Image Coding and Compression with Sparse 3D Discrete Cosine Transform

Hamid Palangi[1], Aboozar Ghafari[1], Massoud Babaie-Zadeh[1], and Christian Jutten[2,⋆]

[1] Electrical Engineering Department and Advanced Communications Research Institute (ACRI), Sharif University of Technology, Tehran, Iran
[2] GIPSA-lab, Grenoble, France
hmd_palangi@yahoo.com, aboozar412@yahoo.com
mbzadeh@yahoo.com, Christian.Jutten@inpg.fr

Abstract. In this paper, an algorithm for image coding based on a sparse 3-dimensional Discrete Cosine Transform (3D DCT) is studied. The algorithm is essentially a method for achieving a sufficiently sparse representation using 3D DCT. The experimental results obtained by the algorithm are compared to the 2D DCT (used in JPEG standard) and wavelet db9/7 (used in JPEG2000 standard). It is experimentally shown that the algorithm, that only uses DCT but in 3 dimensions, outperforms the DCT used in JPEG standard and achieves comparable results (but still less than) the wavelet transform.

Keywords: Sparse image coding, 3 dimensional DCT, wavelet transform.

1 Introduction

In data compression reducing or removing redundancy or irrelevancy in the data is of great importance. An image can be lossy compressed by removing irrelevant information even if the original image does not have any redundancy [1]. The JPEG standard [2] which is based on Discrete Cosine Transform (DCT) [3], is widely used for both lossy and lossless image compression, especially in web pages. However, the use of the DCT on 8×8 blocks of pixels results sometimes in a reconstructed image that contains blocking effects (especially when the JPEG parameters are set for large compression ratios). Consequently, JPEG2000 was proposed based on Discrete Wavelet Transform (DWT) [4,5] which provides more compression ratios than JPEG for comparable values of Peak Signal-to-Noise Ratio (PSNR).

⋆ This work has been partially funded by Iran National Science Foundation (INSF) under contract number 86/994, by Iran Telecommunication Research Center (ITRC) and also by center for International Research and Collaboration (ISMO) and French embassy in Tehran in the framework of a GundiShapour program. We would also like to thank National Elite Foundation of Iran for their financial supports.

T. Adali et al. (Eds.): ICA 2009, LNCS 5441, pp. 532–539, 2009.

Compression systems are typically based on the assumption that the signal can be well approximated by a linear combination of a few basis elements in the transform domain. In other words, the signal is sparsely represented in the transform domain, and hence by preserving a few high magnitude transform coefficients that convey most of information of the signal and discarding the rest, the signal can be effectively estimated. The sparsity of representation depends on the type of the transform used and also the signal properties. In fact the great variety in natural images makes impossible for any fixed 2D transform to achieve good sparsity for all cases [1]. Thus, the commonly used orthogonal transforms can achieve sparse representations only for particular image patterns.

In this article an image coding strategy based on an enhanced sparse representation in transform domain is studied which is based on a recently proposed approach [6] for image denoising. Based on this approach an enhanced sparse representation can be achieved by grouping similar 2D fragments of input image (blocks) into 3D data arrays. *We have used this approach with a 3D DCT transform for image coding purposes*. The procedure includes three steps: 3D DCT transformation of a 3D array, shrinkage of the transform domain coefficients, and inverse 3D DCT transformation. Due to the similarity between blocks in a 3D array, the 3D DCT transform can achieve a highly sparse representation. Experimental results demonstrate that it achieves outstanding performance in terms of both PSNR and sparsity.

The paper is organized as follows. The next section describes the main idea and discusses its effectiveness. The algorithm is then stated in Section 3. Finally, Section 4 provides some experimental results of algorithm and its comparison with DWT.

2 The Basic Idea

The basic idea of this article is achieving an enhanced sparse representation by grouping similar 2D fragments of the input image into 3D arrays, and then using a 3D DCT transformation to transform 3D arrays. In fact this idea has been introduced in [6] for image denoising and has been shown to outperform state of the art denoising algorithms [6]. Then, in this article, we consider applying an approximately similar idea for image compression and study its performance.

A simple justification for the effectiveness of the proposed idea is as follow [6]:

- Assume that the grouping is done, i.e. similar blocks are placed in groups and a 2D DCT transformation is used for each group.
- In each group we have similar blocks and hence after transformation we will have the same number of *high-magnitude* coefficients for each block in a group, say α high-magnitude coefficients for each block.
- Assuming n blocks in each group, we will have $n\alpha$ high-magnitude coefficients in that group. In other words this group can be represented by $n\alpha$ coefficients.
- Now we should perform a 1D DCT transform on the third dimension (along each row) of each group.

- Components of this row are similar (because only similar blocks are in this group), i.e. there is a kind of similarity for all members of the row.
- As an example, after using 1D DCT transform the first or second coefficients of this row will be high-magnitude (because of the compaction property of DCT transform). This means that the whole group can be represented by α or 2α coefficients instead of $n\alpha$ coefficients (i.e. a much more sparse representation).

3 The Algorithm

Based on the main idea of the previous section, the algorithm is as follows:

- Grouping:
  1. Block input image to 8×8 fragments with one pixel overlap
  2. Save blocks in Y.
- while Y is not empty
  for i=1,...,*Number Of Fragments:*
  1. *Choose one block as a reference block (Y_r).*
  2. *Calculate* $d(Y_r, Y_i) = \frac{\|Y_r - Y_i\|_2^2}{N^2}$ *were* Y_i *is the* i^{th} *block.*
  3. *if* $d(Y_r, Y_i) \leq$ *Threshold Distance*
    - *Assign* Y_i *to a group.*
    - *Remove* Y_i *from* Y*.*

  Save resulted group in a 3D array named Group Array
- 3D DCT
  1. for every group of *Group Array*
  Perform a 2D DCT on that group
  2. *Perform a 1D DCT on the third dimension of Group Array*
- Shrinkage
  1. if *Transform Domain Coefficients* $\leq$ *Hard Threshold*
  Discard that coefficient.
- Calculate inverse 3D DCT transform
- Place each decoded block in its original position.

Remark 1. For image blocking we have used (as suggested in [6]) blocks with one pixel overlap to increase PSNR and also overcome the blocking effects resulted from image blocking.

Remark 2. Grouping can be realized by various techniques; e.g., K-means clustering, self-organizing maps, fuzzy clustering, vector quantization and others. A complete overview of these approaches can be found in [7]. A much simpler and effective grouping of mutually similar signal fragments can be realized by matching as discussed in [6]. In matching we want to find blocks which are similar to a reference block. It needs a search between all blocks to find blocks similar to a given reference block. The fragments whose distance from the reference block is smaller than a grouping threshold are stacked in a group. Any image fragment can be used as a reference block and thus a group can be constructed for it. The similarity between image fragments is typically computed as the inverse of some distance measure. Hence, a smaller distance implies higher similarity. In particular, we use the same distance proposed in [6] which is defined below as a measure of dissimilarity.

$$d(Y_r, Y_i) = \frac{\|Y_r - Y_i\|_2^2}{N^2} \tag{1}$$

In (1), Y_r is the reference block from which the distance of the i^{th} block (Y_i) is calculated. N is the size of the chosen blocks (for all our simulations 8×8 blocks are used, that is $N = 8$). This distance can also be computed in the transform domain; i.e., we can do the grouping after 2D transformation (*transform domain grouping*) and then perform a 1D DCT on third dimension along the rows. This idea was tested and the changes in PSNR were in the order of 10^{-2} with the same sparsity.

Remark 3. Note that in the 3D DCT transformation at first a 2D DCT transform is applied on groups and then a 1D DCT transform is performed on the third dimension, which is on the rows of every group. Both of the used DCT transformations are complete DCT transforms.

Remark 4. In the shrinkage we have used a hard thresholding methodology; i.e., we have simply discarded those coefficients in the transform domain whose magnitude is less than some fixed threshold.

Remark 5. Obviously a straightforward implementation of this algorithm is highly computationally demanding. In order to realize a practical and efficient algorithm, some constraints should be considered. For example to reduce the number of processed blocks we can only use a limited number of reference blocks by choosing reference blocks between every N_1 blocks. In this way we will have *(Total Number Of Blocks)*/N_1 reference blocks. A complete set of such ideas to reduce the computational complexity and increase the speed of the algorithm can be found in [6].

4 Simulation Results

In this section, we study the performance of the presented approach and compare it with 2D DCT and wavelet transform for image compression. The wavelet transform that we have used in this comparison is db9/7 which is used in JPEG2000 standard [8]. This wavelet transform is also used in *FBI* fingerprint database [9]. The images which have been used for all simulations are 441×358 *Tracy* and *Barbara* images. Our criterion to measure sparsity is simply the ℓ^0 norm, that is, the number of nonzero coefficients. The simulation results are as presented in Table 1 (note that all transforms mentioned in this table are complete). In this table d_G stands for distance used for grouping and th_C stands for the hard threshold used to shrinkage the coefficients.

As it can be seen in Table 1, generally the performance of 3D DCT is better than 2D DCT (an improvement about $2dB$ in PSNR with the same degree of sparsity). Note that in the last row of the table for Tracy image the results of 2D DCT and 3D DCT are very close to each other. The reason is that in this case the distance threshold used for grouping is very high (245) and therefore we don't have an exact grouping; i.e., similarity of the third dimension is not high and this yields weak results with 3D DCT. Generally it can be deduced from the table that *with more precise grouping we will have better results but only in terms of PSNR. If we want to achieve high sparsity at the same time, we would need some sort of balance between the number of nonzero elements (ℓ^0 norm as a criterion to measure sparsity) and PSNR*. This result was expected because the main idea was based on the similarity between blocks and if this similarity increases then the similarity that exists in the third dimension of every array will increase and therefore more compaction can be achieved. The best result from 3D DCT idea has been shown in bold in the table. It should also be noted that generally 3D DCT results are weaker than results obtained by wavelet transform in terms of PSNR with the same sparsity for Tracy image. It is interesting to note the results for Barbara image. In this case results of 3D DCT are closer to (or even better than) those of the Wavelet transform.

Although the complexity of wavelet transforms depends on the size of filters and the use of floating point vs integer filters, wavelet transforms are generally more computationally complex than the current block- based DCT transforms [10].

Table 1. 3D DCT Versus 2D DCT and Wavelet db9/7

$TestImage$	d_G	th_C	ℓ^0 Norm	PSNR in dB		
				2D DCT	4 Level Wavelet db 9/7	3D DCT
Tracy	10	50	3327	32.8294	38.2536	36.7798
	10	30	4703	36.3101	39.8067	38.2879
	10	**20**	**6175**	**37.9671**	**40.9493**	**39.3012**
	20	20	5608	37.3384	40.5386	39.0774
	50	20	5189	36.8697	40.1906	38.7131
	245	20	6009	37.8331	40.8519	38.1795
Barbara	10	50	9027	27.5271	29.4975	29.3795
	10	30	15068	30.3920	32.0041	32.2399
	10	**20**	**22138**	**33.9195**	**34.5035**	**34.6991**
	20	20	21758	33.6757	34.5024	34.6826
	50	20	21205	33.5548	34.4704	34.6386
	245	20	20273	33.1230	34.1353	33.9469

Figures 1 and 3 show the original images and their decoded versions using wavelet db9/7, DCT and 3D DCT for the bold rows of Table 1. As it can be seen from these figures, the blocking effect when 2D DCT is used is clearly visible. But when 3D DCT is used there is almost no blocking effect. In Figs. 2 and 4, a comparison between the performances of these three transforms is made for both test images.

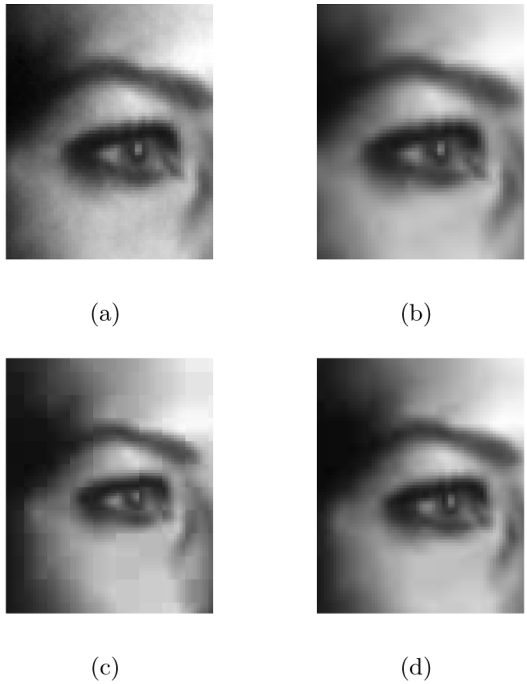

Fig. 1. The zoomed results of using various transforms for *Tracy* test image (a) The original image (b) Decoded image after compression using wavelet db9/7 (c) Decoded image after compression using 2D DCT (d) Decoded Image after compression using 3D DCT

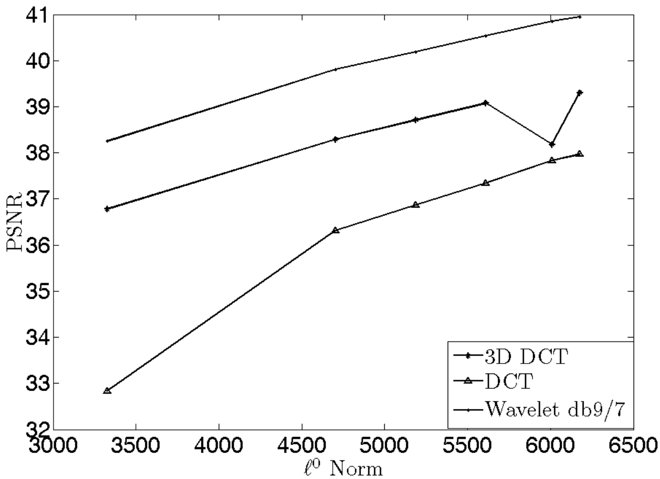

Fig. 2. Comparison between DCT, 3D DCT and wavelet db9/7 for *Tracy* test image

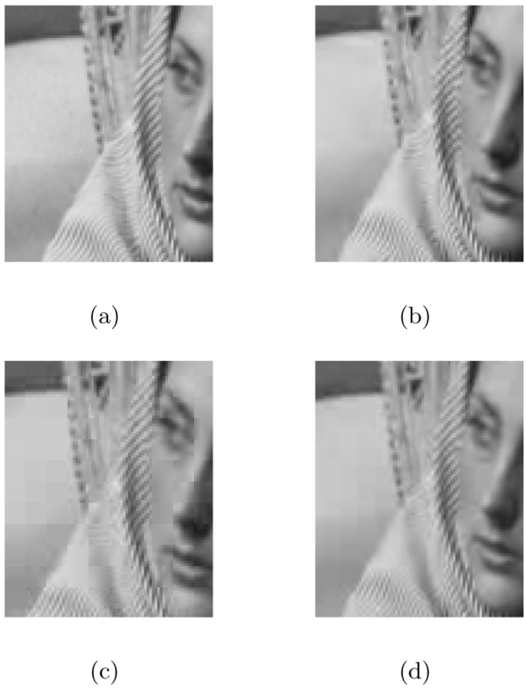

Fig. 3. The zoomed results of using various transforms for *Babara* test image. (a) The original image (b) Decoded image after compression using wavelet db9/7 (c) Decoded image after compression using 2D DCT (d) Decoded Image after compression using 3D DCT.

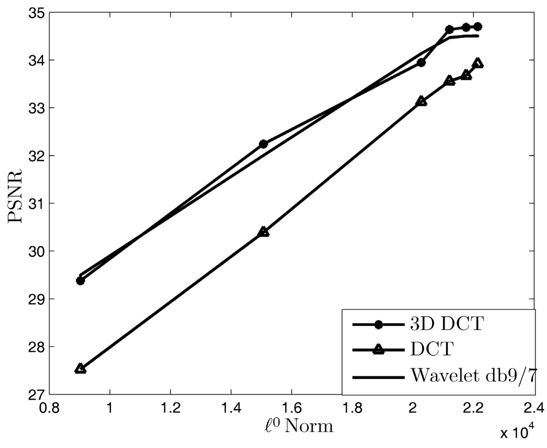

Fig. 4. Comparison between DCT, 3D DCT and wavelet db9/7 for *Barbara* test image

5 Conclusions

In this article the idea of a recently proposed approach for image denoising was studied to be used for image compression. This idea is based on 3D DCT transform to enhance the sparsity of the coefficients. Our simulations show that the usage of this idea enhances the results compared to 2D DCT transform (used in JPEG), and gives the results comparable (but still below) what is obtained using wavelet transform (used in JPEG2000).

References

1. Salomon, D.: Data Compression, The Complete Reference, 3rd edn. Springer, New York (2004)
2. Wallace, G.K.: The JPEG still Picture Compression Standard. Commun. ACM 34(4), 30–44 (1991)
3. Rao, K., Yip, P.: Discrete Cosine Transform, Algorithms, Advantages, Applications. Academic, New York (1990)
4. Taubman, D., Marcellin, M.: JPEG 2000: Image Compression Fundamentals, Standards and Practice. Kluwer, Norwell (2002)
5. Topiwala, P.N.: Wavelet Image and Video Compression. Kluwer, Norwell (1998)
6. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering. IEEE Transactions on Image Processing 16(8), 2080–2095 (2007)
7. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
8. Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I.: Image Coding using Wavelet Transforms. IEEE Transactions on Image Processing 1(2), 205–220 (1992)
9. Wickerhauser, M.V.: Adapted wavelet analysis from theory to software algorithms. AK Peters, Ltd., Wellesley (1994)
10. McDowell, D.: Why Do We Care About JPEG 2000? Image Science and Technology Reporter 14(3) (1999)