Mathematical Problems of Computer Science 42, 113--120, 2014.

Fast Slant-Hadamard Transform Algorithm

Sonik R. Hakobyan

Yerevan State University
e-mail: sonahakobyan@ysu.am

Abstract

In this paper, we investigate an efficient algorithm for computation of parametric Slant-Hadamard transforms. We present the Slant-Hadamard matrix of order 2^n as a product of sparse matrices, develop the appropriate fast Slant-Hadamard transform and its complexity. In the end we present the detailed example of factoring of Slant-Hadamard transform matrix of order 8 and Matlab code for implementation Slant-Hadamard transform.

Keywords: Slant-Hadamard transform algorithm, Sparse matrix.

1. Introduction

A phenomenon characteristic of digital images is the presence of approximately constant or uniformly changing gray levels over a considerable distance or area. The slant transform is specifically defined for efficient representation of such images. It is a piecewise linear basis that follows the spirit of Walsh transform, possessing a discrete saw tooth-like basis vector which efficiently represents linear brightness variations along an image line. The slant transform has been used for signal compression and pattern recognition as well as for Intel's "indo" video compression algorithm [1-2]. The slant transform has the best compaction performance among the non-sinusoidal fast orthogonal transforms.

Historically, Enomoto and Shibata conceived the first, eight-point slant transform in 1971 andused it in TV image encoding. Its major innovation is given by the slant vector, which can properly follow the gradual changes in brightness of natural images. Pratt, Welch, and Chen have generalized it to any order 2^n and compared its performance with other transforms [2].

The slantlet transform has been successfully applied in compression and denoising. Currently, slant transforms are usually constructed via Hadamard or Haar transforms [5-7].

The most common fast slant transform methods [5] are based on the factorization of the slant transform matrix into a set of largely sparse matrices.

2. The Slant-Hadamard Transform

The Slant-Hadamard transform is defined as X = Sx [8], where S - the Slant-Hadamard transform matrix of order N is generated recursively by the following formula:

$$S(n)=\frac{1}{\sqrt{2}}\left|\begin{array}{cc:c:cc:c} 1 & 0 & O_{2\times p} & 1 & 0 & O_{2\times p} \\ a_n & b_n & & -a_n & b_n & \\ \hdashline O_{p\times 2} & & I_p & O_{p\times 2} & & I_p \\ \hdashline 0 & 1 & O_{2\times p} & 0 & -1 & O_{2\times p} \\ -b_n & a_n & & b_n & a_n & \\ \hdashline O_{p\times 2} & & I_p & O_{p\times 2} & & -I_p \end{array}\right| \times \begin{bmatrix} S(n-1) & O \\ O & S(n-1) \end{bmatrix}, \qquad (1)$$

where $p=\frac{N}{2}-2$, $N=2^n$, I_m denotes an identity matrix of order m, and the parameters a_n and b_n are given by

$$a_{n+1}=\left(\frac{3N^2}{4N^2-1}\right)^{1/2}, \quad b_{n+1}=\left(\frac{N^2-1}{4N^2-1}\right)^{1/2}, \qquad (2)$$

and $S(1)=\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ is just a Hadamard matrix of order 2 [8].

Example 2.1: Below we present the Slant-Hadamard transform matrices of order 4 and 8 obtained from (1) and (2)

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & -\frac{3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}, S(3)=\frac{1}{\sqrt{8}}\left|\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 5 & 3 & 1 & -1 & -3 & -5 & -7 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 7 & -1 & -9 & -17 & 17 & 9 & 1 & -7 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -3 & 3 & -1 & 1 & -3 & 3 & -1 \end{array}\right| \begin{array}{l} \\ \cdot 1/\sqrt{21} \\ \cdot 1/\sqrt{5} \\ \cdot 1/\sqrt{105} \\ \\ \cdot 1/\sqrt{5} \\ \cdot 1/\sqrt{5} \\ \\ \end{array}$$

3. Construction of Parametric Slant-Hadamard Transform

The forward and inverse parametric Slant-Hadamard transforms of order 2^n (n =1, 2, 3,...), are defined as $X=S_{2^n}x$, $x=S_{2^n}^T X$ [9-10], respectively, where x is an input data vector and S_{2^n} is generated recursively as

$$S_{2^n} = \frac{1}{\sqrt{2}} Q_{2^n} \begin{bmatrix} S_{2^{n-1}} & O_{2^{n-1}} \\ O_{2^{n-1}} & S_{2^{n-1}} \end{bmatrix} = \frac{1}{\sqrt{2}} Q_{2^n} \left(I_2 \otimes S_{2^{n-1}} \right),\ n > 1,$$

$$S_2 = H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} + & + \\ + & - \end{bmatrix},$$

where O_M denotes the zero matrix of order M, "+" corresponds to 1 and "-" to -1, $\otimes$ denotes the Kronecker product, and Q_{2^n} is the recursion kernel matrix defined as

$$Q_{2^n} = \left| \begin{array}{cc:c:cc:c} 1 & 0 & O_{2\times p} & 1 & 0 & O_{2\times p} \\ a_{2^n} & b_{2^n} & & -a_{2^n} & b_{2^n} & \\ \hdashline \multicolumn{2}{c:}{O_{p\times 2}} & I_p & \multicolumn{2}{c:}{O_{p\times 2}} & I_p \\ \hdashline 0 & 1 & O_{2\times p} & 0 & -1 & O_{2\times p} \\ -b_{2^n} & a_{2^n} & & b_{2^n} & a_{2^n} & \\ \hdashline \multicolumn{2}{c:}{O_{p\times 2}} & I_p & \multicolumn{2}{c:}{O_{p\times 2}} & I_p \end{array} \right|$$

As in [9-10], we introduce the expressions for a_{2^n} and b_{2^n} to construct the parametric Slant-Hadamard transforms matrices of order 2^n

$$a_{2^n} = \sqrt{\frac{3(2^{2n-2})}{4(2^{2n-2}) - \beta_{2^n}}},\ b_{2^n} = \sqrt{\frac{2^{2n-2} - \beta_{2^n}}{4(2^{2n-2}) - \beta_{2^n}}}, \tag{3}$$

where $-2^{2n-2} \le \beta_{2^n} \le 2^{2n-2}$, $a_2 = 1$.

It can be shown that for $\left|\beta_{2^n}\right| > 2^{2n-2}$ the Slant-Hadamard transforms matrices lose their orthogonality. The parametric slant transform matrices fulfill the requirements of the classical slant transform matrix. However, the parametric slant transform matrix is a parametric matrix with parameters $\beta_4,\ \beta_8, ...,\ \beta_{2^n}$.

- When $\beta_4 = \beta_8 = \cdots = \beta_{2^n} = \beta = 1$, we obtain a classical slant transform
- When $\beta_{2^n} = 2^{2n-2}$ for all β_{2^n}, $n \ge 2$, we obtain a Walsh-Hadamard transform.
- When $\beta_4 = \beta_8 = \cdots = \beta_{2^n} = \beta$, for $-4 \le \beta \le 4$, we obtain a constant beta slant transform.
- When $\beta_4 \ne \beta_8 \ne ... \ne \beta_{2^n}$, $-2^{2n-2} \le \beta_{2^n} \le 2^{2n-2}$, $n = 2, 3, 4, ...$, we refer to this case as a multiple betas slant transform [9-10]. In this case some of β_{2^n} can be equal but not all of them.

For n=2 and $\beta_4 = -4.0$ we have the following multiple betas slant transform matrix of order 4.

$$S(2)=\frac{1}{\sqrt{4}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{\sqrt{3}+\sqrt{2}}{\sqrt{5}} & \frac{\sqrt{3}-\sqrt{2}}{\sqrt{5}} & -\frac{\sqrt{3}-\sqrt{2}}{\sqrt{5}} & -\frac{\sqrt{3}+\sqrt{2}}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{\sqrt{3}-\sqrt{2}}{\sqrt{5}} & -\frac{\sqrt{3}+\sqrt{2}}{\sqrt{5}} & \frac{\sqrt{3}+\sqrt{2}}{\sqrt{5}} & -\frac{\sqrt{3}-\sqrt{2}}{\sqrt{5}} \end{bmatrix}.$$

Remark 3.1: *Parametric slant transform matrices fulfill the following requirements of classical slant transform:*

- *Its first row vector is of constant value.*
- *Its second row vector represents the parametric slant vector.*
- *Its basis vectors are orthonormal.*
- *It has a fast algorithm.*

Remark 3.2: *It is easy to verify that the parametric slant transform matrix can be represented as follows:*

$$S_{2^n}=\frac{1}{\sqrt{2^n}}\begin{cases} M_4 H_4, \text{ for } n=2, \\ C_{2^n} H_{2^n}, \text{ for } n>2, \end{cases}$$

where $C_{2^n}=M_{2^n}\otimes\begin{bmatrix} C_{2^{n-1}} & 0 \\ 0 & C_{2^{n-1}} \end{bmatrix}$,

$$M_{2^n}=\left|\begin{array}{cc:c:cc:c} 1 & 0 & O_{2\times p} & 0 & 0 & O_{2\times p} \\ 0 & b_{2^n} & & a_{2^n} & 0 & \\ \hdashline \multicolumn{2}{c:}{O_{p\times 2}} & I_p & \multicolumn{2}{c:}{O_{p\times 2}} & O_{p\times p} \\ \hdashline 0 & 0 & O_{2\times p} & 0 & 1 & O_{2\times p} \\ 0 & a_{2^n} & & -b_{2^n} & 0 & \\ \hdashline \multicolumn{2}{c:}{O_{p\times 2}} & O_{p\times p} & \multicolumn{2}{c:}{O_{p\times 2}} & I_p \end{array}\right| \qquad (4)$$

where $p=2^{n-1}-2$, $M_2=I_2$. $O_{m\times k}$ *denotes a zero matrix of size* $m\times k$, I_m *denotes an identity matrix of order m,* $\otimes$ *is the operator of Kronecker product,* H_{2^n} *is the Walsh-Hadamard matrix of order* 2^n*, and the parameters* a_{2^n} *and* b_{2^n} *are given in (3).*

4. Fast Slant-Hadamard Transform Algorithm

The parametric Slant-Hadamard transform matrix can be represented [11-12] as

$$S_{2^n} = \frac{1}{\sqrt{2^n}} M_{2^n} \left(H_2 \otimes I_{2^{n-1}}\right)\left(I_2 \otimes S_{2^{n-1}}\right), \quad n \geq 2,$$

where $M_2 = I_2$, M_{2^n} is given in (4).

The slant transform matrix of order 2^n can be factored [11-12] as

$$S_{2^n} = S_n S_{n-1} S_{n-2} \ldots S_1 = \prod_{i=1}^{n} S_i \text{, where } S_i = (I_{2^{n-i}} \otimes M_{2^i})(I_{2^{n-i}} \otimes H_2 \otimes I_{2^{i-1}}).$$

Below we represent in detail sparse matrices decomposition of Slant-Hadamard matrix of order 8.

Consider the Slant-Hadamard matrix $S_8 = \frac{1}{\sqrt{8}} S_3 S_2 S_1$ for $\beta_4 = \beta_8 = \beta = 1$. We have

$$S_1 = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{vmatrix}, \quad S_2 = \begin{vmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & 1 & 2 \end{vmatrix} \begin{matrix} \\ \cdot 1/\sqrt{5} \\ \\ \cdot 1/\sqrt{5} \\ \\ \cdot 1/\sqrt{5} \\ \\ \cdot 1/\sqrt{5} \end{matrix}$$

$$S_3 = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & \sqrt{5} & 0 & 0 & -4 & \sqrt{5} & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -\sqrt{5} & 4 & 0 & 0 & \sqrt{5} & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{vmatrix} \begin{matrix} \\ \cdot 1/\sqrt{21} \\ \\ \\ \\ \cdot 1/\sqrt{21} \\ \\ \\ \end{matrix}$$

Below the 8-point transform steps are given in detailed ($S_8 = S_3 S_2 S_1 x$, scaling coefficient $\frac{1}{\sqrt{8}}$ is omitted).

Step 1: $y = S_1 x$, (the number of additions is 8).

$$y = \begin{vmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{vmatrix} = S_1 x = \begin{vmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{vmatrix} \cdot \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{vmatrix} = \begin{vmatrix} x_0 + x_1 \\ x_0 - x_1 \\ x_2 + x_3 \\ x_2 - x_3 \\ x_4 + x_5 \\ x_4 - x_5 \\ x_6 + x_7 \\ x_6 - x_7 \end{vmatrix}$$

Step 2: $z = S_2 y$, (the number of additions is 12 and of multiplications is 8).

$$z=S_2y=\begin{bmatrix}1&0&1&0&0&0&0&0&\\2&1&-2&1&0&0&0&0&\cdot 1/\sqrt{5}\\0&1&0&-1&0&0&0&0&\\-1&2&1&2&0&0&0&0&\cdot 1/\sqrt{5}\\0&0&0&0&1&0&1&0&\\0&0&0&0&2&1&-2&1&\cdot 1/\sqrt{5}\\0&0&0&0&0&1&0&-1&\\0&0&0&0&-1&2&1&2&\cdot 1/\sqrt{5}\end{bmatrix}\cdot\begin{bmatrix}y_0\\y_1\\y_2\\y_3\\y_4\\y_5\\y_6\\y_7\end{bmatrix}=\begin{bmatrix}y_0+y_2\\(2(y_0-y_2)+(y_1+y_3))\cdot 1/\sqrt{5}\\y_1-y_3\\(-(y_0-y_2)+2(y_1+y_3))\cdot 1/\sqrt{5}\\y_4+y_6\\(2(y_4-y_6)+(y_5+y_7))\cdot 1/\sqrt{5}\\y_5-y_7\\(-(y_4-y_6)+2(y_5+y_7))\cdot 1/\sqrt{5}\end{bmatrix}.$$

Step 3: $X=S_3z,$ (the number of additions is 10 and of multiplications is 4).

$$X=S_3z=\begin{bmatrix}1&0&0&0&1&0&0&0&\\4&\sqrt{5}&0&0&-4&\sqrt{5}&0&0&\cdot 1/\sqrt{21}\\0&0&1&0&0&0&1&0&\\0&0&0&1&0&0&0&1&\\0&1&0&0&0&-1&0&0&\\-\sqrt{5}&4&0&0&\sqrt{5}&4&0&0&\cdot 1/\sqrt{21}\\0&0&1&0&0&0&-1&0&\\0&0&0&1&0&0&0&-1&\end{bmatrix}\cdot\begin{bmatrix}z_0\\z_1\\z_2\\z_3\\z_4\\z_5\\z_6\\z_7\end{bmatrix}=\begin{bmatrix}z_0+z_4\\4/\sqrt{21}(z_0-z_4)+\sqrt{5}/\sqrt{21}(z_1+z_5)\\z_2+z_6\\z_3+z_7\\z_1-z_5\\-\sqrt{5}/\sqrt{21}(z_0-z_4)+4/\sqrt{21}(z_1+z_5)\\z_2-z_6\\z_3-z_7\end{bmatrix}.$$

The total number of additions is 30 and the total number of multiplication is 12.

Now calculate the number of operations for S_{2^n} transform.

$$S_i=(I_{2^{n-i}}\otimes M_{2^i})(I_{2^{n-i}}\otimes H_2\otimes I_{2^{i-1}})=(I_{2^{n-i}}\otimes M_{2^i})\left(I_{2^{n-i}}\otimes\begin{pmatrix}I_{2^{i-1}}&I_{2^{i-1}}\\I_{2^{i-1}}&-I_{2^{i-1}}\end{pmatrix}\right).$$

The $\left(I_{2^{n-i}}\otimes\begin{pmatrix}I_{2^{i-1}}&I_{2^{i-1}}\\I_{2^{i-1}}&-I_{2^{i-1}}\end{pmatrix}\right)x$ transform requires only 2^n addition operations, and the $(I_{2^{n-i}}\otimes M_{2^i})y$ transform requires $6\cdot 2^{n-i}$ operations ($4\cdot 2^{n-i}$ multiplications and $2\cdot 2^{n-i}$ additions). Therefore, the S_i requires $2^{n+1-i}+2^n$ addition and $4\cdot 2^{n-i}$ multiplication operations.

So, the fast Slant-Hadamard transform of order 2^n requires $C_{2^n}^{+}=(n+1)2^n-2$ addition and $C_{2^n}^{\times}=2^{n+1}-4$ multiplication operations($n>2$), with $C_4^{+}=8$ and $C_4^{\times}=4$, where C_m^{+} and C_m^{x} are, respectively, the number of additions and multiplications required for the m – point fast algorithm [8].

```
% 1D FSHT Matlab code
function M=M2N(n)
if n==1
   M=eye(2);
else
p=2^(n-1)-2; m=2^(2*n-2);
a=sqrt(3*m/(4*m-1)) ; b=sqrt((m-1)/(4*m-1));
```

```
  M=[[1 0 ;0 b] zeros(2,p) [0 0;a 0] zeros(2,p);
       zeros(p,2) eye(p) zeros(p,2) zeros(p,p);
       [0 0 ; 0 a] zeros(2,p) [0 1 ;-b 0] zeros(2,p);
   zeros(p,2) zeros(p,p) zeros(p,2) eye(p)];
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Y=FSHT(n,x)
Y=x;
for  i=1:n
   Y=kron(eye(2^(n-i)),M2N(i))* kron(eye(2^(n-i)),kron(hadamard(2),eye(2^(i-1))))*Y;
end
end
```

The 2D Slant-Hadamard transform can be defined as SHT(f) = SH * f * SH', where f is the image and SH *and* SH' are Slant-Hadamard matrix and its transpose [9].

5. Cunclusion

In this paper, an efficient algorithm for computation of parametric Slant-Hadamard transforms is investigated. The Slant-Hadamard matrix of order 2^n is presented as a product of sparse matrices and the appropriate fast Slant-Hadamard transform and its complexity are developed. In the end the detailed example of factoring of Slant-Hadamard transform matrix of order 8 and Matlab code for implementation Slant-Hadamard transform on 1D case are presented. Later the Slant-Hadamard transformation will be discussed which will be used in image processing.

References

[1] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice Hall Inc, Englewood Cliffs, NJ 1989.

[2] M. M. Anguh and R. R. Martin, "A truncation method for computing slant transforms with applications to image processing", *IEEE Transactions on Communications*, vol. 43, no. 6, pp. 2103-2110, 1995.

[3] I. W. Selesnick, "The slantlet transform", *IEEE Transactions on Signal Processing,* vol. 47, no. 5, pp. 1304-1313, 1999.

[4] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*, New York, Springer-Verlag, 1975.

[5] J. F. Yang and C. P. Fang, "Centralized fast slant transform algorithms", *IEICE Transactions on Fundamentals of Electronics,Communications and Computer Sciences*, vol. E80A, no. 4, pp. 705-711, 1997.

[6] H. Sarukhanyan, S. Agaian, K. Egiazarian and J. Astola, "Reversible Hadamard Transforms", *Facta Universities (Nis), Series: Electronics and Energetics*, vol. 20, no. 3, pp. 309-330, 2007.

[7] S. Agaian, H. Sarukhanyan, K. Egiazarian and J. Astola, *Hadamard Transforms*, SPIE press, Bellingham, Washington, USA, 2011.

[8] S. Agaian, K. Tourshan and J. P. Noonan, "Parametric Slant-Hadamard transforms with applications", *IEEE Signal Processing Letters,* vol. 9, no. 11, pp. 375-377, 2002.

[9] S. Agaian, K. Tourshan and J. P. Noonan, “Parametric Slant-Hadamard transforms”, *IS&T/SPIE’s 15th Annual Symposium Electronic Imaging Science and Technology, Image processing: Algorithms and Systems II*, pp. 20-24 , Santan Clara, California, 2003.

[10] H. Sarukhanyan, “Hadamard matrices: construction methods and applications”, *In Proc. of The Workshop on Transforms and Filter Banks*, 35p., Tampere, Finland, 1998.

[11] H. Sarukhanyan, A. Anoyan, S. Agaian, K. Egiazarian and J. Astola, “Fast Hadamard transforms”, *Proc. of International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP'2001*, Pula, Croatia, pp. 33-40, 2001.

[12] S. Agaian, H .Sarukhanyan and K. Tourshan, “New classes of sequential slant hadamard Transform”, *Proc. Of International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP’02*, Toulouse, France, 4p., 2002.

[13] S. Minasyan, D. Guevorkian, S. Agaian and H. Sarukhanyan, ”On “Slant-Like” fast orthogonal transforms of arbitrary order”, *VIPromCom-2002, 4th EURASIP – IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications,* Zadar, Croatia, p. 6, 2002.

Submitted 02.08.2014, accepted 28.11.2014.

Սլանթ-Հադամարի ձևափոխության արագ ալգորիթմ

Ս. Հակոբյան

Ամփոփում

Ուսումնասիրվում է պարամետրով Սլանթ–Հադամարի ձևափոխության արդյունավետ ալգորիթմը: 2^n կարգի Սլանթ-Հադամարի մատրիցը ներկայացվում է նոսր մատրիցների արտադրյալի տեսքով, ինչպես նաև մշակվում են Սլանթ-Հադամարի արագ ձևափոխությունն ու նրա բարդությունը: Վերջում մանրամասն ներկայացվում են 8-րդ կարգի Սլանթ-Հադամարի ձևափոխության մատրիցի ֆակտորիզացիան և Սլանթ-Հադամարի ձևափոխությունը իրականացնող Matlab կոդը:

Алгоритм быстрого преобразования Сланта-Адамара

С. Акопян

Аннотация

В данной работе мы исследуем эффективный алгоритм параметрического преобразования Сланта-Адамара. Матрица Сланта-Адамара порядка 2^n представлена в виде произведения разреженных матриц. Разработаны соответствующее быстрое преобразование Сланта-Адамара и его сложность. Представлены подробный пример факторизации матрицы преобразования и Matlab код для реализации преобразования Сланта-Адамара .