FAST MOTION ESTIMATION DISCARDING LOW-IMPACT FRACTIONAL BLOCKS

Saverio G. Blasi, Ivan Zupancic and Ebroul Izquierdo

School of Electronic Engineering and Computer Science, Queen Mary University of London

ABSTRACT

Sub-pixel motion estimation is used in most modern video coding schemes to improve the outcomes of motion estimation. The reference frame is interpolated and motion vectors are refined with fractional components to reduce the prediction error. Due to the high complexity of these steps, sub-pixel motion estimation can be very demanding in terms of encoding time and resources. A method to reduce complexity of motion estimation schemes is proposed in this paper based on adaptive precision. A parameter is computed to geometrically characterise each block and select whether fractional refinements are likely to improve coding efficiency or not. The selection is based on an estimate of the actual impact of fractional refinements on the coding performance. The method was implemented within the H.264/AVC standard and is shown achieving considerable time savings with respect to conventional schemes, while ensuring that the performance losses are kept below acceptable limits.

Index Terms— Video coding, AVC, motion estimation

1. INTRODUCTION

Motion estimation (ME) is a fundamental component of the majority of modern video coding standards, from MPEG-2 [1] to H.264/AVC [2] (referred to as AVC in the rest of this paper) and its successor H.265/HEVC [3]. When using ME, the similarities among temporally adjacent frames are exploited to achieve compression. Rather than transmitting the actual frame samples, a prediction of such samples is extracted from neighbouring frames (referred to as reference frames). The displacement necessary to extract the prediction (referred to as motion vector, MV) is transmitted to the decoder along with the residual information between the prediction and original blocks. ME is commonly obtained using block-matching algorithms [4], where MVs are obtained separately for each block.

In order to improve the outcomes of ME, sub-pixel ME was introduced since the very first efforts in developing video coding technology. Sub-pixel ME is based on the idea that interpolating the reference frames and searching for MVs with fractional components can lead to higher compression efficiency. Sub-pixel ME is typically performed in a hierarchical manner. The best integer precision solution is used as basis for half-pixel precision ME; the best half-pixel precision solution is used for quarter-pixel ME, and so on. Samples at fractional locations need to be estimated from integer positions by means of interpolation, resulting in generally high computational complexity.

The impact of fractional ME in terms of coding efficiency was studied already with the introduction of the very first video coding models [5]. It was demonstrated that fractional refinements can improve compression efficiency only up to a certain limit, and that such limit is highly dependent on the actual content being considered. For sequences with high motion activity, quarter-pixel accuracy seems to be sufficient, while for sequences containing lower motion activity, half-pixel solutions are typically sufficient.

Due to the high computational complexity of sub-pixel ME [6], many fast algorithms were proposed to find the fractional MV refinements. Lee et al. [7] propose a fast two-step method for the half-pixel accuracy ME. When using such method, the number of computations is reduced limiting the amount of candidate blocks, by predicting the direction of minimum error and only searching for a solution towards such direction. Another method for fast fractional accuracy ME was also introduced [8]. In this approach, sub-pixel interpolation and subsequent secondary search after the integer precision ME are unified in a single step based on the motion-compensated prediction errors of the neighbouring pixels around the integer MV. Also, another method [9] based on mathematical models to avoid interpolation was proposed for quarter-pixel ME.

Another class of algorithms was introduced to decrease the computational complexity of sub-pixel ME, referred to as adaptive precision ME. When using such algorithms, the optimal precision level is selected for each block and MVs are computed only up to such level. Block features and statistics are mainly used to characterise the blocks and avoid fractional accuracy ME when possible, or to reduce the number of considered fractional resolutions. Some methods were proposed [10] [11], which make use of texture-based estimation schemes to determine the optimal MV resolution for different blocks. These methods only consider the input block characteristics, while not taking into account the results of integer-precision ME. Zhang et al. [12] proposed a method in which the prediction residuals from a 3×3 region around the minimum integer solution are used for computing parameters. Based on one of the parameters, referred to as deviation from flatness, and on a set of predefined thresholds, the appropriate MV resolution can be selected. However, within this method a fixed predefined threshold is used, and thus specific features of each particular sequence are not exploited. Finally a method was introduced recently [13], also based on a geometrical characterisation of the residual error surface. Such method takes into account the frequency of selection of fractional refinements to estimate the thresholds to select the precision for each block.

The method introduced in this paper is based on a similar idea as this previous approach [13], but it makes use of different techniques to measure the impact of fractional components during the encoding. In particular, instead of focusing on the frequency of selection of fractional solutions, the method takes into account the actual effects of sub-pixel refinements on the prediction error, referred to as fractional impacts. An estimate of the losses in compression efficiency as an effect of discarding fractional components is computed for each block. This is then used to classify whether to perform sub-pixel ME or not. The method is shown considerably reducing complexity while at the same time limiting the impact on compression losses regardless of the sequence being coded.

Email: {s.blasi, i.zupancic, e.izquierdo}@qmul.ac.uk.
This research was supported by the European Commission under contract FP7-285024 Advise.

2. THE IMPACT OF SUB-PIXEL ME

Even when using fast algorithms, sub-pixel ME still comes at the cost of higher computational complexity, and also requires a larger number of bits to transmit the fractional MV components. While its effects are in general beneficial to the coding efficiency, in some cases the outcomes of integer-precision ME are already sufficiently accurate. In these cases the fractional refinements do not impact the ME enough to compensate for the additional computational complexity.

In order to evaluate the effects of sub-pixel ME, some tests were performed on typical test sequences at different resolutions using the AVC standard. In particular, refer to the blocks that result in a fractional MV as fractional blocks. A measure of the impact of sub-pixel ME can then be obtained by computing the percentage of fractional blocks over the total number of inter-predicted blocks. Such percentage was computed for each test sequence and is presented in Table 1. Clearly, fractional refinements are selected in a high percentage of cases in the majority of the test sequences.

Although it can give a measure of how often sub-pixel ME is used, counting the number of fractional blocks might not be sufficient to evaluate the actual impact of fractional solutions on the coding efficiency. Conventional encoders select the optimal solution based on the computation of a rate-distortion (RD) cost. Fractional blocks are characterised by the fact that performing sub-pixel ME returns a RD cost that is lower than that returned by integer-precision ME, regardless of how large such gain is. On the other hand, for the purpose of reducing complexity, it makes sense to evaluate the magnitude of such gain and discard sub-pixel refinements on blocks where the gain is relatively small.

At this purpose, consider the difference between the optimal cost returned by sub-pixel ME, and the optimal cost previously obtained at integer-precision, and refer to such difference as the fractional difference (FD). The FD can be further normalised by the number of pixels in each block to remove any dependency on the block size. Note that the larger is such FD, the higher is the impact of the sub-pixel solution on the block, and consequently the higher is the expected coding gain as a consequence of using such solution instead of the integer-precision solution.

The average value of the fractional difference was computed for all fractional blocks in all sequences, referred to as $\overline{FD}$ in this paper. When a fractional block returns a FD higher than $\overline{FD}$, the encoding is likely to benefit greatly from using sub-pixel ME (as an effect of considerably reducing the prediction error). Such blocks are referred to as FD blocks. Conversely, blocks whose FD is smaller than $\overline{FD}$ are assumed to provide a smaller contribution to the coding efficiency as an effect of using sub-pixel ME; these blocks are referred to as low-impact (LI) fractional blocks. Following from such definitions, a more meaningful measure of the impact of sub-pixel ME on a certain sequence can be obtained by determining the amount of FD and LI blocks. This was computed for each test sequence, and is shown in Table 1 in terms of percentage of FD blocks over the total number of inter-predicted blocks. The percentage of LI blocks can be obtained subtracting this value from the percentage of fractional blocks.

The results in the table highlight the fact that sub-pixel solutions are chosen in many cases even if the actual impact of using fractional refinements might be relatively low. For instance, while 33.2% of the blocks are coded with fractional MV in the case of the Bowing sequence, in fact only 4.2% of the blocks are classified as FD, while the remaining 29% blocks can be classified as LI. Conversely, around a third of all fractional blocks was classified as FD in the case of the Crew sequence, highlighting the fact that sub-pixel refinements have a high impact when coding such sequence.

Finally, all test sequences where encoded by completely disabling sub-pixel ME, to measure its impact in terms of computational time. Results are also reported in Table 1 in terms of the percentage of time saved when disabling fractional ME with respect to conventional AVC.

Some conclusions can be derived from this analysis. The number of blocks in which fractional refinements are actually effective varies considerably depending on the coding conditions and the currently encoded sequence, and still in general the impact of using sub-pixel ME is too high to disable it completely. On the other hand, sub-pixel ME is considerably expensive in terms of computational time and complexity of the encoding. Clearly the complexity of the encoder could be consistently reduced by classifying the blocks among FD and LI before actually performing sub-pixel ME. A method to perform such classification is proposed here as described in the rest of this paper.

3. FRACTIONAL IMPACT AND GEOMETRICAL CHARACTERISATION

Consider that integer-precision ME is performed on a block by means of a certain error metric. In conventional coding schemes such metric is defined in terms of a distortion measure (for instance the sum of absolute differences, SAD) plus an estimated bitrate cost for coding the motion information, to obtain the total RD cost. The encoder tests a number of prediction candidates in the reference frame, each at a certain displacement (m, n). The RD cost $C_{m,n}$ is computed for each candidate until the solution at minimum error (m_0, n_0) is selected as the optimal integer-precision MV.

Conventional schemes would then consider an interpolation of the reference frame, and search for a refinement at fractional precision in the surrounding of (m_0, n_0). Consider now the RD cost values in the integer-precision locations close to (m_0, n_0). If such values are not are not very different from the integer-precision minimum C_{m_0,n_0}, it is reasonable to assume that RD cost values at frac-

Table 1. Percentage of fractional and FD blocks over the total number of inter-predicted blocks

Resolution	Sequence	Fractional Blocks	FD Blocks	Time for sub-pixel ME
416×240	Racehorses	89.6%	41.3%	25.73%
	Keiba	87.9%	43.2%	25.74%
	Mobisode2	44.8%	4.4%	26.45%
352×288	Crew	88.8%	28.9%	26.92%
	Foreman	83.8%	22.1%	26.88%
	Soccer	81.9%	36.6%	25.87%
	Bowing	33.2%	4.2%	26.55%
	Silent	29.7%	7.07%	26.29%

Table 2. Average SC depending on optimal displacement.

Resolution	Sequence	Average SC of FD blocks	Average SC of all other blocks
416×240	Racehorses	3.42	1.73
	Keiba	4.72	2.10
	Mobisode2	2.92	0.70
352×288	Crew	2.80	1.33
	Foreman	3.45	1.99
	Soccer	3.35	2.44
	Bowing	2.54	2.53
	Silent	2.91	3.25

tional locations would also be not very different from C_{m_0,n_0}. Consequently, even if a solution exists that decreases such cost, this is likely to provide relatively small gains, namely such block is probably a LI block.

Conversely, in case the RD cost values around (m_0, n_0) are very different from the current minimum cost, namely if the residual error surface has a very high curvature, then these variations of the surface may lead to a considerably smaller cost at a fractional location close to (m_0, n_0), which means the current block may be an FD block.

In order to quantify the amount of curvature of the residual error surface, a measure was already introduced [13], referred to as sampled curvedness (SC), which takes into account a window of 5×5 SAD values computed at locations $(m_0 + \delta_m, n_0 + \delta_n)$, where $\delta_m, \delta_n = -2, -1, ..., +2$. A characterisation of the curvature can be easily obtained analysing the 8 trajectories which include the center (m_0, n_0) and intercept 3 or 5 SAD values in the window. By computing the polynomial interpolation of such values and then studying the curvature of these interpolations, the curvature of the entire surface can be characterised.

The SC can be taken as a measure of the curvature of the residual error surface in the surrounding of the integer solution. At the beginning of this section, such curvature was assumed to be correlated with the probability that a certain block is an FD or a LI block. In order to validate such assumption, some tests were performed where the SC was computed for all inter-predicted blocks in each sequence. Each inter-predicted block was classified as FD depending on its fractional difference as illustrated in the previous section (using the same value of $\overline{FD}$).

The results of such analysis are shown in Table 2. The average SC found for all FD blocks is presented in the table for each test sequence, along with the average SC found for all other inter-predicted blocks (this includes LI blocks plus blocks predicted using integer-precision MVs). Clearly, FD blocks, namely blocks where fractional displacements produce considerable gains in terms of RD cost, tend to have higher values of the SC than other blocks. The average SC of FD blocks is higher than that of all other blocks in almost all test sequences. In some cases the gap between the two average values is very high. For instance, average SC in FD blocks is more than four times that of all other blocks in the Mobisode2 sequence. As an example, the values of the SC found for the first 500 blocks of the Crew sequence are shown in Figure 1.

These results can be used to select the precision of sub-pixel ME on a per-block basis. Any time a block is inter-predicted, an integer-precision MV is first derived, and the SC is computed. A certain threshold T is considered and:

1. If $SC < T$ no further action is required.
2. Else, sub-pixel ME is performed.

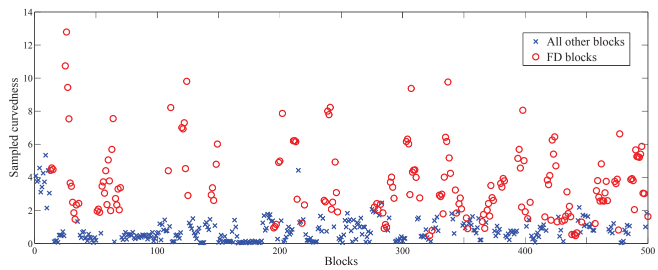

Fig. 1. SC in first 500 blocks of the Crew sequence.

Next block is then considered and encoding continues as conventionally.

4. OPTIMAL THRESHOLD BASED ON FRACTIONAL IMPACTS

Finding a suitable value for the threshold T is a crucial aspect of this method. Using a large value of T means that sub-pixel ME is skipped in too many blocks, consequently decreasing the compression efficiency of the encoder. Conversely, using a small value of T means that fractional refinements are computed in the majority of the blocks decreasing the algorithm performance. The optimal value of the threshold T is such that the number of skipped blocks is maximised while keeping the losses in compression efficiency below an acceptable limit. At this purpose, the algorithm was tested on a number of test sequences using a total of 10 possible values of the threshold spanning from 0 (i.e. when no blocks are skipped) to 4 to evaluate the performance in terms of time savings and compression losses.

Compression losses were measured by means of BD-rate [14], a performance metric which computes the bitrate difference obtained by two codecs in percentage. The method was compared with conventional AVC at four values of the QP (22, 27, 32 and 37, where QP is incremented by 1 in P frames). For simplicity, a low-delay configuration consisting of an I frame followed by a fixed number of P frames was used, but the approach can be easily extended to other configurations.

The approach works by disabling sub-pixel ME in a certain number of blocks, therefore the theoretical maximum time saving produced by the algorithm is given by the saving obtained completely disabling sub-pixel ME. For this reason the time saving gains of the approach are measured here with respect to such theoretical limit. Denote as L_{subpel} the total time (for all tested QPs) saved by disabling sub-pixel ME with respect to conventional AVC. Similarly denote as L_T the time saved by using the algorithm with a certain threshold T. Then the time saving of the algorithm (in percentage) is defined as $100 \times (L_T / L_{subpel})$.

A graphical representation of such tests can be found in the plot in Figure 2. Results are shown for all values of the threshold, on four sequences at CIF resolution. The vertical axis reports the BD-rate losses, whereas the horizontal axis reports the time savings. Obviously, the performances are greatly influenced by the value of the threshold T. Values smaller than 2 (distributed towards the origin of the graph) resulted in BD-rate losses no larger than 7% at the advantage of at least 30% time savings. Larger values of the threshold could be acceptable for some sequences (less than 8% BD-rate losses are reported in the case of the Silent sequence even when using the largest threshold of 4) but produce higher losses in other sequences.

Following from these observations, it is clear that using a fixed value of the threshold is not optimal. A different approach is proposed in this paper where the threshold is initially estimated based on the current sequence being coded, and it is successively updated to adapt to local changes of the content.

Figure 2 shows that the relationship between a certain threshold T and the corresponding performance losses is highly dependent on the content being encoded. While the actual effects of using such threshold in terms of bitrate losses are difficult to compute, an estimate of such losses can be obtained by computing the sum of all FDs (namely the difference between distortion obtained using and not using sub-pixel ME respectively) of the blocks whose SC is below the threshold. Sub-pixel ME would be skipped on such blocks and hence the corresponding gain as an effect of reducing the prediction error by FD would be lost. The sum of all these FDs is referred to in this paper as the fractional impact of using the threshold T.

To make sure that the algorithm achieves roughly the same performances on all sequences (obtaining the same losses in compression efficiency), the value of the threshold should be selected depending on the current content, with the goal of keeping the fractional impact constant. By selecting a reasonable target value of this fractional impact, the encoder would keep the performance losses below an acceptable limit while saving as much time as possible.

In order to achieve this goal, the relationship between T and fractional impact is initially estimated by coding a group of blocks with full sub-pixel ME and analysing the performances of different threshold values on such blocks. At this purpose, define an integer parameter K and consider a group of K blocks coded by an AVC encoder. Blocks are included in the group in the same order in which they are considered by the encoder. Notice that in case the encoder is testing the same portion of a frame several times (for instance when considering different inter-prediction modes), these are considered as individual blocks and separately considered in the group. In case an I frame is encountered which does not require a decoder refresh, the blocks within such frame are not considered while filling the group; this is completed with blocks extracted from the next P or B frame. If a decoder refresh is required, the encoding starts over as if at the beginning of a new sequence.

Consider now three values of the threshold T_i, where $i = 0, 1, 2$. The fractional impacts obtained as a result of using such thresholds while coding the blocks in the group can be easily obtained by considering corresponding three parameters E_i, for $i = 0, 1, 2$, which are initially set to zero. These parameters are updated after coding each block as:

1. If $SC < T_0$ and $FD > 0$ (namely if block would be misclassified using T_0), E_0 is incremented by FD.
2. If $SC < T_1$ and $FD > 0$, E_1 is incremented by FD.
3. If $SC < T_2$ and $FD > 0$, E_2 is incremented by FD.

The encoder considers all K blocks in the group to obtain the final fractional impacts E_i corresponding to using threshold T_i. Clearly from the above expressions $E_0 \leq E_1 \leq E_2$.

The values of E_i obtained for these three threshold values can be used to estimate a general relationship between T and the fractional impact. For simplicity a piecewise linear function is assumed here, as illustrated in Figure 3. Assume now that a certain target fractional impact is considered, referred to as E_{target}. Following from the relationship shown in the figure, a corresponding value of the threshold T_{target} can be obtained as:

1. If $E_1 < E_{target}$, $T_{target} = (T_1 - T_0)\frac{E_{target}-E_0}{E_1-E_0}$.
2. Otherwise $T_{target} = (T_2 - T_1)\frac{E_{target}-E_1}{E_2-E_1}$.

This is the value of the threshold which would have approximately produced a fractional impact equal to E_{target} while coding the group of K blocks. Under the assumption that the following blocks are correlated with this first group of blocks, setting the threshold to $T = T_{target}$ should ensure that roughly the same fractional impact is obtained during the encoding.

The encoder considers these next blocks using such value of the threshold to apply the algorithm presented in the previous subsection. The SC is computed for all blocks. If this is larger than T, the block is coded as in conventional schemes, otherwise if the SC is below T sub-pixel ME is skipped.

Using the proposed method is not optimal in case the K blocks used to estimate the threshold are highly different than the remaining blocks in the sequence. Clearly, the characteristics of a sequence can change considerably with time, which means the fractional impact produced as an effect of a threshold T can get very different than the target value E_{target}. In order to avoid high losses in compression efficiency, the threshold needs to be adapted to local characteristics of the coded sequence. A simple method to perform such adaptation was used in this paper. An index is incremented any time an inter-predicted block is coded, to keep track of the number of blocks input to the algorithm using a certain value of the threshold T. After a predefined number of blocks is input to the algorithm, namely the index becomes equal to a fixed value, the threshold is reset to zero. A new group of K blocks is coded in which the fractional impacts E_i are computed again, Finally, a new value of the threshold is estimated as illustrated in this section based on these fractional impacts.

The proposed approach is capable of obtaining considerable time savings while at the same time limiting the performance losses, as illustrated in the next section.

5. RESULTS

The approach was implemented using the AVC JM reference software v. 18.5 [15]. The method was tested on the same sequences as in Tables 1 and 2 (resolution of these sequences is reported in these tables). Again, four values of the QP were used (22, 27, 32 and 37, where QP is incremented by 1 in P frames). Results are presented in terms of time savings and BD-rate losses with respect to conventional AVC, where these are measured as defined in Section 4. A comparison with results obtained using the previously introduced method based on frequency of selection of fractional components [13], referred to as FS in the rest of this paper, is presented.

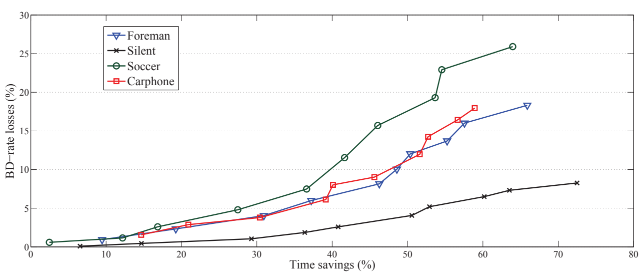

Fig. 2. BD-rate versus time savings.

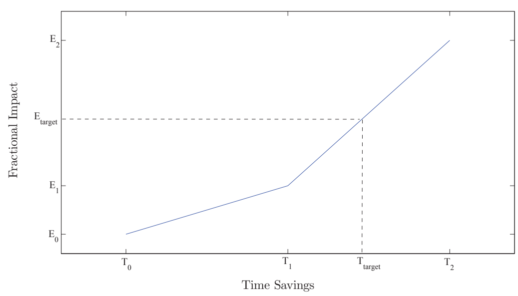

Fig. 3. Selection of the target threshold based on fractional impacts.

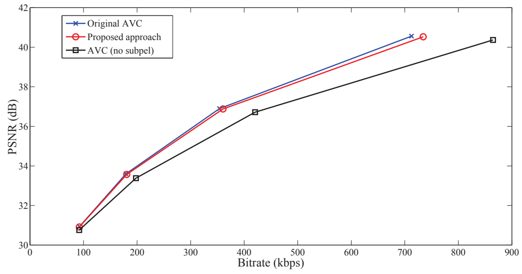

Fig. 4. Bitrate vs PSNR curve obtained for the Crew sequence with the proposed approach compared with original AVC and AVC without sub-pixel ME.

Full results are shown in Table 3. As illustrated in the table, the proposed approach is able to maintain an almost constant compression efficiency throughout all tested sequences. While other methods which do not take into account the impact of fractional refinements incur in either too high losses in terms of BD-rates or smaller time savings, acceptable losses are obtained using this method in all sequences, maximising the time savings in case many blocks can be coded using only integer-precision ME, or minimising the losses in case of high impact of fractional refinements.

BD-rate losses between 2.5 and 3.8 were obtained in all sequences, with time savings as high as 47.9% in the case of the Mobisode2 sequence, and no smaller than 10% in all tests. Sequences which present high fractional impact (such as Crew or Soccer) resulted in considerably smaller BD-rate losses using the approach proposed in this paper than the previously introduced FS approach. Conversely, sequences where the impact of sub-pixel ME is less predominant (such as Silent, Racehorses or Bowing) resulted in slightly larger BD-rate losses than the FS approach , but at the advantage of generally higher time savings.

As an example, the RD curve showing PSNR versus bitrate obtained for the four considered values of the QP when coding the Crew sequence is shown in Figure 4 for the proposed approach, compared with original AVC, and AVC without sub-pixel ME. RD performance of the proposed approach is very similar to conventional AVC with very small losses in compression efficiency as opposite to completely disabling sub-pixel ME, while still providing around 16% time savings as reported in Table 3.

6. CONCLUSIONS

A novel method for fast ME in the context of the AVC standard is presented in this paper, based on adaptive selection of the optimal fractional precision of the MVs. The method is based on a geometrical characterisation of the residual error surface as output from integer-precision ME. By estimating the actual impact of fractional refinements on each block, the encoder can decide whether to perform sub-pixel ME or not, while still ensuring that acceptable losses are obtained in terms of compression efficiency.

The approach is shown obtaining consistent time savings with respect to conventional AVC, while at the same time limiting the losses in compression efficiency. Higher time savings can be obtained with respect to previously introduced methods when low impact of fractional refinements is estimated on particular sequences, whereas acceptable performance losses are obtained in sequences where high impact of sub-pixel ME is predicted. The method is also capable of adapting to changes among frames in a sequence. More complex methods for updating the thresholds can be investigated to improve the approach.

Table 3. Results.

Sequence	**Proposed method**		**FS**	
	BD-rate	Time savings (%)	BD-rate	Time savings (%)
Racehorses	3.31	14.6	1.77	9.5
Keiba	3.03	10.3	1.55	8.6
Mobisode2	3.03	47.9	3.73	28.7
Crew	3.35	16.1	5.02	24.2
Foreman	3.78	22.3	3.82	14.4
Soccer	3.02	20.2	4.05	25.9
Bowing	2.57	43.1	1.08	29.9
Silent	3.40	28.7	1.19	19.9

REFERENCES

[1] ISO/IEC, "ISO/IEC 13818 - generic coding of moving pictures and associated audio information," Tech. Rep., ISO/IEC, 1995.

[2] ITU-T, "ITU-T recommendation H.264, advanced video coding for generic audiovisual services," Tech. Rep., ITU-T, 2003.

[3] G.J. Sullivan, J. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74–90, 1998.

[5] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604 –612, 1993.

[6] J.W. Suh and J. Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 3, pp. 968–973, Aug 2004.

[7] K.H. Lee, J.H. Choi, B.K. Lee, and D.G. Kim, "Fast two-step half-pixel accuracy motion vector prediction," *Electronics Letters*, vol. 36, no. 7, pp. 625–627, Mar 2000.

[8] J.W. Suh and J. Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 3, pp. 968–973, Aug 2004.

[9] J.W. Suh, J. Cho, and J. Jeong, "Model-based quarter-pixel motion estimation with low computational complexity," *Electronics Letters*, vol. 45, no. 12, pp. 618–620, June 2009.

[10] J. Ribas-Corbera and D.L. Neuhoff, "Optimizing motion-vector accuracy in block-based video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 4, pp. 497–511, Apr 2001.

[11] J. Ribas-Corbera and D.L. Neuhoff, "On the optimal motion vector accuracy for block-based motion-compensated video coders," in *Proc. SPIE Dig. Video Compr*, 1996, pp. 302–314.

[12] Q. Zhang, Y. Dai, and C.C.J. Kuo, "Direct techniques for optimal sub-pixel motion accuracy estimation and position prediction," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 12, pp. 1735–1744, Dec 2010.

[13] S.G. Blasi and E. Izquierdo, "Residual error curvature estimation and adaptive classification for selective sub-pel precision motion estimation," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 1193–1196.

[14] G. Bjontegaard, "Improvements of the BD-PSNR model," ITU-T SG16/Q6, 35th VCEG Meeting, Doc.VCEG-AI11, 2008.

[15] ITU, "JM Reference Software version 18.5 (online)," http://http://iphome.hhi.de/suehring/tml/.