Improving the Performance of a Software MPEG Coder by Using Efficient Displacement Estimation

Bruno Carpentieri

Dipartimento di Informatica ed Applicazioni "R. M. Capocelli"

Universita' di Salerno

84081 Baronissi (SA), Italy

Abstract: - Motion estimation is a crucial step for an MPEG coder. In this paper we show how it is possible to improve significantly the performances of a standard, software, MPEG coder by applying some of the ideas developed in [2] and [3].

Key-Words – Video Compression, MPEG, Split-Merge Displacement Estimation

1 Introduction

Data compression techniques offer today the possibility of storing and transmitting the huge amount of data that is necessary to represent digital videos.

Digital video is intrinsically redundant. One of the aspects of this redundancy is the strong correlation between parts of consecutive frames. For example, during a video telephone conversation the background does not generally change completely from frame to frame, and neither the color and shape of the dresses, glasses, eyes, etc. of the person in front of the camera. This *interframe* redundancy can be exploited by coding parts of a frame in terms of their position or "motion" with respect to a previous frame. The *intraframe* redundancy, instead, is the redundancy intrinsic to any still frame and it is handled by standard image compression strategies.

The standardization efforts and the MPEG standards have set a common platform for today video compression technology. In this paper we describe a software MPEG coder, derived by the public domain PVRG-MPEG coder. This new coder improves significantly the performances of the original PVRG-MPEG coder by implementing some of the ideas that are the basis of the Split-Merge Displacement Estimation Technique (see [2] and [3]).

This new variation of the PVRG-MPEG codec, the *Split*-MPEG coder is available, for testing purposes, on the web page *http://www.unisa.it/bc.dir/CSCC99*.

In the next two sections we review the Split-Merge technique and the PVRG-MPEG codec. In section 4 we present the *Split*-MPEG coder, in Section 5 we show our experimental results. Section 6 is devoted to our conclusions and future research directions.

2 Split-Merge Displacement Estimation

Block matching displacement estimation algorithms divide the image into a number of rectangular blocks and compute a displacement vector for each block by correlating the block with a search area in the previous frame: if the blocks are small enough, rotation, zooming, etc. of larger objects can be closely approximated by a translation of the blocks themselves. The goal is to approximate interframe motion by piecewise translation of one or more areas of a frame relative to a reference frame. This technique was introduced by Jain and Jain in [1]. Block matching displacement estimation is simple: it does not require any semantic knowledge of the frames but reduces the motion estimation problem to a matching problem. In fact a semantic analysis of the frames, that identifies and "understands" the objects in each frame and their relationship from frame to frame, is generally a difficult task that is not often practical today for video coding purposes.

The lack of knowledge of the spatial relationship between pixels is also a limitation of the block matching technique: the prediction frames obtained via the displacement vectors do not always maintain the original intraframe correlation: each block is supposed to be undergoing an independent translation; moreover the optimal displacement vector for a block is not always unique and it might be influenced by noise. Because of this, the reconstructed frame might loose a large part of the original spatial correlation between adjacent blocks. Today video standards (MPEG, Px64 etc.) are based on the block matching approach.

In [2] we have presented a new approach to displacement estimation: the Split-Merge displacement estimation technique, which exploits the temporal correlation between frames while preserving the spatial correlation between the parts of the frame itself. This technique is based on the segmentation of the frames into areas (superblocks) moving in the same direction. Each superblock is a connected collection of fixed-size rectangular blocks. This new technique does not require any additional semantic knowledge of the frames but computes the motion vectors through standard block matching. The segmentations obtained from frame to frame retain the "knowledge" accumulated in the previous frames on the spatial relationship between blocks, they improve the quality of the predictions and exploit the performance of the video coder by identifying the "active" parts of the frames: the *splits*. This technique is a bridge between the purely semantic approaches to motion estimation and the, purely syntactic, standard block matching approach. Split-Merge displacement estimation ``learns" from previous frames and performs a classification of the parts in a video frame in terms of their motion vectors. To do this there is no need of any "a priori" knowledge on the sequence, nor of any sophisticated preprocessing analysis of the frames. It just makes use of standard block matching strategies and of a wise management of the collected information. For more details see [2] and [3].

This new displacement estimation technique is compatible with the MPEG philosophy and can be the basis of MPEG video coders.

3 MPEG

The name MPEG is an acronym for Moving Picture Experts Group, a group formed under the auspices of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). This group is responsible for three different standards in video coding (MPEG-1, MPEG-2 and MPEG-4). MPEG-1 (see [4]) is intended for data rates on the order of 1.5 Mbit/s, MPEG-2 (see [5]) is intended for higher data rates (10 Mbit/s or more) and MPEG-4 is intended for very low data rates (64 Kbit/s or less). In this paper we will consider only MPEG-1 referring to it simply as MPEG.

MPEG relies on two basic techniques: block-based motion compensation for the reduction of the temporal redundancy and then transform domain based compression (DCT) to exploit the redundancy in the spatial directions. The DCT's are done on blocks of 8×8 pixels, and the motion prediction is done in the luminance channel (Y) on blocks of 16×16 pixels (macroblocks).

Motion compensated prediction assumes that the current picture can be modeled "locally" as a translation of the picture at some previous time. Locally means that the amplitude and the direction of the displacement need not to be the same everywhere in the picture. The MPEG standard does not specify how the displacement vectors are to be computed. Because of the block-based motion representation, however, block matching techniques are likely to be used.

From the point of view of the decompressor there are three types of coded frames. There are *I* frames (intra frames) that are simply frames coded as still images, not using any past history. Then there are *P* frames (predicted frames) that are predicted from the most recent *I* or *P* frame. Each macroblock in a *P* frame is coded either as a displacement vector with respect to the previous frame (and a DCT coded error) or just "intra" coded (as in the *I* frames) if there was no good match. Finally there are *B* (bidirectional) frames, that are the results of interpolations from the closest two *I* or *P* frames, one in the past and one in the future.

3.1 The PVRG-MPEG Codec

The PVRG-MPEG codec is a public domain video encoder and decoder that was generated according to the Santa Clara August 1991 format. The codec is capable of encoding all MPEG types of frames. The algorithms for rate control, buffer-constrained encoding, and quantization decisions are similar to those of the (simulation model 1-3) MPEG document. The rate control used is a simple proportional Q-stepsize/Buffer loop. The MPEG codec performs compression and decompression on raw raster scanned *YCbCr* (also known as digital *YUV*) files. This package is implemented in software for simulation purposes, it is available by anonymous ftp from *havefun.stanford.edu*.

The motion estimation of the PVRG-MPEG codec compares, through full search, a 16 by 16 macroblock (consisting of four 8 by 8 blocks) in the luminance throughout a small search area of a previously transmitted *I* frame or *P* frame. The default range for such comparison is set between ±8 pixels based on the luminance component of the image at half-pel accuracy. The motion-compensation vector chosen for a given macroblock is the one that minimize the absolute difference in the search area.

The PVRG-MPEG codec uses, as default frame group, a sequence *IBBPBBI*. This is shown in Fig. 1 (the arrows depicts the dependency of the *B* frames on the *P* and *I* frames and the dependency of the *P* frames only on the *I* frames). Therefore the *I* frames are coded individually without any temporal prediction, the *P* frames are forward predicted and the *B* frames are interspersed between the *P* and the *I* frames. The bidirectionally predicted frames can be considered motion-compensated interpolation between the predicted and the intra frames.

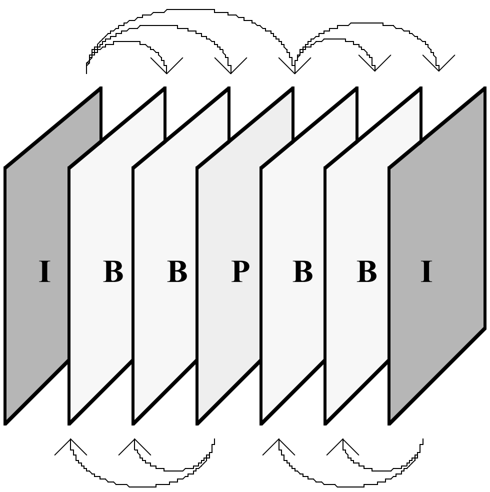


Fig.1 *Default Frame Group of the PVRG-MPEG Codec*

The presence of the *I* frames is motivated by the need of maintaining a stable quality (by using only *P* frames the quality of the coded sequence would rapidly decrease) and by the need that some applications have of allowing random access points not depending on the previous history of the sequence.

We have improved the motion estimation procedures of the PVRG-MPEG coder by implementing some of the ideas developed by the Split-Merge technique. The result has been the new MPEG coder described in the next Section.

4 Improving the PVRG-MPEG Coder

We have modified the PVRG-MPEG coder. The goal was to improve its performances by designing a new MPEG coder based on some of the ideas developed in [2] and [3]. Of course the output of this new coder shall still be an MPEG file, decodeable by any standard MPEG player.

The power of the Split-Merge technique is in the fact that its segmentation of the current frame in *splits* and *superblocks* permits both the retention of the knowledge accumulated in the previous frames and also the identification of the active parts of the sequence. In order to make the output of the new coder fully readable by an MPEG player, we need to give a new definition of *superblocks* and *splits*.

> ***Def.*** ∞ A *16 x 16* macroblock is in a **superblock** if its motion vector is equal to the motion vector of one of its adjacent macroblocks.
>
> ***Def.*** ∞ A *16 x 16* macroblock is a **split** either if it was in a superblock in the previous frame and its current motion vector now is not equal to the motion vector of any of its previous companions, or if it was a split in the previous frame and now it is not in a superblock.

We can now define an *S* frame:

> ***Def.*** ∞ A Predicted-*S* frame (or simply an *S* frame) is a frame in which every macroblock that is in a *superblock* is predicted and motion compensated by using the previous *S* frame, and every macroblock that is a *split* is intra-coded.

The *Split*-MPEG coder will only code *B* and *S* frames. Therefore instead of having *IBBPBBIBBPBBI...* frame sequences, as with the standard PVRG-MPEG coder, with the *Split*-MPEG coder we have *IBBSBBSBBSBBS...* sequences. In an *S* frame, if more than two thirds of the macroblocks are *split*s then the frame is intra-coded

5 Experimental Results

We have implemented a new -split option of the PVRG coder implementing this new *Split*-MPEG coder (i.e. the original PVRG coder plus a new procedure split.c that allows the -split option and takes care of the segmentation in splits and superblocks and allows the usage of the *S* frames)..

We have compared the performances of the PVRG-MPEG coder with the *Split*-MPEG coder by testing the two coders on the test sequences described in the next subsection.

5.1 The Test Sequences

In our experiments we have used four standard MPEG test sequences: *Salesman, Birdshow, Ghostpen, Joel:*

Salesman

This is a 235 frames sequence, 8 bits of grayscale per pixel. Each frame is 360 x 288.

Birdshow

This is a 60 frames color sequence. Each frame is 160 x 128.

Ghostpen

This is a 60 frames color sequence. Each frame is 320 x 200.

Joel

This is a 100 frames color sequence. Each frame is 160 x 128.

In **Appendix 1** we show the first and the last frame of each of the test sequences

5.2 Experimental Results

For each of the test sequences we have compared the *Split*-MPEG coder performances and the performances of the original PVRG-MPEG coder.

The tables shown in **Appendix 2**, one for each of the test sequences, show the compression ratios (number of bytes of the original file / number of bytes of the compressed file) obtained by the two coders on three different levels of quality ("Fair", "Good", "Very Good"). The measure of quality we have used is the average SNR (in *db*s) computed on the luminance frames of the *Y*, *U* and *V* sequences. The three different qualities have been obtained by modifying (via the -q option of the PVRG-MPEG coder) the quantization levels. The quantization values used to obtain the "Fair" , "Good" and "Very Good" rows are respectively q = 22, q = 9, q = 2. For completeness, in the SNR columns, we have indicated the average SNR values for all the *Y*, *U* and *V* frames. A subjective evaluation has confirmed that, at the same SNR, the visual quality of a sequences encoded by the *Split*-MPEG coder is at least as good as the visual quality of the same sequence encoded by the original PVRG-MPEG coder.

Moreover the visual quality (and the SNR) of the *Split*-MPEG coded sequences remains stable and do not degrade over time.

As shown by the tables, the improvement in compression is significant for the "Fair" and "Good" row. In fact we are almost doubling the PVRG-MPEG performances. This mean that our algorithm is particularly efficient at a low bit rate. When the bit rate grows, the number of macro-blocks that have to be sent intra-coded to reach the desired quality increases, therefore the compression difference between the standard coder and the *Split*-MPEG coder decreases.

In terms of computing speed, we have experimentally seen that the *Split*-MPEG coder is at least as fast as the PVRG-MPEG coder.

5 Conclusions and Future Research Directions

We have modified the public domain PVRG-MPEG codec by applying some of the ideas presented in [2] and [3]. The resulting *Split*-MPEG coder has shown better experimental performance with respect to the original PVRG-MPEG coder. At low bit rate it has almost doubled the original compression ratio. The only price paid is the loss of the *I* frames and therefore the lack of fixed synchronization or "random access" frames. This is not a big price for most of the application that are commonly using MPEG-1: a fresh new "entry point" in the video sequence will be probably easily found by examining only a few seconds of video, in fact it is very likely that in a real-life application, like a movie recorded on a CD or an MTV video-clip, after a few seconds we will see a change of scene or many new objects entering the visual field.

Future research involve the development from scratch of a new MPEG software encoder based on the Split-Merge technique.

6 Acknowledgments

We thank our student Giselda De Vita for helping with the implementation and the experimental tests.

References:

[1] J. R. Jain and A. K. Jain, "Displacement Measurement and its Application in Interframe Image Coding", *IEEE Trans. on Communications*, Vol. COM-29, n.12, pp. 1799-1808, Dec. 1981.

[2] B. Carpentieri and J. A. Storer, "Split-Merge Parallel Video Displacement Estimation", *Proceedings of the IEEE*, Vol. 82, n. 6, pp 940-947, June 1994.

[3] B. Carpentieri, "A Bridge between Block Matching and Semantic (Model Based) Motion Estimation", Proceedings of the *13th International Conference on Digital Signal Processing* (*DSP97*),

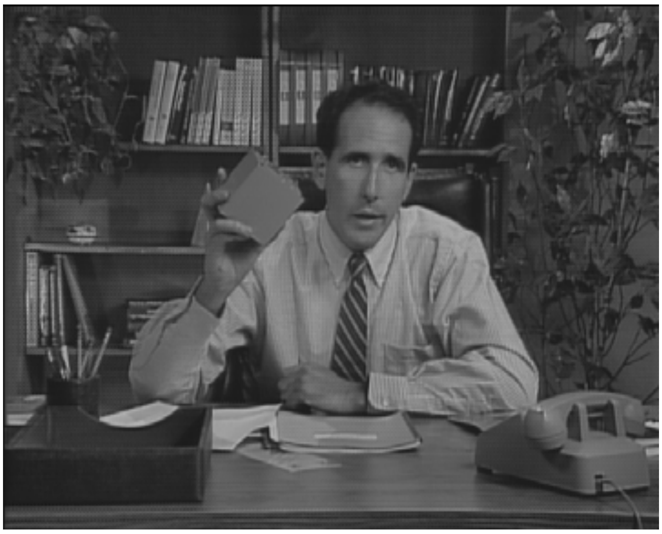
July 1997, Santorini (Greece), pp. 1131-1134, IEEE Press.

[4] J. L. Mitchell, W. Pennebaker, C. E. Fogg and D. J. LeGall, *MPEG Video Compression Standard*, Chapman and Hall, 1996.

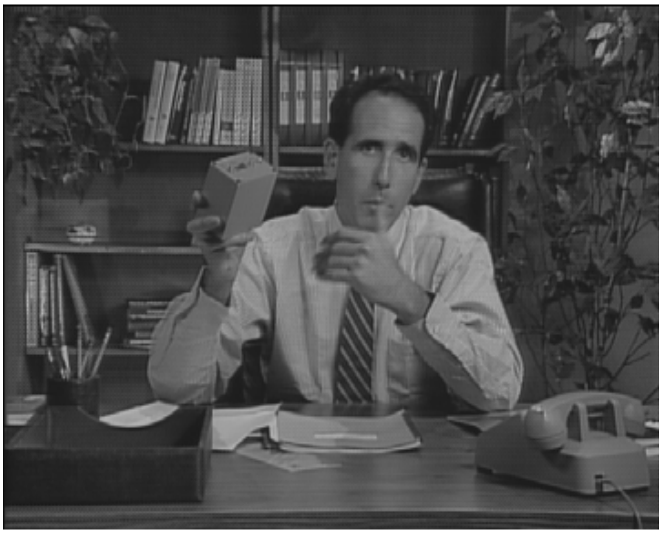
[5] B. G. Haskell, A. Puri, A. N. Netravali, *Digital Video: an introduction to MPEG-2*, Chapman and Hall, 1996.

APPENDIX 1: TEST IMAGES

Salesman

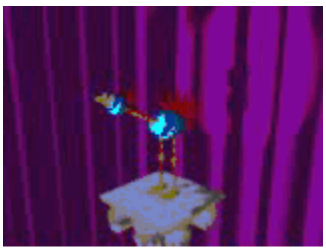


frame n°1

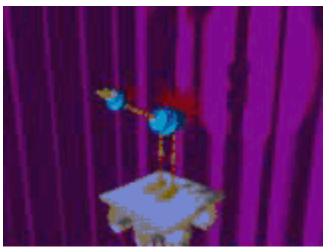


frame n°235

Birdshow

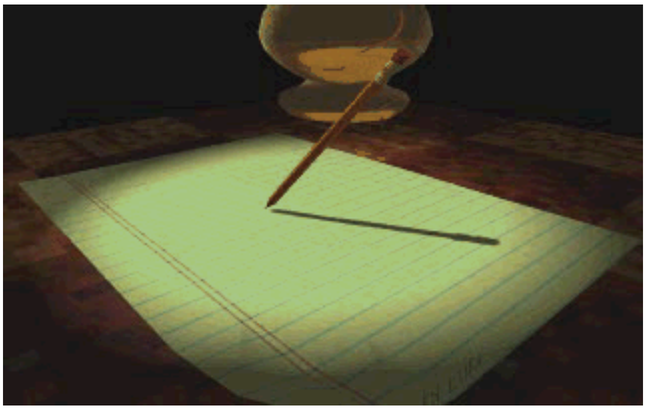


frame n°1

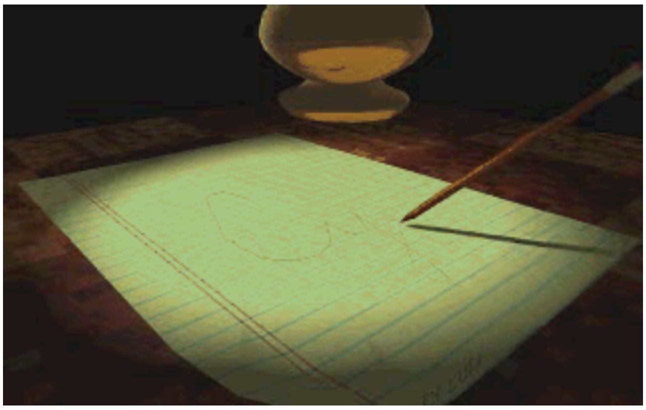


frame n°60

Ghostpen

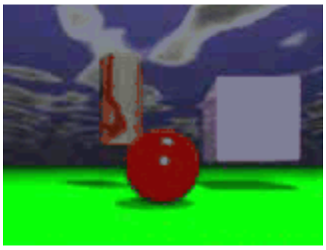


frame n°1

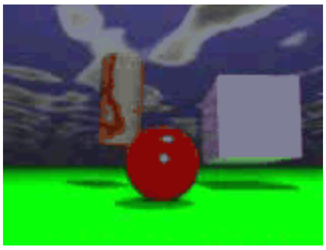


frame n°60

Joel



frame n°1



frame n°100

APPENDIX 2: RESULTS

- SALESMAN

	PVRG-MPEG Compression Ratio	*PVRG-MPEG* SNR	*Split-MPEG* Compression Ratio	*Split-MPEG* SNR
"FAIR"	**68.42**	23.14 46.06 45.83	**120.39**	23.10 45.95 45.79
"GOOD"	**25.55**	28.60 46.00 45.87	**32.92**	28.73 45.93 45.79
"VERY GOOD"	**6.08**	34.93 42.84 43.16	**7.88**	34.74 42.20 42.29

- BIRDSHOW

	PVRG-MPEG Compression Ratio	*PVRG-MPEG* SNR	*Split-MPEG* Compression Ratio	*Split-MPEG* SNR
"FAIR"	**65.51**	24.41 32.73 32.71	**104.04**	24.46 32.82 32.71
"GOOD"	**40.73**	28.20 36.64 36.15	**61.22**	27.98 36.32 35.84
"VERY GOOD"	**14.48**	38.93 47.90 47.67	**18.81**	38.61 47.04 47.09

- GHOSTPEN

	PVRG-MPEG Compression Ratio	*PVRG-MPEG* SNR	*Split-MPEG* Compression Ratio	*Split-MPEG* SNR
"FAIR"	**77.07**	27.97 32.39 33.57	**122.35**	27.54 31.86 33.13
"GOOD"	**41.02**	34.01 38.36 39.81	**70.12**	36.65 38.04 39.52
"VERY GOOD"	**12.03**	43.02 45.05 46.36	**16.93**	42.72 44.71 46.02

- JOEL

	PVRG-MPEG Compression Ratio	*PVRG-MPEG* SNR	*Split-MPEG* Compression Ratio	*Split-MPEG* SNR
"FAIR"	**40.37**	29.09 31.76 30.52	**54.62**	28.93 31.42 30.19
"GOOD"	**21.76**	36.49 47.66 39.55	**27.90**	36.37 39.01 37.59
"VERY GOOD"	**8.24**	43.12 44.89 45.24	**9.93**	44.88 44.58 44.60