FULLY COMPRESSED-DOMAIN TRANSCODER FOR PIP/PAP VIDEO COMPOSITION

Nuno Roma *Leonel Sousa*

INESC-ID / IST, TULisbon
Rua Alves Redol 9, 1000-029 Lisboa - PORTUGAL

ABSTRACT

An efficient architecture to perform Picture-In/And-Picture (PIP/PAP) composition in the compressed DCT-domain is proposed in this paper. One of the main innovative features of this architecture is a quite efficient least-squares motion re-estimation algorithm that is applied to improve the temporal prediction mechanism. Experimental results have shown that the proposed architecture may provide up to 1.3 dB PSNR gain over a traditional DCT-domain approach, without any re-estimation of the composited motion vectors, with a reduction of about 33% on the output bit-rate. Furthermore, the presented DCT-domain approach does not impose any limitation on the composition setup, allowing each foreground video sequence to be placed over any arbitrary location of the background sequence.

Index Terms— Compressed-domain transcoding, PIP/PAP composition, DCT-domain motion vector re-estimation

1. INTRODUCTION

With the recent proliferation of advanced video services and multimedia applications, video compression standards, such as MPEG-x or H.26x, have been developed. Once video signals are compressed, delivery systems and service providers frequently face the need for further manipulation of the compressed bit streams. Video transcoding emerged as a set of manipulation and adaptation techniques to convert a precoded video into another bit stream with different characteristics. Among the several approaches that have been presented, DCT-domain transcoders have proved to provide better performances, both in terms of quality and computational cost.

Picture-In-Picture (PIP) and Picture-And-Picture (PAP) composition schemes, where two or more sequences obtained from multiple video sources are combined into a single scene, either at the same scale, side by side (PAP), or by scaling all but one of the sequences (foreground scenes) and inserting them over the background scene (PIP), are often required by surveillance systems, multi-point videoconferencing and interactive network video. These manipulations can be implemented either at the client side or at the server side. Nevertheless, specialized network transcoding systems at the server side not only provide significant advantages in terms of bandwidth, but also allow the implementation of much simpler and cost effective receiver terminal devices.

Consequently, several different transcoding approaches, either in the pixel-domain or in the DCT-domain, have been proposed. One of the earliest proposals was presented by Chang, firstly in the pixel-domain [1] and then in the DCT-domain, presented in [2] and [3]. More recently, Li et al. [4, 5] presented two compositing schemes to be applied with the H.264 coding standard: the Rate-Distortion Re-Encoding (RDRE) architecture and the Partial Re-Encoding Transcoder (PRET). Although both schemes implement the compositing operation in the pixel-domain, the PRET architecture was shown to provide some significant computational advantages.

Independently of the processing domain, an important issue that still deserves further investigation is concerned with the re-estimation of the Motion Vectors (MVs), in order to improve the temporal prediction mechanism. Most transcoding architectures adopt a rather simplistic approach, that infers the new MVs from those of the original sequences, which are either directly used by the motion-compensated prediction mechanism [3] or are applied to a MV re-mapping scheme, as the one proposed by Chang [2]. However, due to the simplicity of their derivation, these remapped MVs often do not yield the minimum residual signal. They also usually imply the transposition of the DCT encoded Macroblocks (MBs) into the pixel-domain, in order to compute the matching measure.

Moreover, to simplify as much as possible the processing of the involved DCT encoded blocks and to avoid an extra transposition into the pixel-domain, many DCT-domain compositing architectures [1, 2] imply a perfect alignment of the foreground sequences with the adopted (8×8) block grid.

To overcome such limitations, a highly efficient fully compressed DCT-domain architecture is now proposed. This architecture combines two important processing algorithms, previously presented in [6] and in [7], that make it possible to efficiently implement all processing tasks without any conversion into the pixel-domain. Instead of only considering a couple of alternative positions for each re-estimated MV [2], the presented approach significantly improves the temporal prediction mechanism by adopting the least-squares MV re-estimation scheme presented in [6], that directly operates with the DCT coefficient blocks. Moreover, this scheme may also consider any dimension for the search area, thus significantly improving the output video quality and the bit-rate efficiency. This better performance is also obtained by implementing the DCT-domain down-scaling algorithm for any arbitrary integer scaling factor, previously proposed in [7].

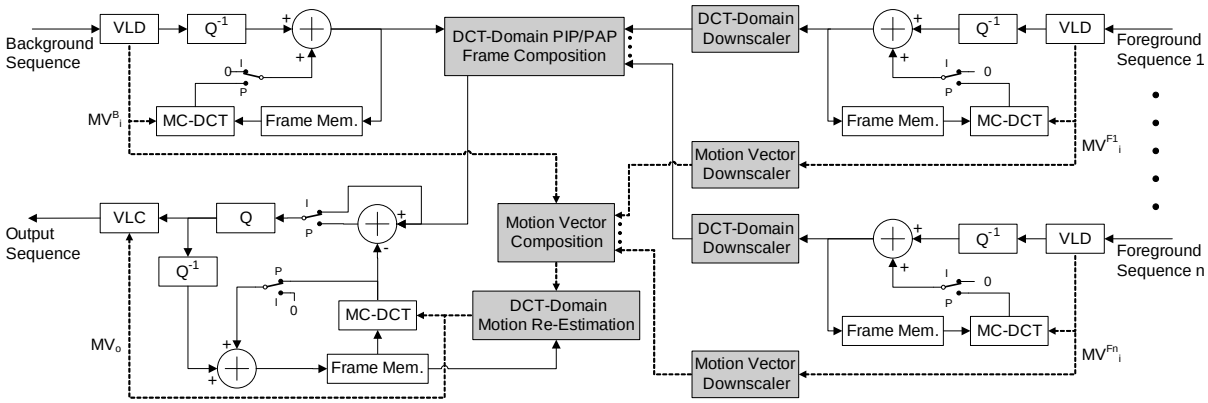

Fig. 1. DCT-domain PIP/PAP compositing transcoder architecture.

2. TRANSCODER ARCHITECTURE

The proposed DCT-domain PIP/PAP compositing transcoder is presented in fig. 1. The main modules that compose this architecture will be described in the following subsections. Moreover, since the PAP compositing layout can be regarded as a particular case of the PIP layout (with unitary scaling factor), from now on only the PIP layout will be considered.

2.1. Frame scaling

The frame scaling of the several considered foreground video sequences is composed by two processing steps: *i)* reduction of the spatial resolution and *ii)* composition and downscaling of the original MVs.

2.1.1. DCT-domain spatial frame scaling

Let $\mathbf{C}$ denote the $(N \times N)$ DCT matrix kernel ($N = 8$), so that $\mathbf{X} = \text{DCT}(\mathbf{x}) = \mathbf{C} \cdot \mathbf{x} \cdot \mathbf{C^t}$. The averaging and sub-sampling DCT-domain algorithm for any arbitrary integer downscaling factor ($\mathcal{S}$) proposed in [7] is adopted to compute each $(N \times N)$ DCT coefficients block $\hat{\mathbf{B}}$, corresponding to the set of $(\mathcal{S} \times \mathcal{S})$ original pixel blocks $\mathbf{b_{i,j}}$:

$$\hat{\mathbf{B}} = \frac{1}{\mathcal{S}^2} \cdot \mathbf{C} \cdot \left(\sum_{i=0}^{\mathcal{S}-1} \sum_{j=0}^{\mathcal{S}-1} \mathbf{p_{i,j}} \right) \cdot \mathbf{C^t} \qquad (1)$$

The matrix $\mathbf{p_{i,j}}$ is computed as:

$$[\mathbf{p_{i,j}}](l,c) = \begin{cases} \overline{\mathbf{p_{i,j}}} & , \text{ for } \begin{cases} l_{min}(i) \le l \le l_{max}(i) \\ c_{min}(j) \le c \le c_{max}(j) \end{cases} \\ 0 & , \textit{otherwise} \end{cases} \qquad (2)$$

with $l_{min}(i) = \left\lfloor \frac{i*N}{\mathcal{S}} \right\rfloor$, $l_{max}(i) = \left\lfloor \frac{i*N+(N-1)}{\mathcal{S}} \right\rfloor$, $c_{min}(j) = \left\lfloor \frac{j*N}{\mathcal{S}} \right\rfloor$ and $c_{max}(j) = \left\lfloor \frac{j*N+(N-1)}{\mathcal{S}} \right\rfloor$. The computation of the non-null elements ($\overline{\mathbf{p_{i,j}}}$) can be implemented as follows:

$$\overline{\mathbf{p_{i,j}}} = \underbrace{\overline{\mathbf{F_{\mathcal{S}}^{i}}} \cdot \mathbf{B_{i,j}} \cdot \overline{\mathbf{F_{\mathcal{S}}^{j}}}^{\mathbf{t}}}_{n_l(i) \times n_c(j) \text{ matrix}} \qquad (3)$$

with $n_l(i) = l_{max}(i) - l_{min}(i) + 1$ and $n_c(j) = c_{max}(j) - c_{min}(j) + 1$. In this equation, $\mathbf{B_{i,j}}$ is the $(N \times N)$ DCT coefficients block, directly obtained from the original bit stream. The $(n(x) \times N)$ $\overline{\mathbf{F_{\mathcal{S}}^{x}}}$ terms (with $0 \le x \le \mathcal{S}-1$) are constant matrices, that can be pre-computed and stored in memory [7].

2.1.2. Motion vector downscaling

To minimize the involved computational cost, the set of MVs decoded from the foreground video sequences are re-used. However, considering that PIP composition usually involves a reduction of the spatial resolution, the received MVs have to be properly adapted by a compositing and a scaling step.

One of the most used MV compositing methods is the Maximum QB-Area (MQBA) technique [8], which relies on each MB compositing area and the corresponding spatial activity to weight the influence of each MV. However, to avoid the inherent cost of computing the inverse DCT, the spatial activity is directly estimated from the DCT coefficients of each block, by counting the number of non-null AC coefficients (θ):

$$\tilde{v} = v(p^*), \text{ where } p^* = \underbrace{\arg\max}_{p \in P} \theta(p) \times A(p) \qquad (4)$$

where P denotes the set of involved compositing MBs from the original bit-stream.

The resulting composited MVs are then scaled by $\mathcal{S}$. To further improve the performance of the temporal prediction mechanism, an accurate half-pixel precision scaler is applied.

2.2. DCT-domain frame composition

The composition of one or more foreground sequences with the background video sequence at arbitrary positions may lead to mismatches of the corresponding block grids [2], as it is illustrated in fig. 2. In such a situation, each $(N \times N)$ DCT coefficients block of the involved foreground scenes ($\mathbf{Y_i}$) has to be re-segmented and translated with respect to the block structure of the background scene ($\mathbf{X}$). The output block $\mathbf{B}$ will then contain the contributions from each original scene:

$$\mathbf{B} = \mathbf{X} - \sum\nolimits_i \mathbf{X_{seg_i}} + \sum\nolimits_i \mathbf{Y_{seg_i}} \qquad (5)$$

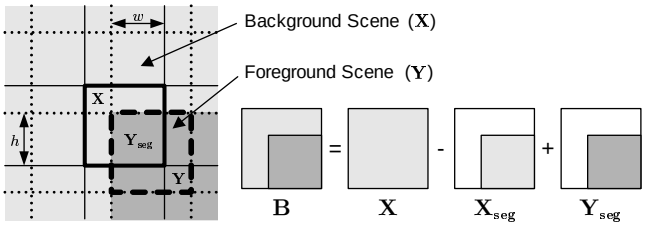

Fig. 2. DCT-domain PIP/PAP compositing.

The mathematical model to obtain the DCT coefficients of a given extracted and translated sub-block was already formulated in [2]. According to the example shown in fig. 2:

$$\mathbf{X_{seg}} = \mathbf{H_1}^x \cdot \mathbf{X} \cdot \mathbf{H_2}^x \quad ; \quad \mathbf{Y_{seg}} = \mathbf{H_1}^y \cdot \mathbf{Y} \cdot \mathbf{H_2}^y \quad (6)$$

where $\mathbf{H} = \mathrm{DCT}(\mathbf{h})$ are pre-computed and stored matrices, with $\mathbf{h_1}^x = \left[\begin{smallmatrix} 0 & 0 \\ 0 & I_h \end{smallmatrix}\right]$, $\mathbf{h_2}^x = \left[\begin{smallmatrix} 0 & 0 \\ 0 & I_w \end{smallmatrix}\right]$, $\mathbf{h_1}^y = \left[\begin{smallmatrix} 0 & 0 \\ I_h & 0 \end{smallmatrix}\right]$, $\mathbf{h_2}^y = \left[\begin{smallmatrix} 0 & I_w \\ 0 & 0 \end{smallmatrix}\right]$. I_h and I_w are $(h \times h)$ and $(w \times w)$ identity matrices, where h and w are the number of rows and columns to be extracted, respectively.

2.3. Motion vector re-estimation

Chang [2] distinguished three distinct areas of the composited scene: the *unaffected area*; the *directly affected area*, corresponding to all MBs where the compositing is performed; and the *indirectly affected area*, whose prediction data may still need to be recalculated, since its prediction blocks could have been modified through error propagation (drift). Hence, to minimize the involved computational cost, the MV re-estimation is usually preceded by a MV prediction stage.

2.3.1. Motion vector prediction

A first estimate of the MVs used by the temporal prediction mechanism of the output video sequence can be directly obtained from the set of MVs of the original background (v_B) and scaled foreground (v_F) video sequences. However, as it was previously referred, the borders of the composited MB often do not align with the original MB grid of the precoded frames. Depending on the actual position of the foreground sequences, the precoded MBs may be only partially or wholly used to compose the MBs of the output sequence.

The following procedure is adopted to obtain the initial MVs predictions ($\hat{v}$) corresponding both to the directly and indirectly affected areas, as well as a preliminary measure of the search area that should be considered in the subsequent MV re-estimation module (p_{mre}), considering $p_{max} = 4$:

- $\hat{v} = (0,0)$; $p_{mre} = p_{max} \rightarrow$ if the current MB and its prediction belong to different compositing sequences;
- $\hat{v} = v_B$; $p_{mre} = p_{max}/2 \rightarrow$ if at least 75% of the current MB area belongs to the background sequence;
- $\hat{v} = v_F$; $p_{mre} = p_{max}/2 \rightarrow$ if at least 75% of the current MB area belongs to the foreground sequence;
- $\hat{v} = (0,0)$; $p_{mre} = p_{max} \rightarrow$ *otherwise*.

2.3.2. Least squares DCT-domain motion re-estimation

The MVs that are obtained from the prediction module ($\hat{v}$) are used as coarse estimates of the desired MVs of the composited frame. These estimates, as well as the DCT coefficients blocks of the current and previously composited frames, are applied to the following iterative Least Squares Motion Estimation (LSME) algorithm, entirely performed in the DCT-domain [6], in order to obtain an accurate refinement of the desired MVs:

Step 0: Fetch the MBs corresponding to the current (R) and previous (S) frames of the composited video sequence, by performing a DCT-domain motion compensation operation using the initial prediction MV: $v^0 = \hat{v}$

Step 1: Compute the prediction error in the DCT domain:
$E = [R - S]_{4M \times 1}$, by considering v^i

Step 2: Compute the partial derivatives of S:
$$\mathbf{J_s} = \left[\begin{array}{c|c} \mathbf{D_1} \cdot \mathbf{S} \cdot \mathbf{D_2}^T & \mathbf{D_2} \cdot \mathbf{S} \cdot \mathbf{D_1}^T \end{array} \right]_{4M \times 2}$$

Step 3: Compute the displacement increment dv^i:
$$dv^i = \left[\begin{array}{c} dv_1^i \\ dv_2^i \end{array} \right]_{2 \times 1} = \left(\mathbf{J_s}^T \cdot \mathbf{J_s} \right)^{-1} \cdot \mathbf{J_s}^T \cdot \mathbf{E}$$

Step 4: Update the motion vector: v^i: $v^i = v^{i-1} + dv^i$

Step 5: Evaluate the stop condition:
If $\|v^i - v^{i-1}\| < \delta$ or $\|v^i - \hat{v}\| > p_{mre}$, stop the algorithm and set $v = v^i$; *otherwise*, re-compute $S|_{v=v^i}$ and return to **Step 1**.

All the considered matrices are processed in vectorized form. $\mathbf{D_1}$ and $\mathbf{D_2}$ are constant $(N \times N)$ matrices that are precomputed and stored in memory [6]. One advantage of implementing this algorithm in the DCT-domain is that it concentrates most of the pixels energy in the lower frequency coefficients of the encoded blocks. As a consequence, to reduce the involved computational cost, only the $(M \times M)$ lower frequency coefficients of each block may be considered. Moreover, to avoid the interference of any eventual difference of the luminance level of the two considered frames, the DC coefficients of the blocks may also not be considered.

3. EXPERIMENTAL RESULTS

The proposed DCT-domain transcoder architecture was applied to implement a PIP compositing layout, by considering a single foreground video sequence that is scaled by $\mathcal{S} = 3$ and positioned over a CIF format background sequence at coordinates $(l, c) = (11; 223)$ (see fig. 3(a)). This particular setup was chosen not only because it corresponds to a quite common layout used by many television applications, but also because it leads to a maximization of the previously described *directly affected area*. The MV re-estimation algorithm was implemented by considering $M = 4$ and a maximum of 3 iterations to converge to the final MV, unless an additional

(a) PIP compositing layout.

Video Sequences	Q	PSNR [dB]			Bit-rate [kbps]		
		PDT	TDT 1	TDT 2	PDT	TDT 1	TDT 2
Mobile+Carphone	4	33.62	34.57	35.84	203.34	308.18	221.63
	8	30.37	29.66	30.83	105.35	159.46	109.31
	12	27.87	26.89	28.06	66.39	101.56	67.54
Coastguard+Silent	4	34.56	35.19	36.41	85.36	144.62	98.10
	8	31.56	31.22	32.18	40.83	65.19	44.03
	12	29.77	29.13	29.98	26.31	39.25	27.00

(b) Average PSNR and bit-rate results.

Fig. 3. Experimental results obtained from the considered PIP compositing setups.

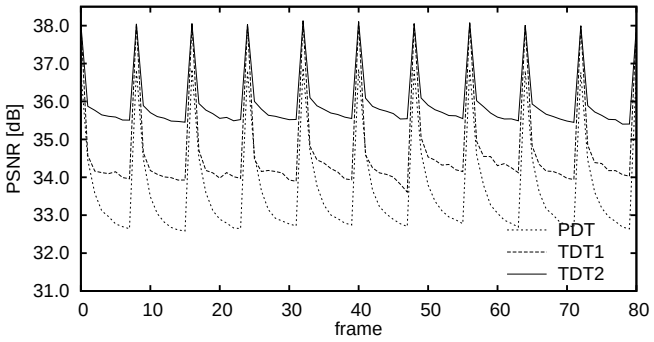

Fig. 4. Variation of the PSNR quality measure of the PIP composited *Mobile+Carphone* video sequence ($Q = 4$).

constraint, corresponding to the adopted stopping condition, was met: $\|v^i - v^{i-1}\| < \delta$, where $\delta = 0.1$.

The average PSNR and bit-rate values presented in the table of fig. 3(b) were obtained by inserting the *Carphone* and the *Silent* video sequences over the *Mobile* and *Coastguard* sequences, respectively. Several quantization steps (Q) were considered. The proposed transform-domain transcoder (TDT2) was compared with a pixel-domain transcoder with a full-search MV re-estimation (PDT) (equivalent to [1, 4, 5]) and a transform-domain transcoder without MV re-estimation (TDT1) (equivalent to [3]). The variation of the PSNR measure along the time for the *Mobile+Carphone* composited video sequence is also presented in fig. 4.

The obtained results evidence that the proposed DCT-domain transcoder can provide significant PSNR gains over the other two approaches, with a greater supremacy over the PDT transcoder. Such quality improvement is mainly due to the absence of arithmetic and round-off errors introduced by DCT and IDCT computational blocks and may lead to average gains as high as 2.2 dB and 1.3 dB, respectively. Moreover, the enhanced temporal prediction mechanism provided by the MV re-estimation module leads to a significant reduction of the average bit-rate (about 33% less), when the proposed architecture is compared with a similar approach without re-estimation of the MVs (TDT1).

4. CONCLUSION

A fully compressed DCT-domain transcoder to perform PIP and PAP composition was proposed. This architecture not only allows the insertion of the foreground sequences at any arbitrary location, independently of the usual ($N \times N$) block grid, but it also includes a quite efficient DCT-domain least-squares motion re-estimator that significantly improves the temporal prediction mechanism. The experimental results obtained with this architecture have shown that it may provide up to 1.3 dB PSNR gain over a traditional DCT-domain approach without any re-estimation of the composited MVs, with a reduction of about 33% on the output bit-rate.

References

[1] S.-F. Chang and D.G. Messerschmitt. Compositing motion-compensated video within the network. In *4th Int. Workshop on Multimedia Communications (MULTIMEDIA'92)*, pages 40–56. IEEE, April 1992.

[2] S.-F. Chang and D. G. Messerschmitt. Manipulation and compositing of MC-DCT compressed video. *Journal on Selected Areas in Communic.*, 13(1):1–11, January 1995.

[3] Y. Noguchi, D.G. Messerschmitt, and S.-F. Chang. MPEG video compositing in the compressed domain. In *Int. Symposium on Circuits and Systems (ISCAS'96)*, volume 2, pages 596–599. IEEE, May 1996.

[4] C.-H. Li, H. Lin, C.-N. Wang, and T. Chiang. A fast H.264-based picture-in-picture (PIP) transcoder. In *Int. Conf. on Multimedia and Expo (ICME'04)*, volume 3, pages 1691–1694. IEEE, June 2004.

[5] C.-H. Li, C.-N. Wang, and T. Chiang. A low complexity picture-in-picture transcoder for video-on-demand. In *Int. Conf. on Wireless Networks, Comm. and Mobile Computing*, volume 2, pages 1382–1387, June 2005.

[6] N. Roma and L. Sousa. Least squares motion estimation algorithm in the compressed DCT domain for H.26x/MPEG-x video sequences. In *Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS'05)*, pages 576–581. IEEE, September 2005.

[7] N. Roma and L. Sousa. Efficient hybrid DCT-domain algorithm for any arbitrary integer re-size video downscaling. *EURASIP Journal on Advances in Signal Processing*, (57291):1–16, 2007.

[8] Y.-P. Tan, Y. Liang, and H. Sun. On the methods and performances of rational downsizing video transcoding. *Signal Processing: Image Communication*, 19:47–65, 2004.