ON THE DOWN-SCALING OF COMPRESSED PICTURES§

A. N. Skodras[1,2] and C. A. Christopoulos[3]

[1]Electronics Laboratory, University of Patras, GR-26110 Patras, Greece
[2]Computer Technology Institute, PO Box 1122, GR-26110 Patras, Greece
Fax: +30 61 997456, Email: skodras@cti.gr

[3]Ericsson Telecom AB, Compression Lab, TN/ETX/PN/XML, S-126 25 Stockholm, Sweden
Fax: +46 8 7190460, Email: ch.christopoulos@clab.ericsson.se

ABSTRACT

An efficient algorithm for the down-scaling of compressed pictures is proposed in this communication. The algorithm operates directly on the DCT (discrete cosine transform) coefficients, thus avoiding the need for decompressing, down-sampling in the spatial domain and re-compressing the images. As a result, the quality of the reconstructed images is higher and the computational complexity is lower than similar algorithms appeared in the open literature. The algorithm can be used in various applications, such as image and video browsing, video compositing and transcoding, and HDTV to SDTV conversion.

Keywords: DCT, transform coding, down-scaling, transcoding, JPEG, MJPEG, MPEG, H.26x

1. INTRODUCTION

Nowadays, the multimedia content of many business and consumer applications is disseminated in compressed form. A very frequent process that is required to be performed on the compressed data, is that of down-scaling or down-sampling, as it happens in the cases of image and video browsing, video compositing, transcoding, HDTV to SDTV conversion, etc. Specifically, in *image and video browsing* applications, it may be sufficient to deliver a lower resolution image or video to the user. Based on user's input, the media server could then provide the higher resolution image or video sequence [1,2].

Compositing several MPEG video sources into a single displayed stream is important for MPEG video applications as for example advanced multimedia terminals, interactive network video and multi-point video conferencing. Compressed domain based down-scaling can be used to implement an efficient picture-in-picture system for MPEG compressed video resulting in significant savings [1, 3].

Efficient *transcoding* could cope with different quality of services in the case of multi-point communications over POTS, ISDN, and ADSL lines [4]. A *HDTV down conversion* decoder can decode the Grand Alliance HDTV bitstreams and display them on SDTV or NTSC monitors [5, 6].

Down-scaling directly in the compressed domain reduces computational complexity by processing less data and avoiding the conversion process back and forth between the compressed and the uncompressed data formats. In compression standards (MPEG, H.26x), compression is computationally 3 to 4 times more expensive than decompression. In the present communication, an efficient down-scaling technique is proposed, in which the transition to the spatial domain is avoided.

The paper is organised in the following way: In section 2 the down-scaling approaches are presented, while the proposed algorithm is described in section 3. The computational complexity of the proposed technique is given in section 4 and the experimental results and the conclusions drawn, are demonstrated in section 5.

2. DOWN-SCALING APPROACHES

The down-scaling of a compressed picture can be performed either indirectly, by performing the transition to the spatial domain, or directly, by remaining in the transform domain and properly manipulating the coefficients. The different alternatives of these two approaches are described below.

2.1 Down-Scaling In The Spatial Domain

The down-scaling of a still picture in the spatial domain consists of two steps. First the image is filtered by an anti-aliasing low pass filter (LPF) and then it is sub-sampled by the desired factor in each dimension. For a DCT-compressed image, the above method implies that the compressed and quantised image has to be recovered first into the spatial domain and then undergo the process of filtering and down-sampling as it is illustrated in Fig 1a.

§ This work was supported by a Joint Research Grant of the British Council and General Secretariat for Research and Technology (Greece).

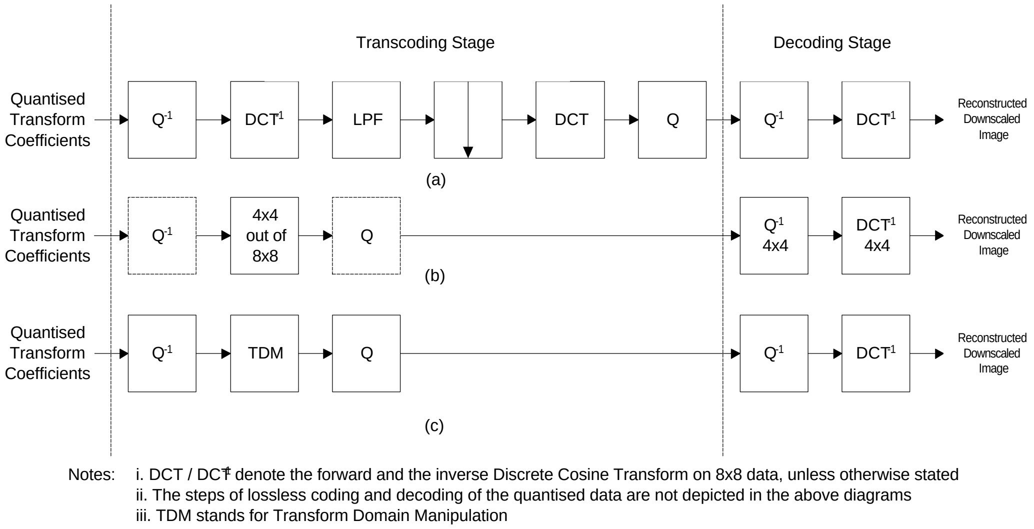

Figure 1. Block diagrams demonstrating the down-scaling approaches of compressed pictures

2.2 Down-Scaling In The Transform Domain

A direct approach would be that of working in the compressed domain, where both operations of filtering and down-sampling are combined in the DCT domain. This could be done by cutting-off the DCT coefficients of high frequencies and using the IDCT with a smaller number of coefficients to reconstruct the reduced resolution image. For example, one could use the lower 4x4 coefficients out of the 8x8 and perform the IDCT on these coefficients in order to reduce the resolution by a factor of 2 in each dimension (Fig. 1b). This approach, which is referred to as *frequency masking* approach, does not result in significant compression gain and most importantly, it requires encoders and decoders to be able to handle 4x4 DCT's and IDCT's. It also requires run-length coding schemes to be optimised for the 4x4 case. This method results in significant blocking artefacts, due to the poor approximations introduced by simply discarding higher order coefficients [5].

This direct approach would have been more useful if we had 16x16 DCT blocks and were keeping the lower 8x8 DCT coefficients. However, most image and video compression standards, like JPEG, H.26x, and MPEG, segment the images into rectangular blocks of size 8x8 pixels and apply the DCT on these blocks. Therefore, only 8x8 DCT's are available. If one could compute the 8x8 out of the 16x16 DCT coefficients by using only 8x8 transformations, then this method would be faster and it would perform better than the one that uses the 4x4 out of the 8x8.

In the present communication, an efficient algorithm is proposed for the computation of an NxN-point DCT given the N/2xN/2 DCT coefficients of four adjacent blocks, where all operations are performed in the compressed domain (transform domain manipulation, TDM).

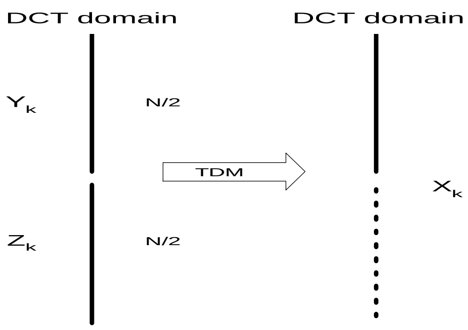

Figure 2.Schematic representation of the 1-D problem definition

3. THE PROPOSED ALGORITHM

The 1-D down-scaling analysis is presented below, in order to simplify the notation and discussion. Since the DCT is separable, all results extend to the 2-D case by simply applying the properties in each of the two dimensions consecutively.

Let us assume that the DCT coefficients Y_k and Z_k, $(k = 0,1,\ldots,(N/2)-1)$, of two consecutive data sequences y_n and z_n, $(n = 0,1,\ldots,(N/2)-1)$, are given, where $N = 2^m$ (Fig. 2). The problem to be addressed is the efficient computation of X_k, $(k = 0,1,\ldots,N-1)$ directly in the DCT domain, given Y_k and Z_k, where X_k are the DCT coefficients of x_n, $(n = 0,1,\ldots,N-1)$, the sequence generated by the concatenation of y_n and z_n.The normalised forward DCT (DCT-II) and inverse DCT (IDCT) of the length-N sequence x_n are given by the following equations [9]:

$$X_k = \sqrt{\frac{2}{N}}\varepsilon_k \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)k\pi}{2N}, \quad k = 0,1,\ldots,N-1$$

and

$$x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \varepsilon_k X_k \cos\frac{(2n+1)k\pi}{2N}, \; n = 0,1,\ldots,N-1$$

where $\varepsilon_k = 1/\sqrt{2}$ *for k = 0* and $\varepsilon_k = 1$ *for* $k \neq 0$. It is pointed out that $\varepsilon_{2k} = \varepsilon_k$ and $\varepsilon_{2k+1} = 1$. The normalised DCT and IDCT for the length-(N/2) sequences y_n and z_n are given by similar expressions, but with N substituted by N/2. The computation is performed separately for the even- and the odd-indexed coefficients.

i. Even-indexed coefficients

$$X_{2k} = \sqrt{\frac{2}{N}}\varepsilon_{2k} \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)2k\pi}{2N} =$$

$$= \sqrt{\frac{1}{2}}\left\{ \sqrt{\frac{2}{N/2}}\varepsilon_k \sum_{n=0}^{\frac{N}{2}-1} y_n \cos\frac{(2n+1)k\pi}{2(N/2)} + \sqrt{\frac{2}{N/2}}\varepsilon_k \sum_{n=0}^{\frac{N}{2}-1} z_n \cos\frac{(2n+1)k\pi}{2(N/2)} \right\}$$

$$= \sqrt{\frac{1}{2}}\left[Y_k + (-1)^k Z_k \right] = \sqrt{\frac{1}{2}}\left[Y_k + Z'_k \right] \quad k = 0,1,\ldots,(N/2)-1.$$

where Z'_k is the DCT of $z'_n = x_{N-1-n}$, $n = 0,1,\ldots,(N/2)-1$.

ii. Odd-indexed coefficients

$$X_{2k+1} + X_{2k-1} =$$

$$= \sqrt{\frac{2}{N}}\left\{ \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)(2k+1)\pi}{2N} + \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)(2k-1)\pi}{2N} \right\}$$

$$= \frac{1}{\varepsilon_k}\sqrt{\frac{1}{2}}\left\{ \sqrt{\frac{2}{N/2}}\varepsilon_k \sum_{n=0}^{\frac{N}{2}-1} (y_n - z'_n) 2\cos\frac{(2n+1)\pi}{2N} \cos\frac{(2n+1)k\pi}{2(N/2)} \right\}$$

or $$X_{2k+1} = \frac{1}{\varepsilon_k}\sqrt{\frac{1}{2}}\left\{ \sqrt{\frac{2}{N/2}}\varepsilon_k \sum_{n=0}^{\frac{N}{2}-1} r_n \cos\frac{(2n+1)k\pi}{2(N/2)} \right\} - X_{2k-1},$$

where $k = 0,1,\ldots,(N/2)-1$ and

$$r_n = (y_n - z'_n) 2\cos\frac{(2n+1)\pi}{2N}$$

$$= \left\{ \sqrt{\frac{2}{N/2}} \sum_{l=0}^{\frac{N}{2}-1} \varepsilon_l (Y_l - Z'_l) \cos\frac{(2n+1)l\pi}{2(N/2)} \right\} 2\cos\frac{(2n+1)\pi}{2N}$$

r_n is a length-(N/2) DCT of the length-(N/2) IDCT of (Y_l-Z_l') multiplied by *2cos(2n+1)ð/2N*. The calculation of the odd-indexed coefficients could be simplified if the processes of DCT^{-1}, DCT and the multiplications were substituted by a matrix multiplication as shown in Fig. 3. In the special case of N=16, i.e. the concatenation of two 8-point coefficient sequences, we have:

G=Y_l-Z_l' is a column vector of length 8 each element of which equals to the difference of the corresponding input DCT coefficients.

g=C^{-1}·G is a column vector of length 8 corresponding to the IDCT of G, where $C^{-1}= C^T$.

r=E·g is a column vector of length 8 each element of which is the product of g by *2cos(2n+1)ð/2N*, where n=0,1,…,7 and N=16. E is a diagonal matrix and is given by E = 2 diag{cos(ð/32),cos(3ð/32),cos(5ð/32), cos(7ð/32),cos(9ð/32),cos(11ð/32),cos(13ð/32), cos(15ð/32)}.

R=C·r is a column vector of length 8 corresponding to the DCT of r.

R=P·R is a column vector of length 8 each element of which is the product of R by $\sqrt{1/2}$, except for the first element that is multiplied by ½, i.e. matrix P equals to: P=diag{½, $\sqrt{1/2}$, $\sqrt{1/2}$ $\sqrt{1/2}$ $\sqrt{1/2}$ $\sqrt{1/2}$ $\sqrt{1/2}$ $\sqrt{1/2}$ }.

R is expressed as R=P·(C·(E·(C^{-1}·G))) = P·C·E· C^{-1}·G or R=T·G, where T=P·C·E·C^{-1}. Note that matrix C·E·C^{-1} is symmetric. The multiplication of these 8x8 matrices results to:

$$T = \begin{bmatrix} 0.6376 & 0.2986 & -0.0585 & 0.0241 & -0.0125 & 0.0071 & -0.0039 & 0.0018 \\ 0.4223 & 0.8433 & 0.3227 & -0.0710 & 0.0311 & -0.0164 & 0.0088 & -0.0039 \\ -0.0827 & 0.3227 & 0.8893 & 0.3057 & -0.0624 & 0.0259 & -0.0125 & 0.0053 \\ 0.0341 & -0.0710 & 0.3057 & 0.8978 & 0.3004 & -0.0585 & 0.0223 & -0.0086 \\ -0.0177 & 0.0311 & -0.0624 & 0.3004 & 0.9018 & 0.2969 & -0.0546 & 0.0170 \\ 0.0100 & -0.0164 & 0.0259 & -0.0585 & 0.2969 & 0.9057 & 0.2916 & -0.0460 \\ -0.0056 & 0.0088 & -0.0125 & 0.0223 & -0.0546 & 0.2916 & 0.9143 & 0.2745 \\ 0.0025 & -0.0039 & 0.0053 & -0.0086 & 0.0170 & -0.0460 & 0.2745 & 0.9603 \end{bmatrix}$$

The flow graph of the proposed algorithm for the case of the concatenation of two 8-point adjacent coefficient sequences (i.e. N=16), is depicted in Fig. 4. Down-scaling by a factor of 2 implies that only coefficients 0, 2, 4, 6, 1, 3, 5, 7 have to be calculated.

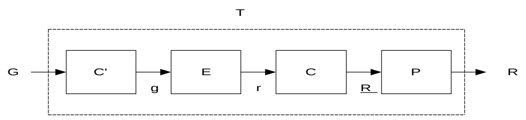

Figure 3. Calculation of the T matrix

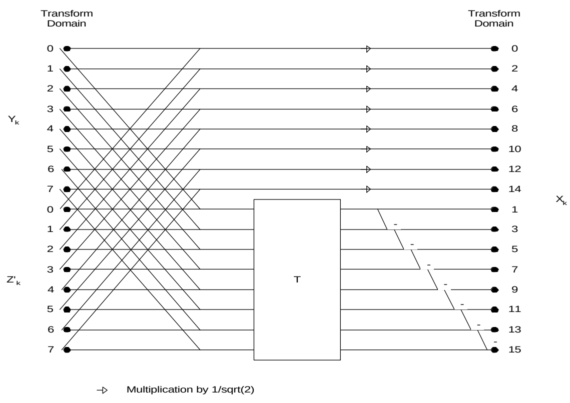

Figure 4. Flow graph of the proposed approach for N=16

4. COMPUTATIONAL COMPLEXITY

The computational complexity of the proposed algorithm is derived separately for the 1-D and the 2-D cases.

4.1 Complexity for the 1-D case

The computational complexity for computing N/2 out of N points, i.e. down-scaling by a factor of 2, is equal to $O_M=N(N+2)/8$ and $O_A=(N(N+6)-1)/8$. For computating the 8 out of the 16 coefficients, i.e. N=16, these expressions give *36 multiplications* and *43 additions*. More specifically:

- the computation of the even indexed coefficients requires 4 multiplications by $\sqrt{1/2}$ and 4 additions,
- the computation of $G=Y_l-Z_l'$ requires 8 subtractions,
- the computation of R requires 32 multiplications and 28 additions (only the upper 4x8 elements of the T matrix are used, since only coefficients X_i, i=1,3,5,7 have to be calculated), and
- 3 post additions are needed for calculating the required odd-indexed coefficients.

Special cases

a. When only the first 4 out of the 8 coefficients Y_k, Z_k' are non-zero (i.e. $Y_k = Z_k'=0$ for k=4,5,6,7), then the above given complexity reduces to *20 multiplications* and *23 additions*. Specifically:

- the computation of the even indexed coefficients requires 4 multiplications by $\sqrt{1/2}$ and 4 additions,
- the computation of $G=Y_l-Z_l'$ requires 4 subtractions,
- the computation of R requires 16 multiplications and 12 additions (since only the upper 4x4 elements of the T matrix are used), and
- 3 post additions are needed for calculating the required odd-indexed coefficients.

b. In all cases the 4 multiplications by $\sqrt{1/2}$ for the computation of the even indexed coefficients could be saved by absorbing them into the quantisation stage that follows the TDM stage.

c. A number of operations could also be saved if the values of the T matrix were rounded to the closest power of 2. In such a case shifts could be used instead of multiplications. The exploitation of the shift and add operation, existing in all modern DSPs and general purpose CPUs, could increase performance considerably.

4.2 Complexity for the 2-D case

In the case of down-scaling 4 adjacent 8x8 DCT blocks down to one 8x8 block, i.e. down-sizing by a factor of 2 in each dimension, by means of the row-column method, the proposed algorithm has to be applied 24 times (16 times across the rows and 8 times across the columns as shown in Fig. 5a). This results in a computational complexity *of 864 multiplications* and *1032 additions* or a total of 1896 operations.

In the special case that only the upper left 4x4 DCT coefficients of each 8x8 block are non-zero, the algorithm has to be applied 16 times only, i.e. 8 times across each dimension, as shown in Fig. 5b. The corresponding complexity is now *320 multiplications* and *368 additions* or a total of 688 operations.

The above given complexity figures could be further reduced if approximate values, e.g. powers of 2, for elements of the T matrix, were used.

Considering that 11 multiplications and 29 additions are needed for each DCT or IDCT computation [7], a total of 1008 multiplications and 1752 additions or 2760 operations are required for down-scaling four adjacent 8x8 DCT blocks down to one 8x8 block according to the approach proposed in [8]. The comparison of this complexity to that corresponding to Fig. 5a, results in a savings of 864 operations (or 31.3%). In the special case of Fig. 5b, 2072 operations (688 multiplications and 1384 additions) i.e. 75%, are saved.

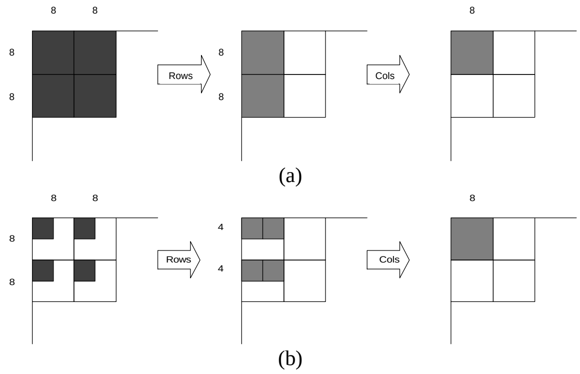

Figure 5. Down-scaling an 8x8 block when (a) all coefficients are non-zero, and (b) only lower 4x4 coefficients are non-zero

5. RESULTS AND CONCLUSIONS

Down-scaling of compressed pictures in the transform domain is not only advantageous from the computational point of view, but from the obtained picture quality as well (Fig. 6). This is due to the fact that a great number of arithmetic and quantisation errors are avoided. The values of the T matrix are off-line calculated to the desired accuracy and the sum of products for the computation of TG can be calculated to the desired accuracy. No intermediate calculation steps of lower accuracy are needed. Experimental comparative results for the down-scaling approaches of Fig. 1, are given in Table I. The test image in each of the cases has been obtained by low-pass filtering (using the filter coefficients suggested by JPEG [9]) and down-sampling by 2 along the horizontal and vertical dimensions the original image in the spatial domain. The SNR values and the file size (i.e. the compression ratio) for each case, with and without quantisation, are shown in Table I. The quantisation table used, was the one given in the JPEG standard for the luminance [9]. It is seen from Table I that the proposed approach outperforms all other methods.

In conclusion, it has been proved that processing digital images in the compressed domain by means of the proposed approach, has many advantages in terms of processing speed, storage efficiency and picture quality.

REFERENCES

[1] V. Bhaskaran: "Mediaprocessing in the Compressed Domain", *Proc. of COMPCON '96*, pp. 204-209, 1996.

[2] N. Merhav and V. Bhaskaran: "Fast Algorithms for DCT-Domain Image Down-Sampling and for Inverse Motion Compensation", *IEEE Trans on Circuits and Systems for Video Technology*, Vol. 7, No. 3, pp. 468-476, June 1997.

[3] Y. Noguchi, D. G. Messerschmitt and S.-F. Chang: "MPEG Video Compositing in the Compressed Domain", *Proc. IEEE ISCAS-96,* 1996.

[4] P. A. A. Assuncao and M. Ghanbari: "Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates", *Proc. IEEE ICASSP'96*, Vol. 4, pp. 1998-2001, May 1996.

[5] J. Bao, H. Sun, T.C. Poon, "HDTV down conversion decoder", *IEEE Trans. On Consumer Electronics*, Vol. 42, No. 3, pp. 402-410, August 1996.

[6] A. Vetro, H. Sun, J. Bao and T. Poon: "Frequency Domain Down-Conversion of HDTV Using Adaptive Motion Compensation", *Proc.1997 IEEE Int. Conf. on Image Processing, ICIP'97*, Santa Barbara, 26-29 Oct. 1997.

[7] C. Loeffler, A. Ligtenberg and G. S. Moschytz: "Practical Fast 1-D DCT Algorithms with 11 Multiplications", Proc. *IEEE Int. Conf. on ASSP (ICASSP-89),* pp. 988-991.

[8] A. N. Skodras and C. A. Christopoulos, "Down-Scaling of Compressed Images in the DCT Domain", *Proc. of EUSIPCO'98,* Rodos, Greece, 8-11 Sep. 1998.

[9] K. R. Rao and J. J. Hwang: "Techniques and Standards for Image, Video and Audio Coding", *Prentice Hall*, 1996.

Figure 6. Frame 20 of the "foreman" sequence (above) down-scaled by 2 in each dimension according to the approach of Fig. 1a (a), Fig. 1b (b), Fig. 1c (c)

Table I

Comparative results for the down-scaling by 2 in each dimension of various pictures (SNR in dB)

Image	**Traditional (Fig. 1a)**		**Frequency Masking (Fig. 1b)**		**Proposed (Fig. 1c)**	
	w/o Q	w Q	w/o Q	w Q	w/o Q	w Q
foreman	58.8116	25.7464	26,44	26,84	26,16	25,76
(176x144)	5,3kB	1,3kB	6,2kB	4,3kB	6,4	1,7kB
lenna	56.358	26.0433	26,69	26,86	26,55	26,19
(512x512)	44,7kB	8,8kB	61,9kB	36,7kB	59,0kB	12kB
peppers	56.5958	26.3926	26,85	26,83	26,63	25,76
(512x512)	47,1kB	9,2kB	62,5kB	36.3kB	59,8kB	12kB
sailboat	57.1996	24.401	24,81	25,21	24,60	24,34
(512x512)	52,2kB	11,2kB	65,7kB	42.2kB	64,7kB	15,7kB
target	49.0976	23.2849	17,82	18,13	18,11	18,95
(512x512)	28,9kB	8,5kB	42,9kB	33.6kB	36,0kB	12,2kB
hotel	54.6589	22.4551	22,64	22,98	22,44	22,38
(720x576)	75,1kB	16,6kB	103kB	60.6kB	96,9kB	23,1kB