DOWN - SAMPLING OF COMPRESSED IMAGES IN THE DCT DOMAIN§

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
skodras@cti.gr
[3]Ericsson Telecom AB, Compression Lab, TN/ETX/PN/XML, S-126 25 Stockholm, Sweden
ch.christopoulos@clab.ericsson.se

ABSTRACT

An efficient down-sampling algorithm of DCT (discrete cosine transform) compressed images is presented in this communication. The algorithm operates directly on the compressed data, 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. Its structure is highly regular, resulting in efficient software and hardware implementations. 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, multimedia, JPEG, MJPEG, MPEG, H.26x

1 INTRODUCTION

The emergence of the compression standards JPEG, MPEG, H.26x has enabled many consumer and business multimedia applications, where the multimedia content is disseminated in its compressed form. However, many applications require processing of the multimedia content prior to presentation. A very frequent process is that of down-sampling (down-scaling, down-sizing) the compressed image, as it happens in the following cases:

Image and video browsing: In applications, such as image and video browsing, 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].

Video compositing: 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. Compositing video 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. Compressed domain based down-sampling can be used to implement an efficient picture-in-picture system for MPEG compressed video and can result in significant savings [1, 2].

Transcoding: Efficient transcoding could cope with different quality of services in the case of multi-point communications over POTS, ISDN, and ADSL lines [3].

HDTV to SDTV conversion: A HDTV down conversion decoder can decode the Grand Alliance HDTV bitstreams and display them on SDTV or NTSC monitors [4, 5].

Traditional approaches for down-scaling rely on decompressing the bitstreams first and then applying the desired processing function (re-compression). In the present communication, an efficient down-sampling technique is presented, in which full transition to the spatial domain is avoided.

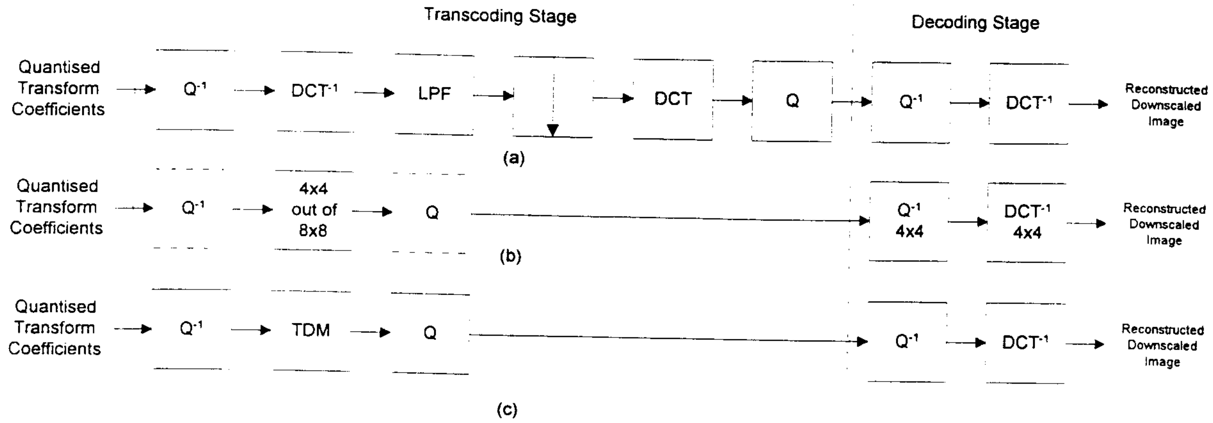
2 THE APPROACH

The down-sampling of a still image in the spatial domain consists of two steps. First the image is filtered by an anti-aliasing low pass filter and then it is sub-sampled by a 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 by inverse DCT (IDCT or DCT^{-1}) and then undergo the procedure of filtering and down-sampling as illustrated in Fig 1a.

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 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 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, referred to as *frequency masking*

§ This work was supported by a Joint Research Grant of the British Council and the GSRT and by the EU ACTS project SCALAR (AC077).

approach, does not result in significant compression gains and 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. Furthermore, this method results in significant amount of blocking effects and distortions, due to the poor approximations introduced by simply discarding higher order coefficients [4].



Notes: i. DCT / DCT^{-1} denote the forward and the inverse Discrete Cosine Transform on 8x8 data, unless otherwise stated
ii. The steps of lossless coding and decoding of the quantised data are not depicted in the above diagrams
iii. TDM stands for Transform Domain Manipulation

Figure 1. Block diagrams of the down-sampling approaches of compressed images

This direct approach would be more useful if we had 16x16 DCT blocks and were keeping the 8x8 DCT coefficients in order to obtain the down-sampled. 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. One way to compute the 16x16 DCT coefficients is to apply inverse DCT in each of the 8x8 blocks and reconstruct the image. Then the DCT in blocks of size 16x16 could be applied and the 8x8 out of the 16x16 DCT's coefficients of each block could be kept. This would lead to a complete decoding (performing 8x8 IDCT's) and re-transforming by 16x16 DCT's, something that would require 16x16 DCT hardware or software. However, 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. This would also mean that by avoiding the computation of DCT's of size 16x16, the memory requirements could also be reduced.

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 (Fig. 2). Only N/2-point transformations are required for this computation and all operations are performed in the compressed domain (transform domain manipulation, TDM).

3 THE ALGORITHM

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

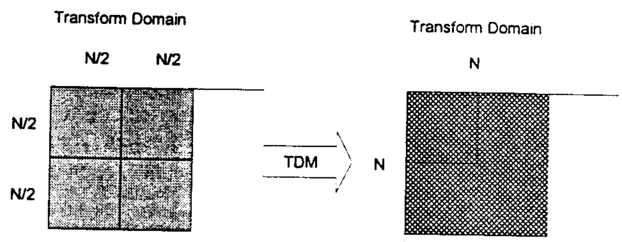


Figure 2. Schematic diagram of the proposed approach (N/2 is usually equal to 8)

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 $N = 2^m$. 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.

3.1 Definitions

The normalised forward DCT (DCT-II) and inverse DCT (IDCT) of the length-N sequence x_n are given by the following equations [6]:

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

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

where $\varepsilon_k = 1/\sqrt{2}$ *for* $k = 0$ and $\varepsilon_k = 1$ *for* $k \neq 0$. Notice 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, where in this case N is substituted by N/2.

3.2 Theoretical Analysis

The computation is performed separately for the even- and the odd-indexed coefficients.

i. Even-indexed coefficients

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

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

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

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

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

ii. Odd-indexed coefficients

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

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

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

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

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 $2\cos(2n+1)\pi/2N$. 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. 3. It is seen that this graph has the familiar structure of the fast transform algorithms, resulting in an efficient implementation in software and hardware. Down-sampling by a factor of 2 implies that only coefficients 0, 2, 4, 6, 1, 3, 5, 7 have to be calculated, simplifying further the implementation.

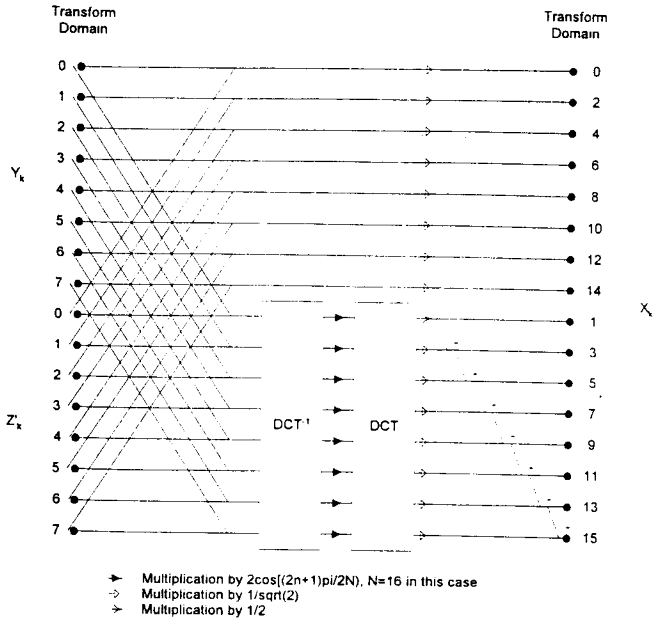


Figure 3. Flow graph of the proposed approach for N=16

3.3 Computational Complexity

For the computation of the even-indexed coefficients only N/2 additions are necessary, while for the computation of the odd-indexed coefficients N/2+(N/2-1) additions, N/2 multiplications, one length-N/2 IDCT and one length-N/2 DCT are needed. Thus, the computation of all N coefficients requires a total of M_N multiplications and A_N additions, where $M_N=(N/2)\log_2 N$ and $A_N=(3N/2)\log_2 N-N+1$. This complexity is equal to that of a length-N fast DCT computation according to well known fast algorithms [7-9]. Down-sampling by a factor of 2 means that the above given complexity figures are further reduced.

Compared to the traditional approach of decompressing the two N/2 sequences and re-compressing the filtered and undersampled length-N/2 sequence, one length-N/2 DCT computation is saved. This savings could be even greater if pruning techniques where incorporated in the computations [10]. In the 2-D case, the savings are more significant. According to the traditional approach of Fig. 1a, 80 DCT's of length-8 are needed for the down-sampling of 4 adjacent 8x8 transform blocks. (Row-column wise calculations are considered). This figure reduces down to 48 DCT's of length-8 each, if the proposed approach is used., i.e. a total of 32 DCT calculations is saved. This means that for the down-sampling of a compressed CIF image, which consists of 99 macroblocks, down to QCIF, 3168 DCT's of length-8 are saved.

The comparison of the proposed approach to that of [4] reveals certain advantages of the first. Specifically, the calculation of the odd-indexed coefficients according to [4] requires two matrix multiplications and a number of additions. This results in a total computational complexity of $N^2/2$ multiplications and $N^2/2+N-2$ additions, which is in the order of N^2 and not of $N\log_2 N$, as in the proposed approach. In addition, that algorithm possesses an irregular structure, it cannot be implemented in a fast transform way and it requires multiplications by both cosine and sine functions.

The proposed approach has a similar complexity to that presented in [11]. However, the proposed method is efficiently implemented in software and hardware by means of existing optimised DCT's and IDCT's, i.e. the proposed processing integrates easily into any block DCT-based image compression system. Besides, it is applicable for any number of points N (N being an even number) and not restricted and optimised to 8 points only. Also, the down-sampling factor can be different than 2 and also it can be different for each dimension of the image.

4 COMPUTER SIMULATION RESULTS AND CONCLUSIONS

Down-sampling of compressed images in the transform domain is not only advantageous from the computational point of view, but from the obtained picture quality as well. This is due to the fact that a great number of arithmetic and quantisation errors are avoided. Comparative experimental results of the approaches depicted in Fig. 1, are given in Table I. The reference image in each of the cases was obtained by down-sampling the original image by means of Photoshop 4.0. The SNR values and the file sizes (i.e. the compression ratios) for each case are included in the Table. The quantisation matrix used, is the one given in the JPEG standard for the luminance [6,12]. It is seen from this Table that the proposed approach outperforms all other methods. This is further corroborated by the subjective comparison of the images. In addition, taking into consideration the computational efficiency of the proposed approach (more than 40% savings as compared to the traditional method), we derive that this could be effectively used in all cases of down-sampling of compressed images.

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 *image quality*.

Acknowledgements: The authors would like to thank J. Poyhtari, S. Koski and V. Fotopoulos for their contribution to the experimental results.

REFERENCES

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

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

[3] 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.

[4] 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.

[5] 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.

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

[7] H. S. Hou: "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform", *IEEE Trans on ASSP*, Vol. ASSP-35, pp. 1445-1461, Oct. 1987.

[8] S. C. Chan and K. L Ho: "Direct Methods for Computing Discrete Sinusoidal Transform", *IEE Proceedings*, Vol. 137, Pt. F, No. 6, pp. 433-442, Dec. 1990.

[9] C. W. Kok: "Fast Algorithm for Computing Discrete Cosine Transform", *IEEE Trans on Signal Processing*, Vol. 45, No. 3, pp. 757-760, Mar. 1997.

[10] A. N. Skodras: "Fast Discrete Cosine Transform Pruning", *IEEE Tran on Signal Processing*, Vol. 42, No. 7, pp. 1833-1837, July 1994.

[11] 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.

[12] V. Bhaskaran and K. Konstantinides: "Image and Video Compression Standards: Algorithms and Architectures", Second Edition, *Kluwer Academic Publishers*, 1997.

Table I

Comparative results for down-sampling of compressed images by 2 in each dimension (SNR in dB, file sizes in KBytes)

Down-Sampled Image	Traditional Approach (Fig. 1a)	Frequency Masking (Fig. 1b)	Proposed Approach (Fig. 1c)
Lenna (256x256)	23.35 8.8KB	32.42 36.7KB	27.14 12KB
peppers (256x256)	23.57 9.2KB	31.97 36.3KB	27.58 12KB
sailboat (256x256)	21.59 11.2KB	31.56 42.2KB	25.54 15.7KB
foreman (88x72)	22.89 1.3KB	32.82 4.3KB	26.67 1.7KB
news (88x72)	16.18 1.2KB	27.29 3.9KB	20.87 1.7KB
target (256x256)	18.57 8.5KB	23.21 33.6KB	23.96 12.2KB
hotel (360x288)	19.57 16.6KB	29.85 60.6KB	24.02 23.1KB