Video Coding with Motion Estimation at the Decoder

Claudia Tonoli, Pierangelo Migliorati, and Riccardo Leonardi

1 Introduction

Compression efficiency is a fundamental requirement in all video coding systems. It becomes even more important in the case of small wireless devices because they are often subject to tighter power and bandwith constraint. In predictive video coding schemes, the key feature to achieve compression efficiency is motion estimation. The basic idea of this approach is to exploit the temporal redundancy across frames, estimated using the motion information. Usually motion estimation is performed at the encoder side, and then the motion field is transmitted to the decoder, together with the compressed prediction error. The decoding of each block of a frame consists in simply extracting the predictor, which is identified thanks to the motion vector, from the reference frame and adding the prediction residue. Both the motion vector and the residue are computed by the encoder. The decoding process, that is applied blockwise, cannot prescind from their complete transmission.

Despite that such systems are based on a blockwise decoding, it is not true that each block of a frame is independent from its neighbors. On the contrary, the structure of natural images generally imposes a strong spatial correlation among adjacent blocks, but such spatial correlation is not completely exploited in traditional system. A coding technique, able to reduce the redundancy between the already decoded part of the frame and the motion information, would improve the compression efficiency, possibly completely discarding the motion information from the transmitted bitstream.

Thanks to arithmetic coding and prediction techniques, motion information is nowadays compressed very efficiently. Nevertheless, especially for low bit rate video coding, the motion field still has a nonnegligible impact on the overall bit-rate. The idea of skipping the transmission of the motion information and reestimate it at the decoder has recently attracted an increasing interest. For example, in [1] an algorithm for motion derivation at the decoder side for the H.264/AVC codec is presented. This algorithm is based on a template similar to those used in texture coding.

AQ1

C. Tonoli (✉), P. Migliorati, and R. Leonardi
Department of Electronics for Automation, Signals and Communication Lab, University of Brescia, Brescia, Italy
e-mail: {name.surname}@ing.unibs.it

D. Giusto et al. (eds.), *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*, DOI 10.1007/978-1-4419-1674-7_19,

In this chapter, we propose a method for motion estimation at the decoder. The proposed approach relies on the knowledge of the prediction residue, transmitted by the encoder, and it is based on Least Square Error prediction. Preliminary simulation results seem to be very promising.

The chapter is structured as follows. In Sect. 2, a brief description of the use of motion compensation in predictive video coding schemes is given. The proposed algorithm is described in detail in Sect. 3, and simulation results are presented and discussed in Sect. 4. Concluding remarks are given in Sect. 5.

2 Motion Compensation in Traditional Video Coding Schemes

Predictive video coding is based on motion estimation at the encoder and motion compensation at the decoder. In this section, the basic ideas of predictive video coding are briefly introduced. The highlighted details will be useful in the following. For a more complete description of this topics, we refer the reader to [2–4].

In predictive coding, the suitable predictor for each block is determined at the encoder, usually performing a block based motion estimation. The prediction residue, i.e., the difference between the current block and its predictor, is computed and encoded. The information sent to the decoder includes the residue, together with the motion field.

The decoder reconstructs each frame operating in a strictly blockwise mode, since each block is reconstructed independently from its neighbors. The motion vector associated to the current block is used as an index for the set of possible predictors. Once the correct predictor has been identified, the prediction residue is decoded and added to the prediction values.

This method obviously requires that one motion vector is transmitted for each block. Due to the efficiency of modern entropy coding techniques, the transmission of the motion field is in general not very expensive in terms of bit-rate. Especially in high bit-rate coding, where DCT coefficients quantization is very fine, motion information represents a small part of the overall transmitted rate. Nevertheless, in low bit-rate coding the amount of rate assigned to the signal coefficient is lower, so the motion rate becomes much more important.

3 The Proposed Algorithm for Motion Estimation at the Decoder

In this section, the proposed method of motion estimation at the decoder side is presented.

First of all, the fundamental ideas are introduced, focusing on the definition of the side information. The algorithm is then outlined, sketching a structure that can be applied with different spatial coherence evaluation parameters. Finally, the LSE based parameter is introduced, and its computation is described in some detail.

3.1 Decoder Side Information

The term "Side Information" can be found very frequently in recent papers on video coding, and it often acquires different meanings, depending on which specific field we are looking at. To a very general extent, it refers to pieces of information that are not exactly the values of the coded signal, but a somewhat higher level correlated information, which is indeed crucial for the proper decoding of the signal. In particular, the centrality of the concept of side information and the way the side information is dealt with is one of the distinguishing elements of the Distributed Video Coding (DVC) paradigm (see, for example, [5, 6]). In this paradigm, each frame is encoded independently from the others. Due to such assumption of independent frame coding, the motion estimation is not performed at the decoder, and the motion field has to be inferred at the decoder side, based on the side information.

Borrowing the concept of side information from Distributed Video Coding, in this work, we assume that the side information for the current frame corresponds to the previously decoded frames. In more detail, since the encoder motion compensation algorithm is known, when the decoder begins to decode a frame block, it already has some knowledge about that block, in terms of correlated information. It is known for sure that the motion compensated predictor for that block belongs to the block matching reference frame, and, more precisely, to the search window. In fact, the motion vector in motion compensation behaves exactly as an index for the set of predictors corresponding to the search window. Since the reference frame has already been decoded, the set of candidate predictors for the current block is completely known. Equivalently, it is possible to say that the final reconstructed block will be one of these candidate predictors corrected with the received prediction residue for that block.

Moreover, we introduce an a priori hypothesis that, despite its generality, turns out to be true in the great majority of cases. We assume that the signal to be coded is characterized by "spatial continuity," i.e., edges preserve their continuity across the block boundaries. This means that, given the neighborhood of a block, it is possible to infer that the more suitable predictor in a candidate set will be the one that matches at best the neighborhood edges. See Fig. 1 for an example of a well matched and a bad matched predictor, respectively. If we assume that block decoding is performed in raster scan order, the causal neighbors of the current block have already been decoded. Therefore, it is possible to use the information carried out by the position of their edges to try to match the candidate predictors.

These remarks about the side information role will be the basis for the selection of a predictor for motion compensation in the absence of the motion vector, and for the consequent motion estimation at the decoder.

3.2 Outline of the Predictor Selection Algorithm

Let us consider a predictive coding scheme based on motion compensation, as described in Sect. 2. Our aim is to avoid the transmission of motion vectors,

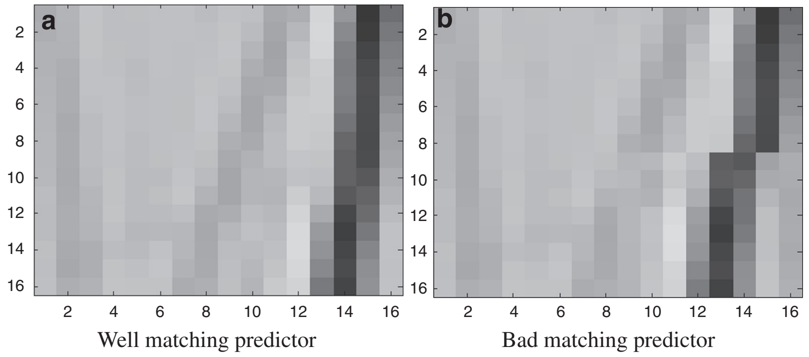


Fig. 1 Spatial continuity at block edges. (**a**) Well matching predictor. (**b**) Bad matching predictor

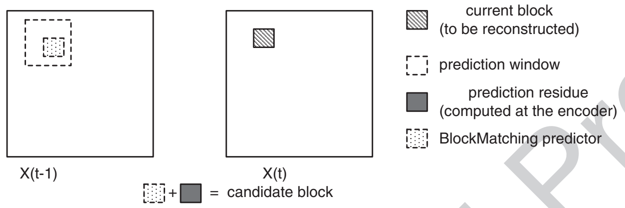


Fig. 2 Candidate set generation

nevertheless achieving a reconstruction quality close to that obtainable in case of transmission of the whole motion field. In order to try to do that, we apply the principles about side information described in Sect. 3.1.

For each block to be reconstructed, the set of candidate predictors is generated, as depicted in Fig. 2. A ranking of the candidates is then performed, in order to find out which candidates fits at best the coherence conditions, as described in the following lines. The already decoded causal neighborhood of the current block is considered. The macroblock composed by the current block and its three causal neighbors is constructed, replacing the current block with the tested candidate. A parameter p measuring the matching of the candidate block with the neighborhood is computed, in order to select the predictor that guarantees the best matching with the side information. Obviously, the matching parameter plays a crucial role in the algorithm performance, since it has to capture the matching of each candidate predictor and to select the most suitable one. In this chapter, we present a method based on Least Squared Error prediction. Such method will be described in more detail in Sect. 3.3. The reason why LSE prediction has been chosen to highlight the spatial coherence is that such technique is based itself on the exploitation of the correlation among adjacent pixels. The presented algorithm relies on the principle that a block correlated to the given neighborhood should be well predictable from the neighborhood, while AQ2 a less correlated block should produce a greater prediction error (Fig. 3).

Since we want to control at the encoder side the quality of the reconstructed signal, as it usually happens in predictive coding schemes, we apply our method

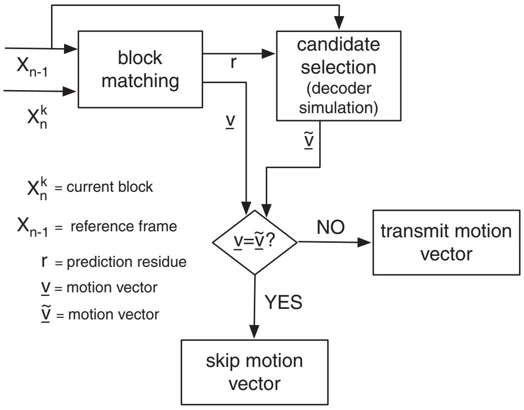
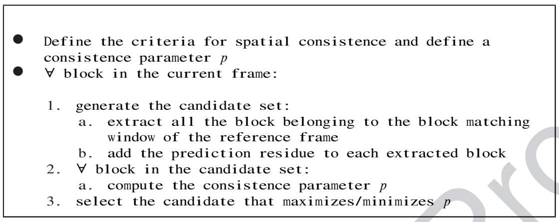

block matching
candidate selection (decoder simulation)
X_{n-1}
X_n^k
r
$\underline{v}$
$\tilde{\underline{v}}$
X_n^k = current block
X_{n-1} = reference frame
r = prediction residue
$\underline{v}$ = motion vector
$\tilde{\underline{v}}$ = motion vector
$\underline{v}=\tilde{\underline{v}}$?
NO
transmit motion vector
YES
skip motion vector

Fig. 3 Scheme of the proposed system



```
● Define the criteria for spatial consistence and define a
  consistence parameter $p$
● $\forall$ block in the current frame:

  1. generate the candidate set:
     a. extract all the block belonging to the block matching
        window of the reference frame
     b. add the prediction residue to each extracted block
  2. $\forall$ block in the candidate set:
     a. compute the consistence parameter $p$
  3. select the candidate that maximizes/minimizes $p$
```

Fig. 4 Algorithm structure

first at the encoder. In detail, for each block, the encoder simulates the operations of the decoder, and, based on the quality of the reconstructed block, decides whether the motion vector for that block is omissible or not. In our implementation, a simple threshold on the quality of the reconstructed block has been applied. A more precise rate-distortion analysis, like the one performed for example in the H.264/AVC encoder, could lead to a performance improvement because the effect of a motion vector skip on the overall reconstructed PSNR could be estimated more precisely.

3.3 Candidate Selection Based on Least Square Error Prediction

In the framework described in Sect. 3.2, in the absence of the motion information, the only criterion for the decoder to select one block among the candidates is the good match with the intra side information, i.e., the neighborhood.

The decoder-based motion compensation algorithm has been implemented according to the steps described in Fig. 4. As stated in step 2, each candidate needs to be tested in order to produce the parameter p, i.e., a "measure" of the correlation of that block with the known neighbors, and to get a ranking of the candidates. The steps to be performed to obtain such ranking are listed below.

1. Test each candidate block in the following way:

   (a) Prediction of the upper left quadrant; for each pixel, the correlation matrix is reestimated, based on the neighbors and on the true value of the past pixels in the block (i.e., the predicted pixels in the past are not considered in the estimation)

   (b) Compute the MSE on the upper left quadrant between the predicted block and the true candidate block

2. Choose the candidate that results to be more predictible, i.e., such that the MSE computed at the step 1 is smaller than that obtained for any other candidate (p = MSE).

The LSE prediction has been implemented as described in [7], where it is used to perform still image compression: each pixel is predicted, based on its causal neighborhood, and the prediction residue is encoded and transmitted. The prediction is shown to be orientation adaptive.

The LSE prediction computation is now briefly reported. Further details are given in [7]. For each pixel, a training window is set, as depicted in Fig. 5. According to the training values, the prediction coefficients are adaptively computed. The correlation matrix is estimated as described in the following.

$$c_i = [x_{i-1}, x_{i-2}, \dots, x_{i-L}] \tag{1}$$

where x_{i-j} is the jth causal neighbor of x_i, for $i = 1, 2, \dots, L$.

A $W \times L$ matrix, whose rows are the c_i vectors, is created:

$$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_W \end{bmatrix} \tag{2}$$

The covariance matrix is then computed as:

$$R_{xx} = C^T C; \tag{3}$$

while the covariance vector r_x is computed as

$$r_x = C^T y \tag{4}$$

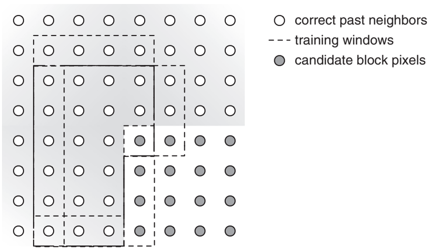


Fig. 5 Sliding training window

where

$$y = [x_{n-1}, \ldots, x_{n-L}]^T \tag{5}$$

According to the theory of least squared error prediction, the coefficient vector a is computed as

$$a = \left(C^T C\right)^{-1} \left(C^T y\right) \tag{6}$$

The main drawback of this algorithm is its huge computational complexity, due to the frequent matrix inversion operations that are needed. In the literature, several techniques have been presented to reduce the complexity [8]. In our implementation, the edge based technique presented in [7] has been used. It can be seen that the complexity can be reduced with a performance impairment of about 1% more wrong block.

4 Experimental Results

In this section, the performance of the proposed algorithm is presented and discussed.

In order to evaluate the proposed method, the percentage of correctly reconstructed motion vectors has been computed. As a groundtruth reference, the lossless case is considered. On the original CIF format sequence, the block matching is performed, on blocks of size 16 × 16, as a means to compute the motion field and the prediction residue. Since the work presented in this paper is aimed at exploring a new field, many optimizations have not been introduced yet: no multiple reference is considered for the block matching, and the reference for each frame is the previous frame.

When the prediction residue has been computed, the motion estimation method is applied and the percentage of correctly estimated motion vectors is computed for each frame. It is worth to remark that no error propagation is taken into account in the presented results. It is always assumed that the encoder controls the decoding process and, when a block cannot be correctly estimated in the absence of motion vector, the motion information is transmitted. So the neighbors of a block are always correct, either because their motion vector has been estimated correctly or because motion has been transmitted.

In Table 1, the results in terms of percentage of correct blocks is reported for the first two frames of four test sequences, namely Foreman, Mobile, Highway,

Table 1 Percentage of correctly predicted blocks in the lossless case

Sequence	Correctly estimated motion vectors (%)
Foreman	75.93
Mobile	34.67
Highway	84.92
Harbour	41.22

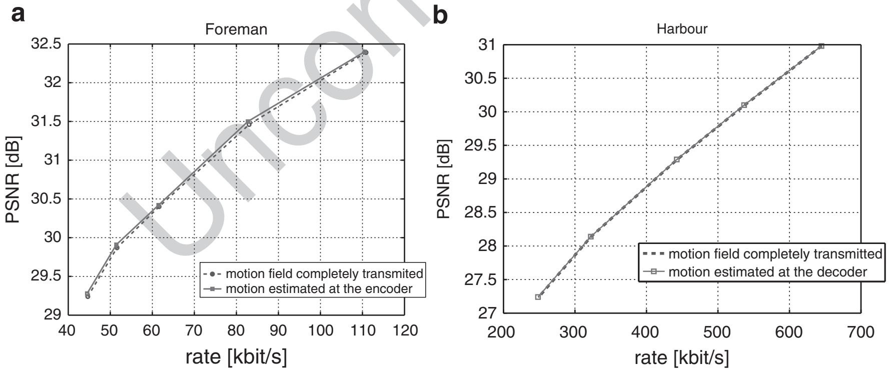
and Harbour. It can be noticed that the performance is strongly dependent on the sequence content. For Highway and Foreman, very good results can be achieved, whereas for Mobile and Harbour, the algorithm is less effective.

In order to give an idea of how the presented algorithm could perform in a realistic scenario, it has been applied to lossy coding. In particular, low bit rate coding has been considered because in this case, skipping the motion information could be particularly advantageous.

In more detail, the rate and PSNR values for the case of transmission of the whole motion field have been obtained using a simplified H.264 codec. The block size has been set to 16 and the considered prediction mode is P, i.e., mono-directional prediction, with a single reference picture. No deblocking has been performed on the reconstructed frame. An important remark has to be given about the rate estimation. The coding efficiency in modern predictive codec, such as H.264 codec, depends heavily on how arithmetic coding is performed. Since our method has not been really implemented in H.264 yet, it is impossible to measure exactly the rate savings. In order to produce a reliable estimate, the bits devoted to the motion transmission for each block have been computed, and, for the correctly predicted block, the result has been subtracted from the overall bit-rate. A signalling overhead has also been taken into account.

The estimated performance of the considered method is reported, for the first ten inter-frames (i.e., frames from 2 to 11, since the first frame is intra-coded) of the Foreman and Harbour sequence in CIF format, at 15 fps, in Fig. 6a and b.

In order to help a more precise performance assessment, the percentage of skipped motion vector is also reported, in Table 2. In the case of lossy coding, it can happen that the selected motion vector is not the correct one, but the selected candidate is not too different from the correct block. In this case, it can be seen that the proposed method can lead to a slight performance improvement.



PSNR curves for the Foreman sequence

PSNR curves for the Harbour sequence.

Fig. 6 PSNR curves for the test sequences Foreman and Harbour. (**a**) PSNR curves for the Foreman sequence. (**b**) PSNR curves for the Harbour sequence

Table 2 Percentage of correctly predicted blocks for the lossy compression of the Foreman sequence

PSNR	Percentage of skipped motion vectors
31.51	21.03
30.42	15.0
29.91	14.69
29.28	17.50

5 Conclusions

Compression efficiency is particularly important in small, low power devices. Side information at the decoder side, i.e., correlated information about the signal that has to be decoded, can be exploited to improve compression efficiency in predictive video coding. Starting from the side information, the encoder can infer important knowledge that helps in decoding the signal. As an example of how side information at the decoder side can be exploited in video coding, in this chapter, we have proposed a method that partially avoids the transmission of motion vectors in predictive video coding schemes based on motion compensation. Simulation results show that the proposed approach can lead to bit-rate savings.

References

1. Kamp S, Evertz M, Wien M (2008) Decoder side motion vector derivation for inter frame video coding. In: Proceedings of 15th IEEE international conference on image processing 2008 (ICIP 2008), 12–15 Oct 2008, pp 1120–1123
2. Wiegand T, Sullivan G, Bjontegaard, G (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576
3. AQ3 Wiegand T, Sullivan G, Luthra A (2003) Draft ITU-T recommendation and final draft international standard of joint video specification
4. Kappagantula S, Rao KR (1985) Motion compensated interframe image prediction. IEEE Trans Commun 33(9):1011–1015
5. Aaron A, Zhang R, Girod B (2002) Wyner-Ziv coding for motion video. In: Proceedings of 36th Asilomar conference on signal, systems and computers (ACSSC '02), Pacific Grove, CA, vol 1, pp 240–244
6. Puri R, Ramchandran K (2003) PRISM: a new reversed multimedia coding paradigm. In: Proceedings of the IEEE internatinal conference on image processing, Barcelona, Spain, Sept 2003
7. Li X, Orchard MT (2001) Edge directed prediction for lossless compression of natural images. IEEE Trans Image Process 10(6):813–817
8. Li X, Orchard MT (2002) Novel sequential error-concealment techniques using orientation adaptive interpolation. IEEE Trans Circuits Syst Video Technol 12(10):857–864
9. AQ4 Wu X, Barthel K (1998) Piecewise 2D autoregression for predictive image coding. In: Proceedings of the IEEE international conference on image processing, Chicago, IL, Oct 1998, vol 3, pp 901–904
10. Park JW, Kim JGK, Lee SU (1997) DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error. IEEE Trans Circuits Syst Video Technol 7(6):845–854