ARBITRARILY-SHAPED VIDEO CODING: SMART PADDING VERSUS MPEG-4 LPE/ZERO PADDING

Andy C. Yu*, Guobin Shen*, Bing Zeng*, and Oscar C. Au*

# Dept of Computer Science, The University of Warwick, United Kingdom. Email: andycyu@dcs.warwick.ac.uk

* Internet Media Group, Microsoft Research Asia, Beijing, China Email: jshen@microsoft.com

& Dept of EEE, The Hong Kong University of Science and Technology, Hong Kong E-mails: {eezeng, eeau}@ust.hk

ABSTRACT
An effective padding scheme, called the smart padding (SmartPad), has been developed recently for the DCT coding of arbitrarily-shaped image/video objects [1]; whereas its superior performance over the MPEG-4’s LPE padding has been confirmed solidly. In the present paper, we propose to extend the use of SmartPad to all INTER frames (of arbitrary shapes), i.e., to use SmartPad to replace the zero padding scheme (as recommended in MPEG-4). Our simulation results show that a very substantial performance gain (3-7 dB) has been achieved, as compared to the MPEG-4’s LPE/zero padding scheme.

1. INTRODUCTION
MPEG-4 video extends conventional block-based video codecs (e.g., H.263 and MPEG-2) to object-based coding of segmented video objects (VOs) that can have arbitrary shapes. One of the prerequisites is the ability to encode arbitrarily shaped VOs, i.e., to have efficient and effective algorithms that can solve the problem of content-based or segment-based coding. The start-of-the-art shape-adaptive DCT (SA-DCT) is the one developed by T. Sikora and his colleagues [2]. One important feature of SA-DCT is that it results in exactly the same number of transform coefficients as that of pixels within the original pixel block (of an arbitrary shape). It shows a remarkable performance gain for coding arbitrarily shaped objects, especially when the bit rate is high. However, SA-DCT has several drawbacks such as the non-orthonormality and the mean weighting defect [3].

Another effective way to encode non-square boundary blocks is to pad them back into fully defined square blocks and then process the padded blocks using the ordinary DCT. Such padding-based methods do not suffer from those drawbacks discussed above for SA-DCT and can directly utilize various existing fast DCT algorithms and commercially available DCT chipset. For example, the LPE padding (for intra frames) and zero padding (for inter frames) recommended in MPEG-4 and the extension interpolation (EI) padding scheme [4] have been developed in order to code the non-squared block. However, the LPE/zero padding leads to much worse coding performance; whereas the EI padding scheme results in much higher computational complexity.

Very recently, an effective padding scheme, called the smart padding (SmartPad), was proposed in [5] that requires less computational time than both SA-DCT and other padding schemes; and does not have any aforementioned shortcomings. In addition, this scheme also preserves the shape information, which is quite helpful in subsequent entropy coding stage. Several literatures have reported the various functionalities of SmartPad [7-8]. However, SmartPad has so far been used only for coding intra frames. In this paper, we extend the application of SmartPad to the coding of inter frames.

The rest of this paper is organized as follows: In Section 2, we review briefly the principle of SmartPad. The proposed way to apply SmartPad to achieve arbitrarily shaped DCT coding in both I-frames and P-frames is presented in Section 3. Extensive simulations have been performed to compare this new coding scheme with MPEG-4; and some of these simulation results are reported in Section 4. Finally, some conclusions are drawn in Section 5.

2. THE SMART PADDING SCHEME
The basic idea of SmartPad is to pad an original vector (of length \(N\) ) into a new vector (of length \(M > N\) ) via concatenating the padding vector (of length \(M - N\) ) to the rear of the original one, such that the DCT coefficients of the padded vector are guaranteed to have at least \(M - N\) trailing zeros.

As a simple example, let us consider an input vector of size \(N = 5\): \([2 \ 4 \ 6 \ 8 \ 10]\). We want to pad three extra values in order to perform the 8-point DCT. If we choose these three values as follows: \([7.7251 \ -1.8216 \ -11.8763]\); and then apply the 8-point DCT on the concatenated vector, we obtain:
Clearly, we have achieved the goal, i.e., all three coefficients in rear (equivalent to high-frequency components) are zero. In fact, it was proven theoretically in [1] that, for any input vector, we could always find the unique padding vector to meet the requirement set above. Furthermore, this unique solution is obtained by multiplying the original vector with an \((M-N)\times N\) matrix, whereas this matrix depends only on the DCT coefficient matrix (thus being independent of the original vector elements):

\[
\begin{bmatrix}
\hat{x}_1 \\
\vdots \\
\hat{x}_{M-N}
\end{bmatrix} = [P]_{(M-N)\times N} \begin{bmatrix}
x_1 \\
\vdots \\
x_{N}
\end{bmatrix} \tag{1}
\]

On the other hand, any shrewd person will find that the resulting DCT coefficients in the above example are not desirable for compression because all non-zero AC coefficients have relatively large magnitudes. This problem could be avoided if we allow the elements of the padding vector to be scattered into the original vector, i.e., we interleave the padding vector with the original vector. For the same example, if we choose the three padding values as \([2.5570 \ 5.4894 \ 6.3555]\) and form the new vector as \([2.5670 \ 4.5489 \ 4.6 \ 3.5555 \ 8 \ 10]\), we will get:

\[
[15.7020 \ -6.8860 \ 0.2763 \ -1.4990 \ 0.9076 \ 0 \ 0 \ 0]
\]

after the 8-point DCT. We see that there are still three zeros at the high frequency end, but the transformed vector is much more suitable for compression.

The question of how to find the optimal interleaving orders has been solved in [1] for the length of an original vector being 2 to 7. Notice that those orders are independent of any original vector and thus are universally usable. With each optimal interleaving order, the corresponding matrix \([P]_{(M-N)\times N}\) has also been given in [1] for \(M = 8\) and \(N = 2,\ldots,7\). Again, these matrices are independent of any original vector and thus are universally usable.

Other useful features of SmartPad include: (1) the number of non-zero coefficients after doing DCT on the padded vector is no larger than the number of pixels included in each original vector, thus yielding no data expansion (such expansion is usually un-avoidable within the LPE/zero padding of MPEG-4); (2) all pixels included in each original vector have never been changed (whereas the opposite happens in the EI padding scheme [4], and therefore a de-padding is needed in the EI scheme).

3. ARBITRARILY-SHAPED VIDEO CODING WITH SMARTPAD

The MPEG-4 standard has its own sophisticated padding scheme: a low-pass extrapolation (LPE) padding scheme is recommended for intra frames and a zero padding scheme for motion-compensated inter frames.

For boundary blocks in each inter frame, MPEG-4 proposes some special treatments for motion estimation and compensation, such as macroblock-based repetitive padding (both horizontal and vertical), extended padding, and modified block (polygon) matching.

All these special treatments will remain in our new approach. What is new in our approach is to replace the LPE/zero padding with SmartPad. Thus, SmartPad is applied to both intra and inter frames without any change. It can be considered as an advantage that a consistent padding technique is used throughout the whole coding process.

In our earlier works, we applied SmartPad to intra frames only. To realize how SmartPad also works effectively in inter frames, we select one \(8\times8\) block as follows:

\[
[\text{news2}]_{\text{residue}} = \begin{bmatrix}
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast \\
\ast & \ast & \ast & \ast & \ast & \ast & \ast & \ast
\end{bmatrix}
\tag{2}
\]

In fact, this block is taken from the residue block of the New2 sequence (Class B).

In both the intra and inter cases, each \(8\times8\) block along the boundary of each video object (VO) consists both defined elements (i.e., pixels belonging to the VO) and un-defined elements (i.e., pixels not belonging to the VO). Different from the intra-frame case, an \(8\times8\) boundary block in inter-frame is taken from the corresponding residue frame that is the macroblock-based difference between the original VO and its reconstructed VO recovered from motion compensation. Normally, all defined elements in each residue block are quite small in magnitude.

In the MPEG-4 scheme, the zero padding technique will substitute zeros to all un-defined elements of the residue block, followed by DCT, quantization, and entropy coding. Blocks (3) and (4) are the reconstructed results after applying the zero-padding technique and the SmartPad technique:

```
(3)  
(4)
```

Compared with the original pixel block (only on the defined elements), the MSE’s of the zero padding scheme and
our SmartPad are 72 and 17, respectively. This shows an improvement of more than four times, or equivalently a gain of more than 6 dB. In other words, SmartPad can still handle very well even though the defined elements are sparse and take small values. More extensive simulations in Section 4 will support this fact.

The contribution of gain gives the credit to the well structured post-DCT block derived from using the proposed padding technique. As we know, the 2-D padding process can be separated to row-wise and column-wise 1-D padding, respectively. Therefore, SmartPad interfaces the significant elements to the padding vector row-by-row into a pre-defined order of original vector at first. For instance, after the 1-D padding, the last row of Block (2) will become

\[
[-5 \quad -1.7626 \quad 1 \quad 2.265 \quad 4 \quad 5.4893 \quad 4 \quad 1] \quad (5)
\]

The pre-defined padded elements evaluated from our algorithm are inserted in the positions of 2, 4, and 6 (underlined). After carrying on 1-D DCT operation, it becomes

\[
[3.89 \quad -6.75 \quad -5.56 \quad 0.75 \quad -2.28 \quad 0 \quad 0 \quad 0] \quad (6)
\]

Note that the energy distribution of (6) tends to the low frequency (left); while zeros (exactly equal to the number of un-defined elements) occupy the high frequency (right) end. Thus, one can expect that the energy distribution of Block (2) will focus at the upper-left corner (low frequency component) of its corresponding frequency block after applying the second directional padding and 8-point DCT.

In contrary, the padded zeroes by the zero-padding technique to all un-defined elements will cause the energy distribution of Block (2) scattered over the whole frequency domain due to energy conservation theory. This is known as the ill structure of DCT block, which is too fragile to be well quantized.

4. SIMULATIONS RESULTS

The proposed SmartPad technique is simulated on three selected sequences: Akiyo (Class A sequence), News1 (Class B sequence), and News2 (Class B sequence), as shown in Fig. 1. We simulated 100 frames from each of sequences and compare the PSNR performance between the SmartPad technique and the LPE/zero padding technique (recommended in the MPEG-4 standard).

The settings are as following: an I-frame is followed by nine P-frames. The motion estimation is performed with the full search algorithm and the searching window size is chosen to be 16×16. A standard MPEG-4 quantization table is applied, in which the QP factor is set to be one in I-frames and two in P-frames, respectively. Finally, the run-length Huffman coding is employed [6].

The following three graphs, as shown in Fig 1, represent, respectively, the results of three testing sequences. Note that only the PSNR of boundary blocks is shown in each graph. This is reasonable because the padding techniques are applied only to boundary blocks in each frame.

![Fig 1. Video object and corresponding shape of (a) Akiyo (Class A); (b) News1 (Class B); and (c) News2 (Class B).](image1)

![Fig 2. Boundary PSNR of Akiyo: LPE/zero pad vs. SmartPad.](image2)

![Fig 3. Boundary PSNR of News1: LPE/zero pad vs. SmartPad.](image3)
It is obvious to see that the proposed padding technique provides much better performances than the MPEG-4’s LPE/zero padding (about 3-7 dB better) in both I-frames (frames 1, 11, and so on) and P-frames (frames 2 to 10, 12 to 20, and so on). In particular, one should notice that SmartPad achieves quite similar performance gains in all P-frames (against the MPEG-4’s zero padding) as in I-frames (against the MPEG-4’s LPE padding).

More simulation results as shown in Fig. 5 indicate that the relationship of average PSNR performance (same Qp factor was set in all I-frames and P-frames) against the Qp factor of quantization table. Note that the gap between the average performances of two padding schemes becomes narrower as the bit rate goes lower. However, the PSNR difference is still maintained at least 2dB.

These simulation results indicate another fact: it seems that a larger performance gain is achieved in the Class B sequences (News1 and News2). The reasons are as following: in the News1 sequence, we found that many boundary blocks in this sequence are composed by “sparse matrix”, i.e. these blocks contain very few defined elements, as shown in the previous example in Section 3; whereas in the News2 sequence, there are a lot of boundary blocks. These two situations are advantageous while handling with SmartPad.

5. CONCLUSION

The superior performance of SmartPad over the LPE/zero padding technique recommended in the MPEG-4 standard has been presented for both intra frames and inter frames through extensive simulation results. This is mainly because that our proposed padding technique constructs well structured DCT blocks that are more suitable to compression and also more robust to quantizations. Moreover, SmartPad seems to be very good too in terms of computational complexity versus rate distortion performance [7], [8]. In particular, it can be implemented by various kinds of computationally fast DCT algorithms or commercialized DCT chipsets. We strongly believe that SmartPad is the best padding technique in DCT-based arbitrarily-shaped video coding.

REFERENCES