 two-dimensional median predictor is used to localize the integer-pel search procedure, which consists of sequentially searching diamond-shaped layers, each representing a set of candidate motion vectors. The search can be stopped only after (1) all candidate motion vectors in the current layer have been considered and (2) the minimum SAD value at the current layer is larger than the one of the previous layer.

Unlike in [11], where all the layers have the same center, and hence contain an increasing number of motion vector candidates as we go away from the center, the proposed algorithm employs equal-size layers having different centers and containing at most[1] four untested candidate motion vectors. This is illustrated in Figure 1. The basic rule is that the layer $i+1$ is centered at the point of minimum SAD of layer i. In other words, successive layers have the same diamond-shaped pattern, always containing the four immediate neighbors of the current search center. The search stopping criterion is identical to the one used in [11]. That is, if the minimum SAD for the current layer is not lower than that of the previous layer the search is halted. In such a case, it is not expected that the potential reduction in the SAD value by searching an additional layer would be computationally justified. Finally, after obtaining the integer-pel motion vector, the best half-pel motion vector is obtained as in Telenor's implementation of TMN5.

3 Semi-fixed-Length Encoding of the Motion Vectors

As specified in the H.263 standard, Telenor's encoder employs the same VLC table to encode each of the two components of the median prediction difference vector. To improve channel error resilience and robustness, it provides the ability to insert a synchronization word in the bit stream after a certain number of Group of Blocks (GOBs), where the synchronization frequency is user-specified. Using the above approach, an error in a motion vector will propagate until the next synchronization word is detected in the bit stream. However, by using semi-fixed-length motion vector encoding techniques [12] and protecting the important part of the motion vector bits (the header represented by the tree in Figure 2), the probability of error propagation can be minimized, at the expense of only a small increase in total bit rate.

Experimental results indicate that a large percentage of the motion vectors selected using Telenor's encoder belong to either Layer 0 or 1 (see Figure 1). By minimizing the lengths of the corresponding codes, most of the coding gain achieved by H.263's VLC technique can still be obtained. In [12], a simple semi-fixed-length coding scheme was developed that improved channel error robustness substantially. However, such a technique works only for integer-pel accuracy motion vectors. Moreover, motions vectors which do not belong to the probable region are assigned codes representing the nearest candidates which belong to the probable region. Our proposed technique is designed for half-pel accuracy motion vectors, as they yield a significant performance improvement [13], especially at the higher bit rates (e.g., as much as 2 dB at 64 kbps). Furthermore, motion vectors belonging to the outer layers are also assigned codes, as

[1] One or two of the four candidate motion vectors could have been considered in the previous layer.

their exclusion can often result in a considerable performance degradation.

Based on the statistical behavior of the localized motion field, the search area centered at the predicted motion vector is divided into *five* regions. Figure 3 depicts the difference motion vector regions and Table 1 represents the associated codewords and their probabilities. A tree representation of the header part is shown in Figure 2. The popular *gray* encoding method was employed to generate the specific codes for the difference motion vectors. Region 0 corresponds to the $(0,0)$ difference motion vector and is represented by *one* bit, as the $(0,0)$ is almost always the most probable. Region 1 represents the half-pel vectors around the $(0,0)$ integer-pel difference vector. Region 2 represents the $(1,0)$, $(0,1)$, $(-1,0)$, $(0,-1)$ integer-pel difference motion vectors, and their corresponding half-pel vectors. The remaining difference vectors within the $(\pm 16.5, \pm 16.5)$ rectangular region are represented by regions 3 and 4. In regions 1, 2 and 3, *three* bits are necessary to represent the half-pel vectors, but only *two* bits are necessary in region 4.

In terms of encoding efficiency, the proposed technique is very close to that of H.263's VLC technique. Using a training set of 11 standard QCIF video sequences, our technique yields an average difference motion vector code length of 4.64, as compared to an average of 4.21 for the VLC technique. The advantage of our coding technique, however, is that we are now able to protect a small portion of the motion vector code (header shown in Figure 2), substantially increasing channel error robustness. Of course, this technique may impose some additional complexity and processing delay, since the motion vector bits are here treated as data blocks [2].

To illustrate the higher channel error robustness of the proposed technique, a rate compatible punctured convolutional (RCPC) protection code was used [14, 15] to protect the motion vector header. This type of convolutional code provides an efficient means of implementing variable protection rates using a single encoder/decoder pair. Variable rate protection can be achieved in the following way: we design the error protection system so that it offers two or more different overall bit rates. The system uses channel-state information to decide what rate should be used. The rate chosen can be specified in the bitstream syntax. The RCPC code used here is described in [15] and is based on a rate 1/4 mother convolutional code.

4 Experimental Results

In our simulations, the proposed motion vector estimation and coding techniques are evaluated within a low bit rate video coding framework employing Telenor's TMN5 coder *version 2.0*

[2] In H.263, the motion vector bits for a given macroblock are mixed with other bits into the same bitstream.

[8] with the advanced prediction mode enabled. The rate control implemented by Telenor in their encoder release 1.6 was used (as opposed to the rate control described in TMN5 [9]) with a frame rate of 10 fps. Frames 0 to 149 of the following 11 QCIF sequences were used: CARPHONE, MISS AMERICA, CONTAINER SHIP, COASTGUARD, CLAIRE, FOREMAN, GRANDMA, MOTHER-DAUGHTER, SALESMAN, SUZIE, and TREVOR.

In the first set of experiments, we evaluate the computation-performance tradeoffs of our motion vector estimation technique. Table 2 represents the improvement or degradation of our fast-search technique as compared to Telenor's full-search technique. As can be seen from the table, our technique yields, on the average, a 0.01 dB loss in PSNR performance. However, the full-search technique requires 900 SAD calculations per macroblock whereas ours requires approximately 12 SAD calculations (on average). Figure 4 shows the motion estimation time on a frame per frame basis for the sequence CARPHONE. The computation time for our technique can be made constant (if desired) by fixing the number of layers to be searched. For example, the search can be limited to only layers 0 and 1, reducing the number of SAD calculations to a fixed number of 5 per macroblock. This is shown in Table 3, where little performance degradation can be observed.

Motion estimation based on full-search BMA's accounts for a major part (normally 75%) of the video encoder's computational load. Thus, our fast-search technique can make an enormous difference in applications where processing speed is very limited. In fact, our technique improves overall encoding time by a factor of as much as 4 : 1, as compared to Telenor's encoder[3]. This is encouraging, in light of the fact that the resulting low bit rate video coder is still H.263-compliant.

In the second set of experiments, we first compared the semi-fixed-length encoding technique to H.263's VLC one in a noiseless channel environment (with no header protection), while employing full-search motion estimation in both cases. The results are shown in Table 4. The VLC technique is only 0.02 dB better that ours, on the average. Table 5 presents the results obtained when both the new motion estimation and semi-fixed-length coding techniques are combined. The performance of our resulting encoder is very close (within 0.05 dB on average) to that of Telenor's encoder. Notice that the resulting encoder even slightly outperforms Telenor's in PSNR when coding slow-motion video sequences. This is due to the fact that the proposed technique uses less bits for small difference motion vectors, making available more bits for quantization of the difference macroblocks.

[3] Please visit our FTP site *ftp://monet.ee.ubc.ca/pub/tmn* to download the corresponding H.263 encoder software.

A unique advantage of the new encoder is that we now have the flexibility to protect the motion vector header using a few additional bits, yielding substantially increased channel error robustness. Results obtained by protecting the motion vector header using the RCPC code are shown in Figure 5. As expected, when no protection is applied, both the VLC and the semi-fixed-length coding techniques achieve essentially the same level of PSNR performance. Protecting the header of the motion vectors in our case, however, substantially improves channel error robustness as illustrated in the figure. Not shown in the figure is the fact that loss of synchronization due to channel errors usually localizes the decoding errors, making the decoded video sequence unusable. This is mainly due to the loss of a whole sequence of macroblocks. Of course, the sequence length depends greatly on the frequency of synchronization words. Using the proposed technique, the quality degrades but the decoded video sequence is always usable, at least as long as the header is received without error at the decoder. Adding a few protection bits to the header can eliminate the potentially catastrophic behavior of Telenor's coder, where a large number of whole macroblocks in a P-frame can be lost, that is, until the next synchronization word is found.

To summarize, the new encoder employing both fast-search motion estimation and semi-fixed-length motion vector coding is very close (on average) in PSNR to Telenor's encoder, even better for some slow-motion video sequences. In terms of subjective quality of the decoded video sequences, no difference was ever detected. The two main advantages of our encoder are increased encoding speed (as much as 4 : 1) and increased channel error robustness.

5 Conclusions

We have presented an efficient motion estimation and coding method that reduces the computational load and increases channel error robustness, with very little loss in PSNR performance and no loss in subjective quality. The new motion estimation algorithm allows for a highly localized and efficient search by incorporating explicitly a computation constraint. This resulting software encoder, which is still compliant with respect to the H.263 standard, can be downloaded from our FTP site *ftp://monet.ee.ubc.ca/pub/tmn/*.

The main contribution of this paper is the motion vector semi-fixed-length coding technique which takes advantage of the new motion field structure. The proposed technique produces a new motion vector code structure that is similar in concept to the one proposed in [12] but has two important advantages. First, it is more accurate by allowing half-pel motion vectors. Second, it is more robust with respect to input video statistics by allowing a relatively large search-area. Unlike the technique of [12], which leads to a 0.2 - 0.5 dB drop in PSNR as compared to Telenor's implementation, the proposed technique yields essentially the same levels in PSNR and subjective quality, while still leading to substantially higher channel error robustness if a small part of the motion vector code is protected. Finally, while incorporating the proposed semi-fixed-length coding technique makes the resulting coder incompliant with H.263, it will be proposed for inclusion in H.263+.

6 Acknowledgments

The authors would like to thank Dr. Gary Sullivan for his helpful and constructive comments and suggestions regarding this manuscript.

References

[1] K.-H. Tzou, H. G. Musmann, and K. Aizawa, "Special Issue on Very Low Bit Rate Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, pp. 213–367, June 1994.

[2] W. Li, Y.-Q. Zhang, and M. L. Liou, "Special Issue on Advances in Image and Video Compression," *Proc. of the IEEE*, vol. 83, pp. 135–340, Feb. 1995.

[3] H. Li, A. Lundmark, and R. Forchheimer, "Image Sequence Coding at Very Low Bitrates: A Review," *IEEE Trans. on Image Processing*, vol. 3, pp. 568–609, September 1994.

[4] B. Girard, K. Younes, R. Bernstein, P. Eisert, N. Farber, F. Hartung, U. Horn, E. Steinbach, and T. W. K. Stuhl muller, "Recent advances in video compression," in *IEEE International Symposium on Circuits and Systems*, Feb. 1996.

[5] B. Girod, "Advances in digital image communication," in *Proceedings 2nd Erlangen Symposium*, (Erlangen, Germany), Apr. 1997.

[6] ITU Telecom. Standardization Sector of ITU, "Video Coding for Low Bitrate Communication," *Draft ITU-T Recommendation H.263*, May 1996.

[7] Y. Lee, F. Kossentini, M. Smith, and R. Ward, "Predictive RD-constrained motion estimation for very low bit rate video coding," *Special Issue of the IEEE Transactions on Selected Areas in Communications, in press.*

[8] Telenor Research, "TMN (H.263) encoder/decoder, version 2.0," *TMN (H.263) codec*, June 1996.

[9] ITU Telecom. Standardization Sector of ITU, "Video Codec Test Model Near-Term, Version 5 (TMN5)," *H.263 Ad Hoc Group*, Jan. 1996.

[10] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture*. Boston: Kluwer Academic Publishers, 1995.

[11] F. Kossentini and Y. Lee, "Computation-constrained fast MPEG-2 encoding," *Signal Processing Letters, in press.*

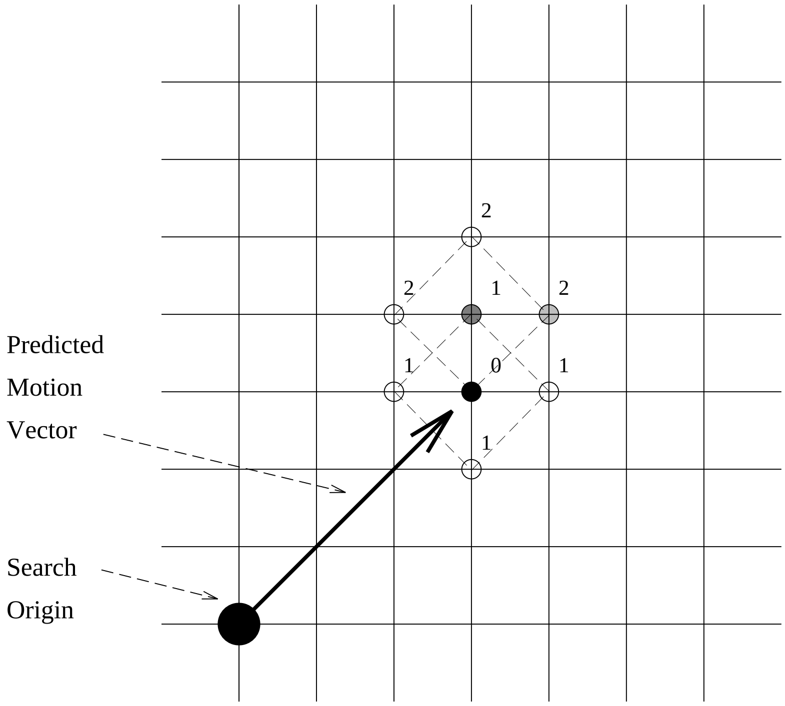
[12] Y. Lee, F. Kossentini, R. Ward, and M. Smith, "Towards MPEG4: An improved H.263-based video coder," *Signal Processing: Image Communication: Special Journal Issue on MPEG-4*, vol. 10, July 1997.

[13] B. Girod, E. Steinbach, and N. Faerber, "Comparison of the H.263 and H.261 video compression standards," in *Standards and Common Interfaces for Video Information Systems, K.R. Rao,editor, Critical reviews of optical science and technology*, vol. 60, (Philadelphia, Pennsylvania), pp. 233–251, Oct. 1995.

[14] S. Wicker, *Error Control Systems for Digital Communication and Storage.* Toronto: Prentice Hall Canada Inc., 1995.

[15] J. Hagenauer, "Rate-compatible punctured convolutional codes and their applications," *IEEE Trans. on Communications*, vol. 36, pp. 389–400, Apr. 1988.

List of Figures



0 - Layer 0 search location (predicted motion vector)

1 - Layer 1 search location

2 - Layer 2 search location

Note: A shaded point represents the minimum SAD of the respective layer.

Figure 1: Computation constrained layered search: an example of an integer-pel search extending two layers from the predicted motion vector.

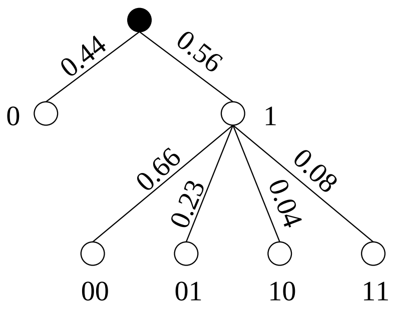


Figure 2: Tree representation of the headers of the semi-fixed-length codes (with their respective probabilities).

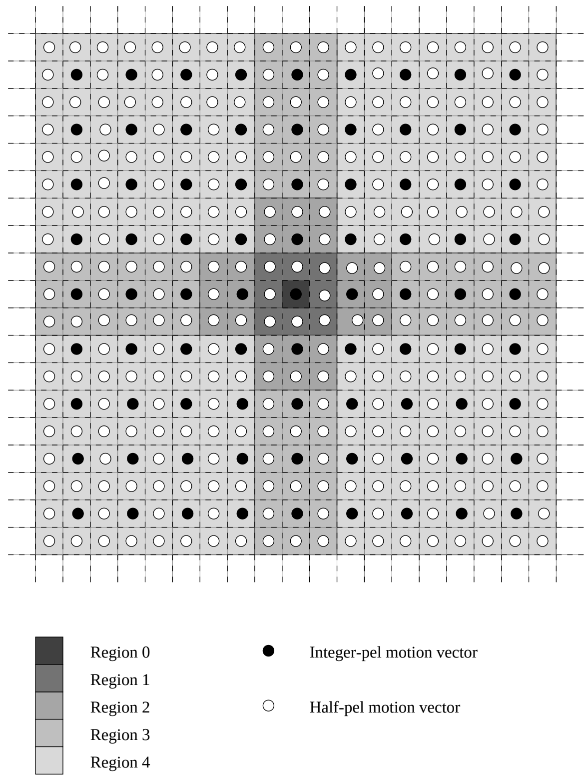


Figure 3: The semi-fixed-length coding technique: search area around the predicted motion vector.

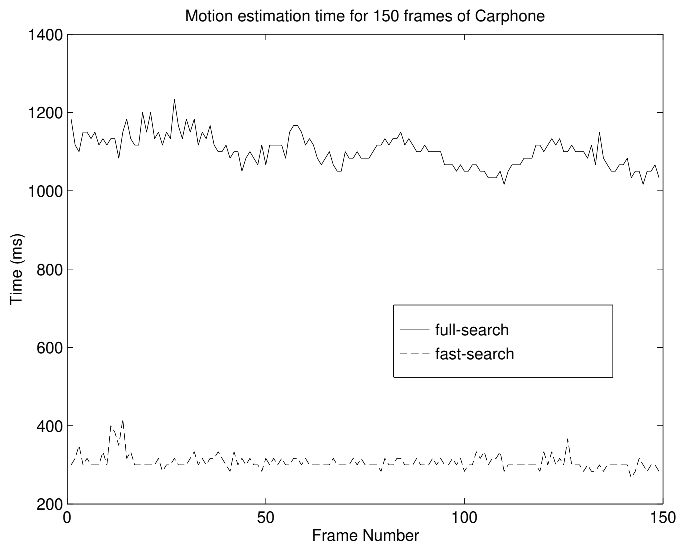


Figure 4: Motion estimation time for 150 frames of CARPHONE.

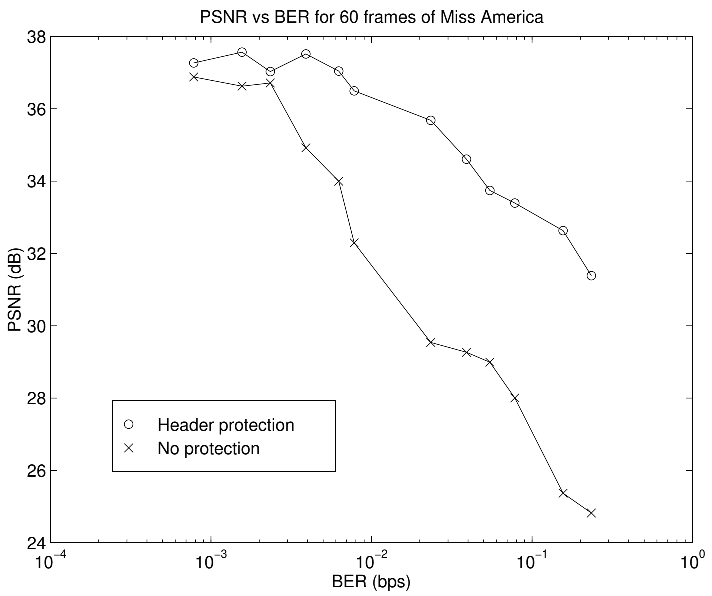


Figure 5: PSNR of our new motion vector coding technique over a binary symmetric noisy channel versus bit error rate (BER).

List of Tables

Region	Integer-pel code (Header)	Half-pel code	Probability
0	(0)	N/A	0.4387
1	(1 00)	XXX	0.3683
2	(1 01) XX	XXX	0.1295
3	(1 10) XX XXXX	XXX	0.0242
4	(1 11) XXXXX XXXXX	XX	0.0473

Table 1: Semi-fixed-length codes for the difference motion vectors

Sequence	8 kbps	10 kbps	12 kbps	14 kbps	16 kbps	24 kbps	Average
CARPHONE		+0.1649	+0.0618	+0.0019	-0.0339	-0.0465	+0.0296
MISS AMERICA	-0.0727	-0.0928	-0.0948	-0.1041	-0.0995	-0.0790	-0.0905
CONTAINER SHIP	+0.1124	+0.1108	+0.0498	-0.0029	-0.0587	-0.0346	+0.0295
COASTGUARD					+0.0603	+0.0163	+0.0383
CLAIRE	-0.1173	-0.0749	-0.1552	-0.0319	-0.0474	-0.0161	-0.0738
FOREMAN			-0.0143	+0.1136	-0.0087	-0.1349	-0.0110
GRANDMA	-0.0095	+0.0090	+0.1543	-0.0262	-0.0125	-0.0085	+0.0178
MOTHER-DAUGHTER	+0.0194	-0.0360	+0.0090	-0.0240	-0.0239	+0.0166	-0.0065
SALESMAN	+0.0400	+0.0076	-0.0022	-0.0163	-0.0400	-0.0061	-0.0028
SUZIE	+0.2451	+0.0251	+0.0122	-0.0688	-0.1033	-0.0840	+0.0044
TREVOR		-0.0211	+0.0015	-0.0687	-0.0401	-0.0380	-0.0333
Average	+0.0311	+0.0047	+0.0022	-0.0207	-0.0371	-0.0377	-0.0095

Table 2: Improvement (+) or degradation (-) in PSNR (dB) for the proposed fast-search technique over the full-search technique used by Telenor's encoder.

Sequence	8 kbps	10 kbps	12 kbps	14 kbps	16 kbps	24 kbps	Average
CARPHONE		-0.0075	-0.0926	-0.0926	-0.1074	-0.1010	-0.0802
MISS AMERICA	-0.0443	-0.0874	-0.1190	-0.1169	-0.0945	-0.0812	-0.0905
CONTAINER SHIP	+0.1125	+0.1148	+0.0487	-0.0045	-0.0577	-0.0346	+0.0299
COASTGUARD					-0.0730	+0.0114	-0.0308
CLAIRE	-0.1000	-0.0281	-0.1538	-0.0095	-0.0340	-0.0058	-0.0552
FOREMAN			-0.0625	+0.0312	-0.1153	-0.2081	-0.0887
GRANDMA	-0.0221	+0.0044	+0.1324	-0.0383	-0.0142	-0.0009	+0.0102
MOTHER-DAUGHTER	-0.0184	-0.0614	-0.0160	+0.0068	-0.0168	-0.0089	-0.0191
SALESMAN	-0.0106	-0.0187	-0.0221	-0.0191	-0.1411	-0.0932	-0.0508
SUZIE	+0.0592	-0.2429	-0.2103	-0.3479	-0.3433	-0.3181	-0.2339
TREVOR		+0.0204	-0.0054	-0.0446	-0.1129	-0.0636	-0.0412
Average	-0.0033	-0.0340	-0.0501	-0.0635	-0.1009	-0.0809	-0.0573

Table 3: Improvement (+) or degradation (-) in PSNR (dB) for the proposed fast-search technique (layers 0 and 1), over the full-search technique used by Telenor's encoder.

Sequence	8 kbps	10 kbps	12 kbps	14 kbps	16 kbps	24 kbps	Average
CARPHONE		-0.1359	-0.1344	-0.0590	-0.1000	-0.0349	-0.0928
MISS AMERICA	-0.0345	-0.0158	-0.0279	-0.0327	-0.0216	+0.0112	-0.0202
CONTAINER SHIP	+0.0717	+0.0892	+0.0659	+0.0195	+0.0493	+0.0164	+0.0502
COASTGUARD					-0.0330	-0.0420	-0.0375
CLAIRE	-0.0401	-0.0129	0.0084	+0.0703	-0.0867	+0.0261	-0.0058
FOREMAN			+0.0075	+0.0161	+0.0041	-0.0240	+0.0009
GRANDMA	-0.0079	+0.0460	+0.1218	+0.0118	-0.0157	+0.0377	+0.0323
MOTHER-DAUGHTER	-0.0165	-0.0726	-0.0137	-0.0250	-0.0309	-0.0066	-0.0275
SALESMAN	-0.0293	+0.0014	+0.0012	-0.0019	-0.0724	+0.0424	-0.0097
SUZIE	-0.0736	-0.1641	-0.0535	-0.1551	-0.1051	-0.0456	-0.0995
TREVOR		-0.0114	+0.0148	-0.0606	-0.0915	-0.0007	-0.0297
Average	-0.0186	-0.0306	-0.0009	-0.0216	-0.0457	-0.0018	-0.0210

Table 4: Improvement (+) or degradation (-) of PSNR in dB for the proposed semi-fixed-length coding over H.263 VLC.

Sequence	8 kbps	10 kbps	12 kbps	14 kbps	16 kbps	24 kbps	Average
CARPHONE		-0.0001	-0.1247	-0.1188	-0.0710	-0.0863	-0.0802
MISS AMERICA	-0.1168	-0.1063	-0.1030	-0.1290	-0.0892	-0.0411	-0.0976
CONTAINER SHIP	+0.1236	+0.1571	+0.0527	+0.0238	-0.0376	+0.0471	+0.0611
COASTGUARD					-0.0507	-0.0056	-0.0282
CLAIRE	-0.1134	-0.0873	-0.1227	-0.0358	-0.0235	+0.0122	-0.0618
FOREMAN			-0.1694	-0.0741	-0.1709	-0.2170	-0.1579
GRANDMA	-0.0274	+0.0745	+0.1382	+0.0116	-0.0156	+0.0210	+0.0338
MOTHER-DAUGHTER	-0.0476	-0.0426	-0.0237	-0.0028	+0.0272	+0.0179	-0.0119
SALESMAN	-0.0028	+0.0060	+0.0006	-0.0130	-0.0719	+0.0220	-0.0098
SUZIE	-0.1320	-0.1749	-0.1282	-0.2347	-0.1981	-0.1055	-0.1551
TREVOR		-0.1418	-0.1289	-0.1343	-0.1104	-0.0830	-0.1193
Average	-0.0452	-0.0350	-0.0609	-0.0665	-0.0787	-0.0378	-0.0570

Table 5: Improvement (+) or degradation (-) of PSNR in dB for proposed coder using both fast-search motion estimation and semi-fixed-length coding over Telenor's encoder.