Efficient MV Prediction for Zonal Search In Video Transcoding

S. Marcelino [1], S. Faria [2], P. Assunção [3]

ESTG - Polytechnic Institute of Leiria / Instituto de Telecomunicações
Leiria, Portugal
[1] stam1985@gmail.com
[2,3] {sfaria,assuncao}@estg.ipleiria.pt

S. Moiron [4], M. Ghanbari [5]

University of Essex,
Colchester, United Kingdom
[4,5] {smoiro,ghan}@essex.ac.uk

***Abstract*—This paper proposes a method to efficiently find motion vector predictions for zonal search motion re-estimation in fast video transcoders. The motion information extracted from the incoming video stream is processed to generate accurate motion vector predictions for transcoding with reduced complexity. Our results demonstrate that motion vector predictions computed by the proposed method outperform those generated by the highly efficient EPZS (Enhanced Predictive Zonal Search) algorithm in H.264/AVC transcoders. The computational complexity is reduced up to 59.6% at negligible cost in R-D performance. The proposed method can be useful in multimedia systems and applications using any type of transcoder, such as transrating and/or spatial resolution downsizing.**

I. Introduction

In the past few years there has been an exponential growth in video services and applications along with different terminals and multimedia transport channels. The increasing diversity of terminal equipment and networks raised the need for adaptation functions in order to match the multimedia content to the specific characteristics of the target technology. This includes, for example, the video display resolution, terminal computational/energy resources or channel bandwidth. Such functions are accomplished through the use of video transcoders operating in network nodes along the communication path. Among the most common transcoding functionalities, one can find bitrate conversion (i.e., transrating) or spatial rescaling for mobile video services, e.g., mobile TV [1]. Since in general there are many streams to be adapted at the same network node, this is usually accomplished through the use of several real-time transcoders running simultaneously and sharing the same hardware platform. Therefore, minimising transcoding complexity is necessary to optimize the available processing resources and reduce power consumption.

Motion estimation (ME) is known to be one of the most demanding functions in standard video encoders. This is particularly relevant in H.264/AVC (Advanced Video Coding), where multiple reference frames and different block partitions are allowed for motion search [2]. Since in transcoding, most of the video signal properties, including the motion flow, are maintained throughout the processing chain, the general rule for finding new motion vectors (MV) is to reuse the motion information already embedded in the incoming stream. This is used even in the case of heterogeneous transcoding where different coding standards are involved [3], [4]. However, the incoming MVs should not be directly used in the transcoded video without further processing because they become inaccurate motion descriptors after processing the signal. Therefore MV processing is necessary in video transcoders, which includes either temporal or spatial scaling and refinement.

Zonal search ME is particularly suited for video transcoding because the MV search process starts from a set of precomputed MV candidates, which can also be efficiently computed from the MVs available in the video stream to be transcoded. Despite the fact that reduction of computational complexity is implicit in many transcoding methods, only recently the Enhanced Predictive Zonal Search (EPZS) algorithm was considered for such purpose. In [5] the authors propose an EPZS-based method for transcoding between MPEG-2 and H.264/AVC. However, the more complex case of homogeneous H.264/AVC transcoding, where different reference frames can be used with various block partition sizes, was not addressed yet in the context of zonal search. Other recent work addressed the problem of fast transcoding with spatial resolution reduction [6] and [7]. Both approaches achieve significant computational complexity reduction but at the expense of considerable drop in the objective signal quality. In [6], the transcoded video suffers from a PSNR loss up to 1.5dB, while in [7] the objective image quality loss reaches 1dB.

In this paper we propose a method to find efficient MV predictions for zonal search algorithms in video transcoding

MMSP'10, October 4-6, 2010, Saint-Malo, France.
???-?-????-????-?/10/$??.??

systems and applications. A modified version of the original EPZS algorithm is used, where the MVs from the incoming video stream are used to find efficient MV predictions. The number of initial MV predictions is reduced to a maximum of two, which is shown to greatly reduce the number of checking points, i.e, block comparisons, needed to find the best MV. Our results show that MV predictions obtained through the method proposed in this paper are more accurate than those used in the EPZS algorithm currently implemented in the H.264/AVC reference codec. Overall, the proposed method is capable of significantly reducing the computational complexity of ME at negligible cost in R-D performance when compared with the current state-of-the-art EPZS algorithm. It also exhibits a better performance than similar transcoders recently proposed in [5], [6] and [7].

This paper is organised as follows. In the next section, a brief overview of MV prediction in zonal search algorithms is provided. In section III, the proposed method is described while the experimental results are presented and discussed in section IV. Finally, section V concludes the paper.

II. MV PREDICTION IN ZONAL SEARCH ALGORITHMS

Zonal search motion estimation was introduced as a much more efficient method to find accurate MVs than other MV search methods, such as three-step search (TSS) and diamond search (DS) proposed some years ago [8][9]. When compared with DS and TSS, zonal search algorithms are able to achieve a similar R-D performance with much lower computational complexity [10]. These higher efficiency algorithms not only significantly reduce the number of checking points examined in the search process, but can also retain, and even in many cases improve, the video quality. Zonal search algorithms can accomplish this by initially computing several highly likely predictors, and by using very reliable early-stopping criteria to terminate the search procedure at any checking point. This is improved even further by using very efficient checking patterns, which optimise the whole process.

This type of ME algorithms were improved in the recent years giving rise to the EPZS algorithm, which is considered a good option for ME in H.264/AVC encoding. As a result, it has been adopted in the H.264/AVC reference software (JM15.1) [11] and remains as the flagship ME algorithm.

Predictor selection is a factor of major importance in the good performance of EPZS algorithms. In general, a set of several MV predictors is computed from previously coded neighbour blocks in both the current and previous frames, exploiting different types of motion characteristics such as acceleration and direction, and taking into account their inherent correlations. The median predictor is often considered the most likely candidate to be the optimal predictor and can be considered on its own as a predictor candidate. Other candidates are also included such as the (0,0) MV and the accelerator MV. The accelerator MV is found by considering not only the MV of the co-located MB in the previous frame, but also that of the frame before. The idea behind this predictor is that motion velocity may not be constant but accelerating instead.

Then the best predictor is found among previously computed candidates, defining the search window centre and the initial MV cost. This cost is updated whenever a lower one is found at any spatial position which results in better R-D efficiency. The motion search process is stopped by either the early-termination mechanism, when the current MV cost is smaller than a predefined threshold, or when the ME position reaches the search window boundaries. The threshold is defined according to equation 1, where the minimum Sum of Absolute Differences (SAD) is computed for all candidate blocks and a_k, b_k are scaling and shifting parameters [10].

$$T_k = a_k \times min(MSAD_1, MSAD_2..., MSAD_n) + b_k \quad (1)$$

By using the scaling and shifting parameters a_k, b_k on the thresholding T_k, the number of stopping cases with low error can be increased, while roughly keeping the number of cases with significant error.

III. METHOD TO FIND EFFICIENT MV PREDICTIONS FOR ZONAL SEARCH IN TRANSCODING

The proposed method aims at further reducing the computational complexity of EPZS algorithm in video transcoders, by minimising the number of both the initial predictors and checking points. This is achieved by computing a predictor obtained from the set of MVs associated with spatially adjacent blocks in the current slice (previously transcoded) and adding a new one computed from the motion vector field of the incoming coded stream. The challenge is to prove that these two predictors are good enough to obtain lower complexity and better efficiency than all others normally used in the original EPZS algorithm. The proposed method comprises four main steps: (*i*) extraction of MVs and creation of a motion vector field for each slice, (*ii*) normalization of these MVs, (*iii*) computation of predictors and (*iv*) MV search to find the best match according to a predefined stop criterion. Two types of transcoding functions are envisaged: rate reduction and spatial downsizing, *i.e.*, transrating by keeping the spatial resolution and transrating with reduction of spatial resolution.

A. Extraction of motion information

In the first step, the input video stream is parsed by the transcoder and the embedded motion vectors are extracted and stored in a memory buffer as a motion vector field, where each single 4×4 block partition is assigned a corresponding MV. Since each incoming MV may be associated to block partitions of different size (e.g.,4×4, 8×8, 8×16, etc), such motion vector field is organised as a matrix, where each element is one MV associated to a single 4×4 partition. This means that those MVs associated to block partitions greater than 4×4 are replicated for each 4×4 subpartition included in the block.

B. Motion vector normalisation

Since H.264/AVC can use multiple temporal references, those that are temporally farther from the current slice produce longer MVs than closer ones. Therefore, since the selection of a candidate MV predictor involves comparisons between different MVs, prior normalisation must be done.

The normalisation process converts all MVs to the same temporal reference by scaling those MVs from distant reference slices according to their corresponding temporal distances. The normalised MV, $V_T(i)$, is obtained by using equation 2, where $v_t(i)$ is the MV pointing to reference i and C_f, R_{fn} are the temporal positions of the current and references slices, respectively. N_{MV} is the number of MVs.

$$V_T(i) = \frac{v_t(i)}{|C_f - R_{fn}|}, i = 1..N_{MV} \quad (2)$$

Note that MV normalisation is common to both types of transcoding, *i.e.*, with and without spatial downsizing.

C. Predictor Computation

A first predictor V_o is always computed for the current block, as the median of the set of normalised MVs associated with the N previously transcoded neighbour blocks, as given by equation 3.

$$V_o = median\{V_T(i), i = 1..N_{MV}\} \quad (3)$$

Figure 1 shows an example of the neighbour blocks involved in this process: Left (L), Upper (U), Upper-Right (UR) and the corresponding MVs $v_t(i), i = 1, 2, 3$. The predictor V_o is shown on the current block.

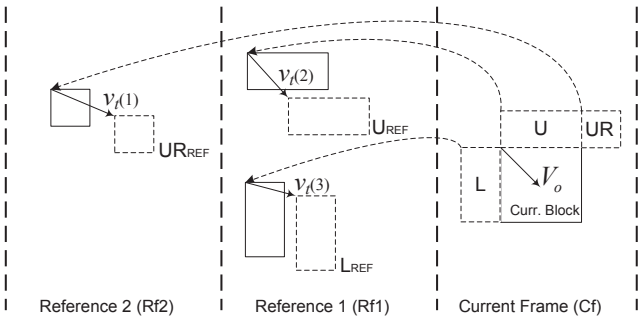

Fig. 1. Contributing MVs from neigbour blocks with different references

In the case of transcoding without spatial downsizing, the second predictor V_{So} is computed as the median of a set of MVs taken from the motion vector field previously built from the incoming video stream. Such set of MVs include those that are associated to all 4×4 block partitions of the incoming video, which are spatially coincident with the current block. As pointed out before, the transcoding mode of the current block may define partitions of greater size than those originally used in the incoming video.

In the case of transcoding with spatial resolution downsizing, the second predictor is computed taking into account that several different block partitions of the incoming video might match a single block in the corresponding lower resolution video signal. In this case, the MVs from the incoming higher resolution slice $v_s(i)$ are scaled down according to the spatial resolution reduction ratio r, and the resulting prediction V_{So} is computed as the median of the downscaled MVs, as given by equation 4. The number of MVs involved, k, is variable according to the number of block partitions.

$$V_{So} = median\frac{\{v_s(i), i = 1..k\}}{r} \quad (4)$$

Figure 2 shows an example of transcoding with spatial downsizing, where 5 MVs contribute to determine the predictor V_{So}.

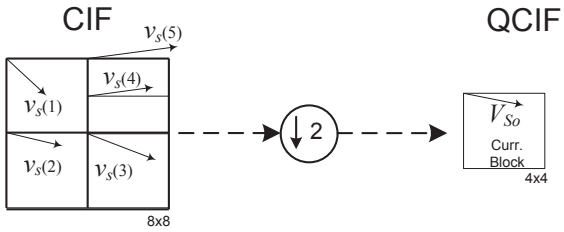

Fig. 2. Contributing MVs from higher spatial resolution

D. MV search using new predictor

The final step consists in searching for the best MV by using the predictors in the EPZS ME algorithm, taking advantage of its features such as refinement and adaptive early termination. Instead of using the original EPZS predictors mentioned in section II, the proposed method replaces all original EPZS predictors by only the two ones described in the previous subsection. Therefore, this method reduces the initial number of predictors to be checked and adds a new highly efficient predictor derived from the incoming video stream. As it shall be seen in the results, such a predictor can significantly reduce the number of checking points.

Then the best predictor for the current block is found by minimising equation 5, where **m**=$[m_x, m_y]^T$, **p**=$[p_x, p_y]^T$ are the predictors under test, x, y is the spatial position and λ_{MOTION} is the Lagrange multiplier [11]. The term $R(m-p)$ represents the cost of motion information.

$$J(m, \lambda_{motion}) = SAD(s, c(m)) + \lambda_{motion} \cdot R(m - p) \quad (5)$$

The SAD is computed using equation 6, where **s** is the input video signal and **c** is the reference one after decoding.

$$SAD(s, c(m)) = \sum_{x=1,y=1}^{B,B} |s[x, y] - c[x - m_x, y - m_y]| \quad (6)$$

$$B = 16, 8, 4$$

Finally the MV search for each transcoded block proceeds by following a diamond search pattern which stops as previously described.

IV. Experimental Results

The performance of the proposed method (**Proposed**) was evaluated by comparing with two reference transcoders with different motion estimation algorithms, the classic Full Search (FS) algorithm and the EPZS. These two methods are compared with the algorithm proposed in this work. Two different experiments were carried out in order to evaluate the performance of the following transcoding functions: (*i*) transcoding for transrating (keeping the spatial resolution) and (*ii*) transcoding for spatial resolution reduction. These transcoders were implemented based on the JVT reference software JM15.1 using the baseline profile [11]. The original streams were encoded in CIF format, setting the quantization parameter (QP) to 28 for I and P frames, and disabling skip mode and intra blocks in P frames. Three sequences with different motion and spatial activity were used: *Akiyo, Bus* and *Stefan*.

In order to evaluate the performance of the proposed method two types of metrics were used: (*i*) computational complexity and (*ii*) R-D efficiency. The complexity was measured as the total number of SAD computations, while R-D performance was measured as the bit rate (kbps) and the PSNR of luminance for different QPs lying within an acceptable range of quality, i.e., 30, 34, 38 and 42. The "extended diamond" search pattern was used since this is known to have very good complexity/performance ratio [12]. In order to compare the proposed method with the original EPZS, three different simulations were carried out for all sequences:

- **FS Ref.**: We assume this method as the original for comparison, where we intended to do a comparison with one of the most known motion estimation algorithms. In this experiment, we use a 32 pixel search windows.
- **EPZS Ref.**: This is the EPZS reference method used for comparison. All default settings of the EPZS algorithm were used (window based predictors, temporal predictors, spatial memory predictors and block type predictors); III-C;
- **Proposed**: This is the proposed method. Although based on the original EPZS, all default EPZS predictors were replaced by the proposed predictors as described in the previous section.

A. Transcoding complexity

Table I shows the complexity results obtained for transcoding for both cases of transrating and spatial resolution reduction by a factor of 2. In the case of spatial downsizing, the CIF input was transcoded into QCIF for the same set of QPs.

Comparing the computational complexity between FS and the proposed ME (**Proposed. vs FS**)we can see clearly the enormous evolution achieved by recent search methods. The exhaustive FS motion search algorithm presents from 100 times to 250 more SAD computations when compared with the proposed modified EPZS algorithm.

In the case of the proposed method (**Proposed vs EPZS**), the results show that a significant reduction in computational complexity is achieved. The total number of SAD calculations is always reduced to less than half (up to 59.6%) of the original EPZS. This is due to the fact that, on average the proposed MV predictors are much closer to the best MV than those originally computed by the EPZS algorithm, which results in much less checking points in the search process. Overall, these results show that the proposed method reaches the optimum MV much faster than the original EPZS algorithm.

B. Transcoding R-D performance

The R-D performance was also evaluated for the same set of transcoding parameters used in the previous subsection. Table II shows the R-D results obtained for transcoding with and without reduction of spatial resolution (i.e., transrating). No RD curves are shown because all results are very close to each other. Comparing the FS results with the proposed method (*Proposed vs FS*), one can see that, in terms of RD performance, generally there is only a slight loss in PSNR and a small decrease in bitrate, therefore the RD performance remains almost the same. Only in sequence *Stefan* there is a decrease in both PSNR and bitrate, resulting in a small decrease in performace, but like in the other sequences there is a huge gain in complexity, that compensates these losses.

In the case of transcoding with the proposed MV predictors, the results shows a negligible loss in R-D performance. In fact, both the bit rate and PSNR obtained in the case of **Proposed vs EPZS** are basically not changed when compared to the EPZS. Taking into account the complexity results of table I, the proposed method achieves the same practical R-D performance as the original EPZS at about half computational complexity.

Overall, the results show that in both cases of transcoding (transrating and downscaling) the proposed method is capable of achieving a significant reduction in computational complexity without compromising the R-D performance. Such relative gains were found to be consistent when using different sequences and coding modes, e.g., skip mode enabled. This is due to the additional MV prediction introduced in the EPZS algorithm by the proposed method, which proves to be more accurate than all others considered in original EPZS algorithm. The main reason is the good accuracy of the initial MV prediction, which is one of the key factors which most influences the complexity in predictive zonal search algorithms.

V. Conclusion

In this paper an efficient MV prediction method was proposed for zonal search motion estimation algorithms in video transcoders. The proposed method computes a motion vector field from the incoming video stream and computes optimised predictors to replace the EPZS original predictors. The simulation results show that the proposed method is able to reduce the computational complexity more than 50% with negligible loss in R-D performance. Transcoding systems and applications can greatly benefit from the method described in this paper to reduce computational resources and improve efficiency, especially in platforms operating with many simultaneous video transcoders.

TABLE I
TRANSCODING COMPLEXITY (NUMBER OF SAD COMPUTATIONS)

		Transrating only				Spatial Downscaling			
Seq.	QP	FS	FS\Proposed	EPZS	Proposed. vs EPZS(%)	FS	FS\Proposed	EPZS	Proposed vs EPZS(%)
Akiyo	30	843499182	123	14 880 410	-53.7	350052186	166	5 199 891	-59.6
	34	728091093	97	16 516 199	-54.4	283206006	130	5 294 546	-58.7
	38	587447831	77	16 626 803	-54	229138812	105	5 159 423	-57.5
	42	431238114	60	15 099 623	-52.2	171633246	81	4 903 461	-56.6
Bus	30	4388892183	230	39 488 265	-51.6	1241975568	244	10 077 003	-49.5
	34	3525661074	178	40 291 895	-50.6	967973391	177	10 804 879	-49.3
	38	2122851309	102	41 944 352	-50.3	728326043	127	11 311 301	-49.2
	42	1984982891	102	39 332 435	-50.4	536214016	101	10 658 872	-50
Stefan	30	4066659023	248	34 322 481	-52.2	1128154113	266	9 043 659	-50.8
	34	3235374145	182	36 693 650	-51.7	895098708	179	10 566 668	-51.2
	38	2443173653	126	39 948 787	-51.3	693459078	124	10 761 732	-50.7
	42	1816534804	100	37 641 509	-51.9	531516430	108	10 254 431	-50.9

TABLE II
TRANSCODING R-D PERFORMANCE

			Transrating only				Spatial Downscaling			
	QP	kbps dB	FS	Proposed vs FS	EPZS	Proposed vs EPZS	FS	Proposed. vs FS	EPZS	Proposed vs EPZS
Akiyo	30	BR	159.89	-1.055%	158.27	-0.03%	54.2	-0.24%	54.11	-0.07%
		PSNR	41.44	-0.05	41.37	+0.02	37.92	-0.01	37.92	-0.01
	34	BR	118.44	-1.32%	117.02	0.1%	40.79	-0.44%	40.67	-0.15%
		PSNR	37.77	-0.04	37.66	=	35.09	-0.01	35.08	=
	38	BR	95.71	-2.74%	94.91	-1.84%	31.76	-0.28%	31.75	-0.25%
		PSNR	34.62	-0.09	34.57	-0.04	32.3	-0.01	32.36	-0.01
	42	BR	81.58	-2.49%	80.33	-0.9%	26.67	-2.34%	26.29	-0.87%
		PSNR	32.08	-0.06	32.04	-0.02	29.88	-0.05	29.84	-0.01
Bus	30	BR	1042.99	-2.07%	1024.33	-0.2%	279.54	-1.92%	273.01	+0.45%
		PSNR	36.04	-0.04	35.92	+0.08	33.77	-0.02	33.74	+0.01
	34	BR	589.41	-2.83%	573.05	+0.03%	163.25	-1.88%	159.11	+0.7%
		PSNR	31.8	-0.03	31.76	+0.01	30.68	-0.03	30.65	=
	38	BR	334.51	-2.76%	324	+0.5%	90.95	-1.2%	89.44	0.48%
		PSNR	28.67	-0.04	28.63	=	27.73	-0.02	27.71	=
	42	BR	206.97	-2.91%	199.51	+0.8%	54.31	-2.22%	52.86	+0.51%
		PSNR	26.17	-0.08	26.11	-0.02	25.2	-0.08	25.14	-0.02
Stefan	30	BR	907.41	-0.33%	904.12	-0.03%	277.4	+0.54%	276.80	+0.76%
		PSNR	36.69	=	36.63	+0.03	33.77	-0.05	33.75	-0.03
	34	BR	515.13	-0.94%	511.5	-0.22%	161.12	+0.48%	160.47	+0.88%
		PSNR	32.33	-0.02	32.31	=	30.48	-0.05	30.43	=
	38	BR	310.22	-0.18%	308.47	+0.39%	95.05	+1.44%	94.55	+0.79%
		PSNR	29.01	-0.03	28.98	=	27.36	-0.04	27.33	-0.01
	42	BR	204.07	+0.53%	202.51	+1.29%	58.76	+0.6%	58.43	+0.71%
		PSNR	26.11	-0.05	26.07	-0.01	24.44	-0.04	24.41	-0.01

REFERENCES

[1] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 84–97, Jan. 2005.

[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, pp. 7–28, 1st Quarter 2004.

[3] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *Multimedia, IEEE Transactions on*, vol. 2, no. 2, pp. 101–110, Jun 2000.

[4] S. Moiron, S. Faria, A. Navarro, V. Silva, and P. Assunção, "Video transcoding from H.264/AVC to MPEG-2 with reduced computational complexity," *Signal Processing: Image Communication*, vol. 24, no. 8, pp. 637 – 650, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6V08-4WK43RM-1/2/dacbfdefd61f99eca337e47e9222756b

[5] X. Lu, A. Tourapis, P. Yin, and J. Boyce, "Fast mode decision and motion estimation for h.264 with a focus on MPEG-2/H.264 transcoding," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, vol. 2, May 2005, pp. 1246–1249.

[6] J. De Cock, S. Notebaert, K. Vermeirsch, P. Lambert, and R. Van de Walle, "Efficient spatial resolution reduction transcoding for H.264/AVC," in *Image Processing 2008. ICIP 2008. 15th IEEE International Conference on*, Oct. 2008, pp. 1208–1211.

[7] H. Shen, X. Sun, F. Wu, H. Li, and S. Li, "A fast downsizing video transcoder for H.264/AVC with rate-distortion optimal mode decision," in *Multimedia and Expo, 2006 IEEE International Conference on*, July 2006, pp. 2017–2020.

[8] A. H. Y. I. T. Koga, K. Iinuma and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. Nat. Telecommun. Conf. New Orleans*, Dec 1981, pp. G5.3.1–G5.3.5.

[9] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block matching motion estimation," in *Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on*, vol. 1, Sep 1997, pp. 292–296.

[10] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *VCIP*, 2002, pp. 1069–1079.

[11] http://iphome.hhi.de/suehring/tml/, *Suehring JM 15.1 H.264/AVC*.

[12] H. So, J. Kim, W.-K. Cho, and Y.-S. Kim, "Fast motion estimation using modified diamond search patterns," *IEE Electronics Letters*, vol. 41, no. 2, Jan. 2005.