Applied Mathematics and Computation 269 (2015) 731–743

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Evolving generation and fast algorithms of slantlet transform and slantlet-Walsh transform

Xiuqiao Xiang[a], Baochang Shi[b,*]

[a] School of Information Engineering, China University of Geosciences, Wuhan 430074, PR China
[b] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, PR China

ARTICLE INFO

Keywords:
Slantlet transform
Slantlet-Walsh transform
Mutation
Fast algorithm
Signal processing

ABSTRACT

The recently developed slantlet transform (SLT for short) is a wavelet with two zero moments, which can achieve a better balance between the time domain localization and smoothness. Due to its excellent characteristics, SLT is expected to be widely used in the information science fields such as compression and denoising of various signals. However, SLT without the in-place algorithm is not suitable for the parallel implementation, and the matrix formation and algorithm design of SLT lack universality, either. To address these issues, we present a new method for generating SLT recursively from the perspective of matrix rather than wavelet, by utilizing the basic techniques of bisection evolution, including so called Haar copy, Walsh mutation, slant mutation and slantlet mutation techniques. Meanwhile, we obtain a new orthogonal transform named as the slantlet-Walsh (SLW for short) transform, design the in-place fast algorithms of SLT and SLW transforms with the symmetry structure, which are very suitable for the parallel computing. In addition, the proposed scheme is clear and easy to be extended, we may take other copy means or select other mutation parameter value to generate new orthogonal matrices with required properties that may have special application value in some fields. As validated in our experiment, the proposed SLT is well-suited to deal with piecewise linear signal in comparison with transforms such as Haar, Walsh, Slant, Slant Haar, SLW transforms and DCT.

1. Introduction

The orthogonal transform and its fast algorithm play an important role in digital signal/image processing, cryptography and other fields. Commonly used orthogonal transforms include discrete cosine transform (DCT), Fourier transform, wavelet transform, Laplace transform, Walsh transform, etc. [1–5]. As a global transform, DCT cannot be effective on the recognition of local features. Fourier transform and the subsequent short-time Fourier transform may analyze the signal characteristics from the viewpoint of frequency domain, but the time frequency localization capability is limited. Because of the localization recognition ability and the multi-resolution features, wavelet analysis has a good performance in the signal processing. For many years, the researches on the theory and application of wavelet analysis have been continuously performed, and have still been unfolding currently.

Known from wavelet analysis and pyramidal decomposition theory [6–8], Haar function is the simplest one of all wavelets. Only two rows of Haar orthogonal matrix correspond to global functions, and the rest rows denote local functions. Walsh

* Corresponding author: Tel.: +86 027 87543231, +86 027 67883728; fax: +86 027 87543231
E-mail addresses: xiangxq0005@126.com (X. Xiang), shibc@hust.edu.cn (B. Shi).

http://dx.doi.org/10.1016/j.amc.2015.07.094
0096-3003/

function is Haar wavelet packet, i.e., the high frequency part with multi-resolution feature is further decomposed, and all rows of Walsh orthogonal matrix correspond to global functions. Accordingly, Haar type orthogonal matrices (including Her, Ter matrix, etc.) are their compromise [8,9]. It is these unique properties that different transform may have special application value in some fields [10,11]. For instance, Walsh transform with block wavelet has been widely used in DSP, image, video processing, communication in CDMA, spread spectrum, information safety, etc. [9–15]. It is appropriate to utilize an orthogonal transform containing slant base vector to represent the image with the gradual change of brightness, while slant base vector is the discrete sawtooth wave form falling within the scope of a homogeneous ladder [16]. Newly proposed slantlet transform (SLT) is a kind of orthogonal discrete wavelet transform with two zero moments. Compared with the ordinary wavelet transform, SLT can attain favorable balance between the time domain localization and smoothness [17]. In addition, SLT has a clear expression of wavelet basis function that owns piecewise linear characteristics and is very suitable for processing the signal with strong discontinuity and jump features. This excellent feature makes SLT be widely applied to the information science such as signal denoising, recognition, compression, digital watermark, etc. [18–22].

The SLT mentioned above has good property, nevertheless, SLT without the in-place algorithm is unfavorable to the parallel implementation, and the generation and algorithm design of SLT lack generality, either. To this end, we refer the bisection evolution idea [8,9,23,28], and propose a new recursive scheme to generate SLT by adopting Haar copy, Walsh mutation, slant mutation and slantlet mutation techniques. Using Walsh copy and mutation schemes, we produce a new orthogonal transform named as the slantlet-Walsh (SLW for short) transform in this paper. Meanwhile, we obtain the in-place fast algorithms of SLT and SLW transforms, and the experimental results indicate that SLT proposed in this paper owns better performance in processing piecewise linear signal compared with Haar, Walsh, slant, slant Haar, SLW transforms and DCT.

The rest of this paper is organized as follows. Section 2 presents the evolving generation and fast algorithms of Walsh matrix, slant matrix and slant Haar matrix. Then, the evolving generation and fast algorithms of SLT and SLW transforms are, respectively, proposed in Section 3 and Section 4, and the comparison experiments with previous transforms are performed in Section 5. Finally, conclusions are briefly drawn in Section 6.

2. Evolving generation and fast algorithms of three types of matrices

In this section, we simply review the recursive evolution and fast algorithms of Walsh matrix, slant matrix and slant Haar matrix mentioned in [8,23], which are closely related to our paper. Before this, we emphasize three important concepts, bisection evolution, copy and mutation, which are the basic techniques utilized in our paper. At the same time, we illustrate the meaning of symbols appeared in this paper to help us understand the subsequent contents.

Remark 1. Bisection evolution is an efficient downsizing calculation technology by which the calculation problem is recursively processed to the similar ones with halved size, finally, the required solution is directly obtained until the scale of the problem is sufficiently small. Copy means to make an imitation or reproduction of something so that it is exactly or almost exactly like the original existing thing, while mutation implies the act or process of changing partial elements into other different forms.

Remark 2. For Kronecker product $\otimes$, $n \times m$ matrices $A = (a_{ij})$, $B = (b_{ij})$, $i = 1, 2, \dots, n-1, n$, $j = 1, 2, \dots, m-1, m$, we have matrix $A \otimes B = (a_{ij}B)$ [8,9,23–28].

For example, let $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $B = (b_{ij})$, we get $A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$

Remark 3. $K \times K$ Walsh copy matrix F_K, Walsh mutation matrix H_K and slant mutation matrix X_K in this paper have the following forms:

$$F_K = \begin{pmatrix} I_{K/4} & & I_{K/4} & \\ & I_{K/4} & & -I_{K/4} \\ I_{K/4} & & -I_{K/4} & \\ & I_{K/4} & & I_{K/4} \end{pmatrix}, \quad X_K = \begin{pmatrix} I_{K/2} & & & & \\ & a_K & & b_K & \\ & & I_{K/4-1} & & \\ & -b_K & & a_K & \\ & & & & I_{K/4-1} \end{pmatrix}$$

$$H_K = \begin{pmatrix} 1 & & & & 1 & & & \\ & I_{K/4-1} & & & & & & \\ & & 1 & & & & -1 & \\ & & & I_{K/4-1} & & & & \\ 1 & & & & -1 & & & \\ & & & & & I_{K/4-1} & & \\ & & 1 & & & & 1 & \\ & & & & & & & I_{K/4-1} \end{pmatrix}. \tag{1}$$