International Workshop on Systems, Signals and Image Processing IWSSIP'97, May 28-30, Poznań, Poland

A new approach to generation of shape-adaptive transforms

Ryszard Stasiński

Poznań University of Technology
Institute of Electronics and Communications

Janusz Konrad

INRS-Télécommunications
Institut National de la Recherche Scientifique

Abstract

In the paper[1] we describe a new approach to generation of orthogonal transforms that self-adapt to arbitrary shapes. The new algorithms are derived from flowgraphs of standard fast transform algorithms by a suitable modification of their substructures. For simplicity we show how to derive a shape-adaptive transform from the discrete Walsh-Hadamard transform (DWHT) flowgraph. We compare performance and computational complexity of new algorithms with those of several well-known approaches. It can be clearly seen that for DCT the proposed approach gives a very beneficial performance/complexity ratio compared to other well-known techniques.

1 Introduction

All image and video compression standards today are based on uniform partitioning of data into rectangular blocks. The schemes suffer from two basic drawbacks. First, for very low bit rates block boundaries emerge. Secondly, bit flow organization around rectangular blocks does not allow for flexible manipulation of data streams. One of approaches to alleviate the problems consists in image partitioning into irregularly-shaped regions, followed by separate compression of each region. The benefit is that with bit rate reduction distortion increases uniformly within the whole region since it is treated as an entity. Additionally, a segment-sequential transmission or storage becomes possible facilitating such functionalities as object-by-object progressive transmission or database query by object. Two approaches to the region transformation have been dominant to date: extrapolation of data followed by a standard transform and shape-adaptive transform [2, 3, 5].

In the paper we propose a new approach to fast shape-adaptive orthogonal transforms generation. From flowgraphs of fast DWHT and DCT algorithms we derive their new shape-adaptive variants, section 2. We study the complexity of the shape-adaptive DCT-like transform with respect to some well-known algorithms, section 3. In section 4 we show that the new method has good energy compaction properties.

[1] The research has been supported by a KBN grant "Nowe techniki kompresji barwnych cyfrowych sekwencji wizyjnych dla kanałów o różnej przepustowości", and by the Ministère de l'Éducation du Québec within the program "Coopération Québec-Europe Centrale et Orientale".

2 New algorithm

For our purposes we can assume that image samples either belong to a region, or to its background. The underlying idea in the new shape-adaptive algorithm is the replacement of some operations of a fast orthogonal transform algorithm in such a way that the background data samples are ignored, while the overall transform remains orthogonal.

Consider the following matrix representation of a fast orthogonal transform algorithm:

$$\mathbf{T} = \mathbf{T}_K \cdot \ldots \mathbf{T}_{i+1} \cdot \mathbf{T}_i \cdot \mathbf{T}_{i-1} \cdot \ldots \mathbf{T}_2 \cdot \mathbf{T}_1. \qquad (1)$$

Let's assume that products $\mathbf{T}_K \cdot \ldots \mathbf{T}_{i+1}$, and $\mathbf{T}_{i-1} \cdot \ldots \mathbf{T}_2 \cdot \mathbf{T}_1$ are proportional to orthogonal matrices, and that the matrix $\mathbf{T}_i$ is orthogonal. If the matrix $\mathbf{T}_i$ is replaced by another orthogonal matrix $\mathbf{Q}_i$, then obviously the new transform:

$$\widehat{\mathbf{T}} = \mathbf{T}_K \cdot \ldots \mathbf{T}_{i+1} \cdot \mathbf{Q}_i \cdot \mathbf{T}_{i-1} \cdot \ldots \mathbf{T}_2 \cdot \mathbf{T}_1.$$

represents also an orthogonal transform, although different from that represented by $\mathbf{T}$.

We need to define the new matrix $\mathbf{Q}_i$ in such a way that the overall transform $\widehat{\mathbf{T}}$ be shape-adaptive. We impose the following two requirements:

1. To be defined inside a region, $\widehat{\mathbf{T}}$ must not mix samples inside and outside of the region boundary.

2. To maximize the computational efficiency of the algorithm, $\widehat{\mathbf{T}}$ must not process the background samples at all.

Let $\mathbf{x} = [x(0) \ldots x(N-1)]$ be the input vector of the i-th step of a fast orthogonal transform algorithm described by an $N \times N$ matrix $\mathbf{T}_i$. Suppose that the input sample $x(k)$ belongs to background and we wish that it appear at position j at the output. Then, in order that $x(k)$ not be "mixed" with samples from the region, column k of $\mathbf{Q}_i$ should consist of zeros, except for $[\mathbf{Q}_i]_{j,k}$ equal 1 to pass $x(k)$ to the output. At the same time second condition implies that all entries in the row j, except $[\mathbf{Q}_i]_{j,k}$, can be set to zero. So, the matrix $\mathbf{Q}_i$ takes the following form:

$$\mathbf{Q}_i = \begin{bmatrix} q_{0,0} & \cdots & q_{0,k-1} & 0 & q_{0,k+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ q_{j-1,0} & \cdots & q_{j-1,k-1} & 0 & q_{j-1,k+1} & \cdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots \\ q_{j+1,0} & \cdots & q_{j+1,k-1} & 0 & q_{j+1,k+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}.$$

The orthogonality of $\mathbf{Q}_i$ is guaranteed if the matrix $\mathbf{Q}'_i$ obtained from $\mathbf{Q}_i$ by the rejection of column k and row j is orthogonal as well. If there are more samples at the input to $\mathbf{T}_i$, the same idea can be applied to $\mathbf{Q}'_i$ again.

The special case of N=2 is a very important one since a 2×2 matrix $\mathbf{Q}_i$ describes a two-point butterfly. Two-point butterflies are building blocks of many fast orthogonal transform algorithms. Namely, when any input sample belongs to the background the $\mathbf{T}_i$'s simplify to

$$\mathbf{V}_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ or } \mathbf{V}_S = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (2)$$

With the above permuting butterflies, background samples can be passed through at the same position ($\mathbf{V}_I$) or they can be moved around ($\mathbf{V}_S$) without affecting samples from the region.

Due to its particular simplicity, we will use the Discrete Walsh-Hadamard (DWHT) algorithm to demonstrate the proposed approach. Probably the most elegant definition of the N-point non-normalized DWHT when N is a power of 2 is given by the following recursion [1]:

$$\mathbf{T}_N = \begin{bmatrix} \mathbf{T}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_{N/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix},$$

where subscripts show dimensions of submatrices, and $\mathbf{T}_1 = [1]$. For $N = 8$ the recursion stops after formulation of the algorithm as a product of 3 matrices: $\mathbf{T} = \mathbf{T}_3 \cdot \mathbf{T}_2 \cdot \mathbf{T}_1$, where e.g.:

$$\mathbf{T}_1 = \begin{bmatrix} 1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&1 \\ 1&0&0&0&-1&0&0&0 \\ 0&1&0&0&0&-1&0&0 \\ 0&0&1&0&0&0&-1&0 \\ 0&0&0&1&0&0&0&-1 \end{bmatrix},$$

Operations performed by the three matrices above can be described by a flowgraph presented in Fig. 1. The flowgraph shows that each algorithm stage consists of four independent operations (butterflies). For example, the third butterfly from the first stage (marked by bold lines in Fig. 1) can be described by the following matrix:

$$\mathbf{T}_{1,3} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&1&0&0&0&-1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}. \quad (3)$$

Fig. 2 shows a two-point multiplierless butterfly and its equivalent form. Based on the equivalent form from Fig. 2 matrix $\mathbf{T}_{1,3}$ can be decomposed into two ones, an orthogonal matrix:

$$\begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&\sqrt{2}/2&0&0&0&\sqrt{2}/2&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&\sqrt{2}/2&0&0&0&-\sqrt{2}/2&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix},$$

followed by a diagonal one performing multiplications by $\sqrt{2}$. So, if sample $x(6)$ at the input to $\mathbf{T}_{1,3}$ belongs to background, then there exist only two possible replacements for $\mathbf{T}_{1,3}$:

$$\mathbf{Q}^I_{1,3} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&\sqrt{2}&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix},$$

and

$$\mathbf{Q}^S_{1,3} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&\sqrt{2}&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}.$$

Note that in column 6 the non-zero entry equals 1 although it should be $\sqrt{2}$. This is done to save computations as value of background sample is irrelevant.

At the output of the first stage the background sample $x(6)$ will appear as a sample with index 2 or 6. This means that the corresponding butterfly from the second stage should be replaced by a permuting butterfly. The same reasoning applies to the subsequent stages. The resulting DWHT-based transform completely ignores the background sample $x(6)$.

3 Implementation and complexity of the new method

Flowgraphs of DCT algorithms are not so regular as that of the DWHT one [6], nevertheless, the only important complication is due to the appearance of non-trivial orthogonal butterflies, which can be transformed into permuting butterflies, too. The shape-adaptive algorithm consists of two stages. The first stage generates a decision table. In the second stage calculations are performed; each butterfly is preceded by a switch that decides what type of operation is done (usual DCT or permuting butterfly). The switch uses the decision table from the first stage. So, our estimate is that the computational complexity of the new algorithm is approximately that of two DCT algorithms.

Among other methods the lowest computational complexity is attained by data extrapolation techniques followed by a rectangular DCT; their complexity is that of one DCT. The method of Sikora and Makai [5] is the only shape-adaptive method that have computational complexity comparable to the new algorithm, when small data blocks and fast DCT algorithms are used. However, a technical realization of all necessary DCT algorithms is complicated, while computation of DCTs directly from the definition makes Sikora's method computationally complex.

The computational complexity of other shape--adaptive transform methods tested here is much greater; for KLT and Gilge method it is of the order of $O(N^6)$ for $N \times N$-point segments. For comparison, complexity of Sikora's method computed from the DCT definition is $O(N^3)$, while that of the DCT and new method is $O(N^2 \log_2 N)$.

4 Experimental results

To evaluate the performance of the new algorithm we have performed experiments with synthetic region shapes applied to synthetic data. In addition to the proposed DCT-like shape-adaptive algorithm we have software-simulated and tested 6 approaches: KLT, Gilge's approach, Sikora's approach, and DCT with extrapolation based on mirror-image extension and zero padding.

Fig. 3 shows the basis restriction error ε as a function of p, the fraction of the highest-energy coefficients used for image reconstruction, for two synthetic shapes: regular "ellipse" and highly irregular "atol". To generate the synthetic data we have used a 2-D AR process based on Markov-1 model in horizontal and vertical directions with model parameter $\rho = 0.9$, similarly to [4]. The KLT clearly outperforms the other methods. Note the very close performance of Sikora's and Gilge's methods in the case of ellipse but better performance of Gilge's approach (by about 2dB) for atol. The new DCT-like algorithm performs about 1dB below Gilge's and Sikora's methods for ellipse, while for atol it attains Sikora's performance for lower compression but loses up to 1dB for higher compression. The mirror-image and zero-padding DCTs perform much worse at lower compression for ellipse; for higher compression, however, mirror-image DCT performs similarly to our algorithm but in this range all the differences are small anyway. For atol mirror-image DCT performs slightly better than our algorithm at high compression but loses up to 1dB for lower compression.

5 Summary and conclusions

We have presented a new class of fast shape-adaptive orthogonal transforms. We have discussed an example from this class, a shape-adaptive transform derived from DCT flowgraph. We have compared its computational complexity with those of other well-known shape-adaptive transforms and we have evaluated its performance experimentally.

The method of Sikora [5] seems to be currently the most promising approach for shape-adaptive transformation. However, when applied to coding of large regions the technique becomes computationally complex. In contrast, the proposed algorithm has the complexity of about two DCTs, only. Algorithm's performance has been shown to be slightly inferior to that of Sikora's method.

References

[1] N. Ahmed, K.R. Rao, "Orthogonal transforms for digital signal processing", Springer-Verlag, Berlin 1975.

[2] S.-F. Chang and D. Messerschmitt, "Transform coding of arbitrarily-shaped images segments," *Proc. ACM Multimedia Conf.*, pp. 83–90, 1993.

[3] M. Gilge, T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform," *Signal Process., Image Commun.*, vol. 1, pp. 153–180, 1989.

[4] T. Sikora, S. Bauer, and B. Makai, "Efficiency of shape-adaptive 2-D transforms for coding of arbitrarily shaped image segments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 254–258, 1995.

[5] T. Sikora and B. Makai, “Shape-adaptive DCT for generic coding of video,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 59–62, 1995.

[6] R. Stasiński, “Alternative algorithms for computing discrete cosine and sine transforms,” *ISYNT’89*, Zagreb, pp. 74–77, 1989.

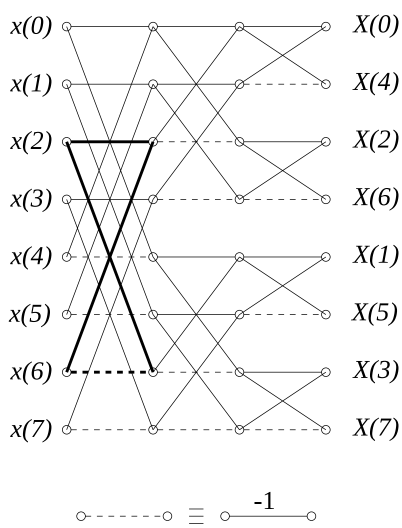


Figure 1: 8-point DWHT algorithm.

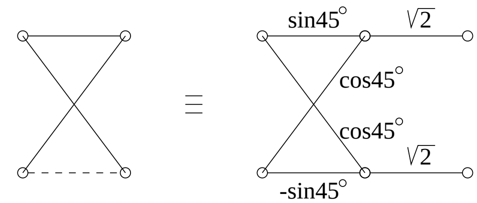
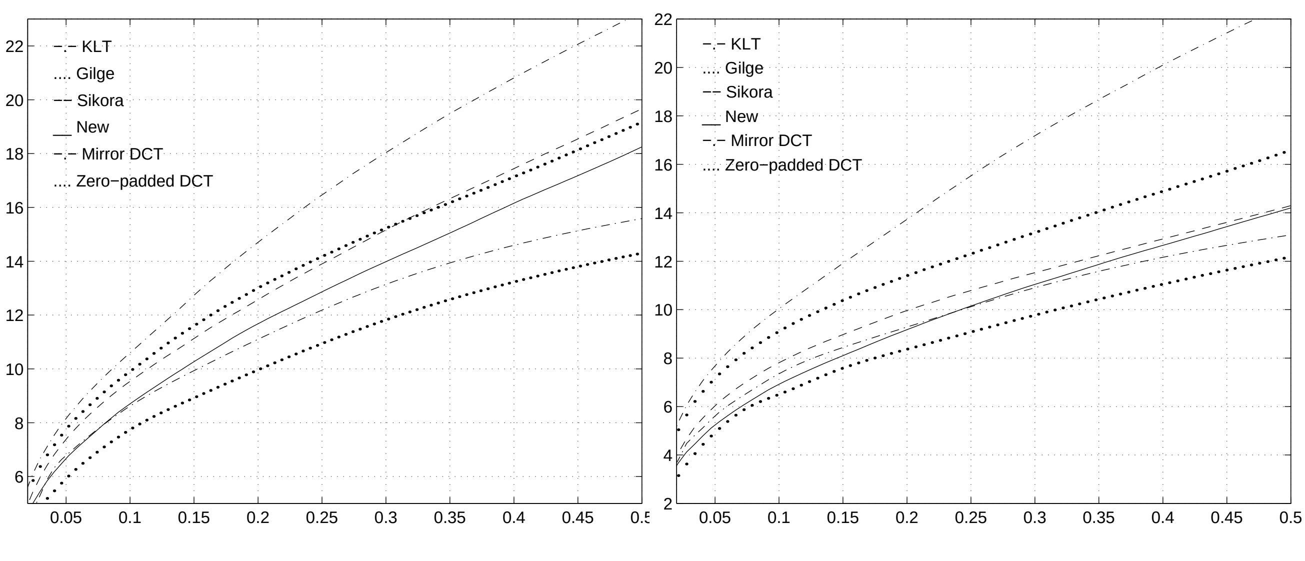


Figure 2: Standard multiplierless two-point butterfly and its representation by an orthonormal operation followed by connection multipliers.



(a)

(b)

Figure 3: Basis restriction error ε [dB] as a function of fraction p of highest-energy coefficients retained for 2-D Markov AR process with ρ=0.9 for (a) ellipse; and (b) atol.