Remigiusz KOŃCA

West Pomeranian University of Technology in Szczecin, Faculty of Computer Science and Information Technology

doi:10.15199/48.2015.02.10

Fast computation of the SLTF transform

***Abstract**. The paper presents computation process of the fast SLTF transform that use matrix-vector algebra. Examples explaining the course of the calculations both analysis and synthesis transform, are also illustrated by the graph-structural models that helps to understand the algorithm principle. Additionally, an improved calculation procedure reducing redundant data redirection was proposed.*

Streszczenie.** W pracy przedstawiono proces obliczania szybkiej transformaty SLTF z wykorzystaniem operacji wektorowo-macierzowych. Przykłady objaśniające przebieg obliczeń zarówno transformaty prostej jak i odwrotnej zilustrowano grafami ułatwiającymi zrozumienie zasady działania algorytmu. Dodatkowo zaproponowano ulepszoną procedurę obliczeniową redukującą nadmiarowe przeadresowania danych. **Proces obliczania szybkiej transformaty SLTF z wykorzystaniem operacji wektorowo-macierzowych

Keywords: SLTF Transform, fast algorithms.
Słowa kluczowe: Transformata SLTF, szybkie algorytmy.

Introduction

Progress in the field of Digital Signal Processing requires using more and more advanced mathematical methods to achieve complex tasks. SLTF Transform due to its unique properties is the subject of interest in many areas. It has already found application in chemical and biomedical studies [1-4].

In case of stationary signals analysis, good results can be achieved by obtaining the spectral form with the Fourier transform [5]. In the case of non-stationary signals, time-frequency analysis, that gives better results for this purpose, is desirable [6]. This kind of analysis is provided by a short-time Fourier transform STFT [7, 8] and a Gabor transform [9-11]. In case of STFT transform with sliding window, the window width remains constant. S transform [12-15], that is an extension of STFT transform and Wavelet transforms, introduces a variable width of Gaussian window, dependent on frequency with adjustable resolution in the time or frequency domain. Gabor transform, that is a special case of the STFT transform, where window in the time domain has a Gaussian shape, provides optimal localization in the field of time-frequency domain.

Presented by the Osama A. Ahmed, SLTF ("Stable Linear Time-Frequency") Transform [1], similar to the transforms presented above, allows time-frequency analyze. The same as the Gabor transform, it has optimal localization in a time-frequency domain, but in addition it also guarantees the stabilization and localization of both the window and its bio-orthogonal functions.

The purpose of this paper is to introduce an improved calculation procedure and analyze the specific examples that are illustrating the principle of operations and rapid technique for SLTF transform calculation, visualized by the graphs.

SLTF transform definition

The SLFT transform is defined for a finite discrete signals $x(k)$, where $0 \le k < K$ and $K = M N$ is the length of the signal and the parameters M, N are the number of samples in time and frequency. Analysis transform is given by the formula:

$$a_{m,n} = \sum_{k=0}^{K-1} x(k)\gamma_m^*(k)\, csin\frac{\pi(k+0.5)(n+0.5)}{N},$$

for $m = \overline{1, M-1}$ and $n = \overline{1, N-1}$.

Synthesis transform can be described as:

$$x(k) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} a_{m,n} h_m(k)\, csin\frac{\pi(k+0.5)(n+0.5)}{N}.$$

Elements $a_{m,n}$ are the SLTF transform coefficients. Function $csin(x)$ and $h_m(k)$ are defined as:

$$csin(x) = \begin{cases} sin(x), \text{ when } m \text{ is odd,} \\ cos(x), \text{ when } m \text{ is even,} \end{cases}$$

$$h_m(k) = \delta^{-\frac{1}{2}} e^{-\frac{\pi}{2\delta^2}\left(k - mN - (N-1)/2\right)^2}.$$

$h_m(k)$ is in this case the normalized Gaussian window, shifted to the center of the m^{th} window with δ controlling the window width. $\gamma_m(k) = \gamma_m(k-mN)$ is a bio-orthogonal function to $h(k)$, which means, that the following condition is meet:

$$\sum_{k=0}^{K-1} h_m(k)\gamma(k)\, csin\frac{\pi(k+0.5)(n+0.5)}{N} = \delta_n \delta_m.$$

Fast SLTF transform calculation

Nuances of the SLTF transform calculation in the existing studies have not been adequately disclosed. Additionally the computational procedure can be improved, namely permutations of the elements in the input and output are executed twice. It turns out that if the computations are performed exactly at is described in the paper [1], perceptible time investments are required. To simplify the notation and to reduce redundant steps, that in this case lead to additional redirection of data during the calculation process, additional notation is introduced:

$$\mathbf{P}_K^{(5)} = (\mathbf{P}_K^{(1)})^{\mathrm{T}} (\mathbf{P}_K^{(4)})^{\mathrm{T}}$$

and

$$\mathbf{P}_K^{(6)} = \mathbf{P}_K^{(4)} \mathbf{P}_K^{(2)}.$$

The whole calculation process of the analysis transform can be presented by matrix-vector algebra as follows:

$$\mathbf{Y}_{K\times 1} = \mathbf{E}_K \mathbf{G}_K^{-1} \mathbf{X}_{K\times 1} = \mathbf{E}_K \mathbf{P}_K^{(5)} (\mathbf{E}_K^{(1)})^{\mathrm{T}} \mathbf{P}_K^{(4)} \times$$

$$\text{(1)} \quad \times diag\{(\mathbf{B}_2^{(0,0)})^{-1}, \ldots, (\mathbf{B}_2^{(0,M/2-1)})^{-1}, (\mathbf{B}_2^{(1,0)})^{-1},$$

$$\ldots, (\mathbf{B}_2^{(N-1,M/2-1)})^{-1}\} (\mathbf{P}_K^{(4)})^{\mathrm{T}} \mathbf{E}_K^{(1)} \mathbf{P}_K^{(6)} \mathbf{X}_{K\times 1}.$$

Similarly to the analysis transform, inverse transform can be described as:

$$\mathbf{X}_{K\times1}=\mathbf{G}_K\mathbf{E}_K\mathbf{Y}_{K\times1}=$$

(2) $$=\mathbf{P}_K^{(5)}\mathbf{E}_K^{(1)}\mathbf{P}_K^{(4)}diag\{\mathbf{B}_2^{(0,0)},\ldots,\mathbf{B}_2^{(0,M/2-1)},\mathbf{B}_2^{(1,0)},$$

$$\ldots,\mathbf{B}_2^{(N-1,M/2-1)}\}(\mathbf{P}_K^{(4)})^{\mathrm{T}}(\mathbf{E}_K^{(1)})^{\mathrm{T}}\mathbf{P}_K^{(6)}\mathbf{E}_K\mathbf{Y}_{K\times1}.$$

Matrices that are components of the formulas (1) and (2) are of the following form:

$$\mathbf{G}_K=\begin{bmatrix}\mathbf{G}_N^{(0)} & \mathbf{G}_N^{(M-1)} & \ldots & \mathbf{G}_N^{(2)} & \mathbf{G}_N^{(1)}\\ \mathbf{G}_N^{(1)}\mathbf{J}_N & \mathbf{G}_N^{(0)}\mathbf{J}_N & \ldots & \mathbf{G}_N^{(3)}\mathbf{J}_N & \mathbf{G}_N^{(2)}\mathbf{J}_N\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \mathbf{G}_N^{(M-2)} & \mathbf{G}_N^{(M-3)} & \ldots & \mathbf{G}_N^{(0)} & \mathbf{G}_N^{(M-1)}\\ \mathbf{G}_N^{(M-1)}\mathbf{J}_N & \mathbf{G}_N^{(M-2)}\mathbf{J}_N & \ldots & \mathbf{G}_N^{(1)}\mathbf{J}_N & \mathbf{G}_N^{(0)}\mathbf{J}_N\end{bmatrix}.$$

$\mathbf{G}_K^{(m)}=(-1)^{\lfloor(m+1)/2\rfloor}diag\{h_m(0),\ldots,h_m(N-1)\}$ are the diagonal matrices, with the elements $h_m(k)=\delta^{-\frac{1}{2}}e^{-\frac{\pi}{2\delta^2}(k-mN-(N-1)/2)^2}$ on the diagonal, where $\lfloor\bullet\rfloor$ is the integer part of $\bullet$. $\mathbf{J}_N$ is a rows exchange matrix defined as follows:

$$\mathbf{J}_N=\begin{bmatrix}0 & 0 & \ldots & 0 & 1\\ 0 & 0 & \ldots & 1 & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 1 & \ldots & 0 & 0\\ 1 & 0 & \ldots & 0 & 0\end{bmatrix}.$$

$\mathbf{E}_K$ is a block-diagonal matrix with the matrices that represents N-point DCT-IV and DST-IV on the diagonal, arranged in the following manner:

$$\mathbf{E}_K=diag\{\underbrace{\mathbf{C}_N,\mathbf{S}_N,-\mathbf{C}_N,-\mathbf{S}_N,\ldots,\mathbf{C}_N,\mathbf{S}_N}_{M}\}.$$

$\mathbf{C}_N$ is a matrix of the N-point DCT-IV transform:

$$\mathbf{C}_N=[c_{n,k}]=\left[\sqrt{\frac{2}{N}}cos\frac{\pi(n+0.5)(k+0.5)}{N}\right].$$

$\mathbf{S}_N$ is a matrix of the N-point DST-IV transform:

$$\mathbf{S}_N=[s_{n,k}]=\left[\sqrt{\frac{2}{N}}sin\frac{\pi(n+0.5)(k+0.5)}{N}\right].$$

$\mathbf{P}_K^{(1)},\mathbf{P}_K^{(2)},\mathbf{P}_M^{(3)},\mathbf{P}_K^{(4)},\mathbf{P}_K^{(5)},\mathbf{P}_K^{(6)}$ are the permutation matrices. Such a matrix can be represented by the coding vector p, and element $p(k)$, is the index of the column, where "1" is located in the w k raw of the matrix $\mathbf{P}_K^{(i)}$, for $i=\overline{1,6}$. Permutation matrices are orthogonal, so $(\mathbf{P}_K^{(i)})^{-1}=(\mathbf{P}_K^{(i)})^{\mathrm{T}}$. $\mathbf{P}_K^{(i)}$ used in the formula (1) and (2) are represented by the following coding vectors:

$$p_1(k)=\lfloor k/M\rfloor+N(k\,mod\,M),$$

for $k=\overline{0,K-1}$,

$$p_2(k)=l(N-1)+(-1)^l\lfloor k/M\rfloor+N(k\,mod\,M),$$

where $l=k\,mod\,2$,

$$p_3(k)=\lfloor m/(M/2)\rfloor+2(m\,mod\,M/2),$$

for $m=\overline{0,M-1}$,

$$\mathbf{P}_K^{(4)}=diag\{\underbrace{\mathbf{P}_M^{(3)},\ldots,\mathbf{P}_M^{(3)}}_{N}\}.$$

$\mathbf{E}_K^{(1)}$ is a block-diagonal matrix with the elements $\mathbf{E}_{M/2}^{(2)}=[e_{m,k}]$ on the diagonal. Therefore it takes a form:

$$\mathbf{E}_K^{(1)}=diag\{\underbrace{\mathbf{E}_{M/2}^{(2)},\mathbf{E}_{M/2}^{(2)},\ldots,\mathbf{E}_{M/2}^{(2)}}_{2N}\}.$$

Each of the elements $\mathbf{E}_{M/2}^{(2)}$ is the $M/2$-point Discrete Fourier Transform represented as the:

$$e_{m,k}=\frac{1}{\sqrt{M/2}}e^{-\frac{i2\pi mk}{M/2}}.$$

Matrices $\mathbf{B}_2^{(n,m)}$ of the dimension 2×2 are formed according to the formula:

$$\begin{bmatrix}\mathbf{B}_2^{(n,0)}\\ \mathbf{B}_2^{(n,1)}\\ \vdots\\ \mathbf{B}_2^{(n,M/2-1)}\end{bmatrix}=\frac{M}{2}(\mathbf{P}^{(3)})^{\mathrm{T}}\begin{bmatrix}\mathbf{E}_{M/2}^{(2)} & \\ & \mathbf{E}_{M/2}^{(2)}\end{bmatrix}\mathbf{P}^{(3)}\begin{bmatrix}\mathbf{D}_2^{(n,0)}\\ \mathbf{D}_2^{(n,1)}\\ \vdots\\ \mathbf{D}_2^{(n,M/2-1)}\end{bmatrix},$$

$$\mathbf{D}_2^{(n,m)}=\begin{bmatrix}h_{2m}(n) & h_{(2m-1)\,mod\,M}(n)\\ -h_{2m+1}(n1) & h_{2m}(n1)\end{bmatrix},$$

for $m=\overline{0,M/2-1}$, $n=\overline{0,N-1}$, $n1=N-1-n$.

Inverse matrices $(\mathbf{B}_2^{(n,m)})^{-1}$ of the dimension 2×2 can be calculated directly based on the dependency:

$$\begin{bmatrix}a & b\\ c & d\end{bmatrix}^{-1}=\frac{1}{ad-bc}\begin{bmatrix}d & -b\\ -c & a\end{bmatrix}.$$

Fast SLTF transform example

In this section we will show the example of the SLTF transform calculation process.

Let N=M=4. Then K=16.

Procedure for the SLTF analysis transforms, described by the formula (1) for the M=4, N=4 takes the form:

$$\mathbf{Y}_{16\times1}=\mathbf{E}_{16}\mathbf{G}_{16}^{-1}\mathbf{X}_{16\times1}=$$

$$=\mathbf{E}_{16}\mathbf{P}_{16}^{(5)}(\mathbf{E}_{16}^{(1)})^{\mathrm{T}}\mathbf{P}_{16}^{(4)}diag\{(\mathbf{B}_2^{(0,0)})^{-1},(\mathbf{B}_2^{(0,1)})^{-1},$$

$$\ldots,(\mathbf{B}_2^{(3,0)})^{-1},(\mathbf{B}_2^{(3,1)})^{-1}\}(\mathbf{P}_{16}^{(4)})^{\mathrm{T}}\mathbf{E}_{16}^{(1)}\mathbf{P}_{16}^{(6)}\mathbf{X}_{16\times1}.$$

The $\mathbf{E}_{16}^{(1)}$ matrix on the diagonal contains the 2-point Discrete Fourier Transform matrices

$$\mathbf{E}_2^{(2)}=\frac{1}{\sqrt{2}}\mathbf{H}_2,$$

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Inverse matrices $(\mathbf{B}_2^{(n,m)})^{-1}$ are made by applying the scheme (2) for the $\mathbf{B}_2^{(n,m)}$ matrices. This gives the matrices of the form:

$$(\mathbf{B}_2^{(n,m)})^{-1} = \begin{bmatrix} b_{0,0}^{n,m} & b_{1,0}^{n,m} \\ b_{0,1}^{n,m} & b_{1,1}^{n,m} \end{bmatrix},$$

for $n = \overline{0,3}$, $m = \overline{0,1}$.

$$\mathbf{E}_{16} = diag\{\mathbf{C}_4, \mathbf{S}_4, -\mathbf{C}_4, -\mathbf{S}_4\}$$

Matrices $\mathbf{C}_4$ and $\mathbf{S}_4$ symbolize 4-point DCT-IV and DST-IV transforms matrices.

The graph-structural model representing calculation process for SLTF analysis transform is shown in Figure 1. Graf is oriented from left to right. The straight lines represent data transfer channels.

Similar as in case of analysis transform, the procedure for computing inverse transform (2) for M=4, N=4 can also be described as a matrix product:

$$\mathbf{X}_{16\times 1} = \mathbf{G}_{16}\mathbf{E}_{16}\mathbf{Y}_{16\times 1} = \mathbf{P}_{16}^{(5)}\mathbf{E}_{16}^{(1)}\mathbf{P}_{16}^{(4)} diag\{\mathbf{B}_2^{(0,0)},$$

$$\mathbf{B}_2^{(0,1)}, \ldots, \mathbf{B}_2^{(3,0)}, \mathbf{B}_2^{(3,1)}\}(\mathbf{P}_{16}^{(4)})^{\mathrm{T}}(\mathbf{E}_{16}^{(1)})^{\mathrm{T}}\mathbf{P}_{16}^{(6)}\mathbf{E}_{16}\mathbf{Y}_{16\times 1}.$$

The graph-structural model showing the inverse transform calculation procedure is presented in Figure 2.

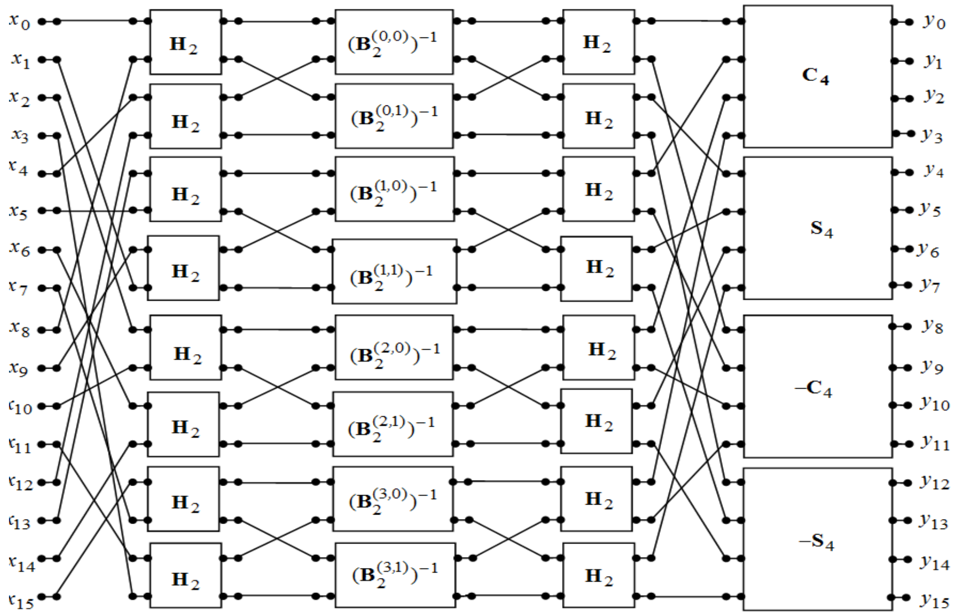

Fig. 1. The graph-structural model of calculation process for the SLTF analysis transform for N=M=4.

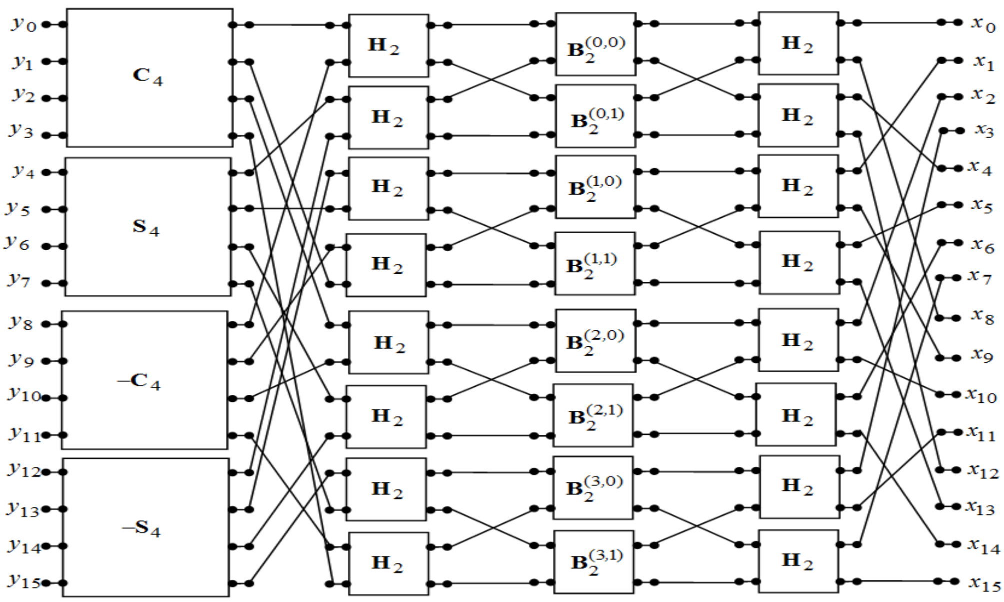

Fig.2. The graph-structural model of calculation process for the SLTF inverse transform for N=M=4.

Summary

In this paper calculation process of the fast algorithm for the SLTF transform was presented. The procedure requires calculation of:

- M times N-point DCT-IV or DST-IV transform,
- $2N$ times $M/2$-point FFT transform,
- $2N$ times $M/2$-point inverse FFT transform,
- $NM/2$ multiplications by the $(\mathbf{B}_2^{(i,j)})^{-1}$ or $\mathbf{B}_2^{(i,j)}$ matrix of the dimension 2×2.

Therefore to calculate analysis or synthesis transform $K(log_2K - 0{,}5\cdot log_2N)$ multiplications and $K(2log_2K - log_2N - 3)$ additions are required. Procedure presented in this paper rationalize the number of data permutations by removing redundant steps during the calculation of the matrix $\mathbf{G}_K$ and $\mathbf{G}_K^{-1}$. Improved procedure removes two of data permutations of the vector with length K. Graph-structural models of the SLTF analysis and synthesis transforms for $N=M=4$ can be generalized to any dimension.

Acknowledgments

I would like to thank Professor Aleksandr Cariow for his support and invaluable ideas in the field of the calculation improvements and advices on organization of the paper.

REFERENCES

[1] Ahmed O.A., Fast Computation of Discrete SLTF Transform, *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, (2001), 317-320

[2] Ahmed O.A , Fahry M.M., NMR Signal Enhancement Via a New Time-Frequency Transform, *IEEE TRANSACTIONS ON MEDICAL IMAGING*, vol. 20 (2001), no. 10, 1018-1025

[3] Ahmed O.A., New denoising scheme for magnetic resonance spectroscopy signals, *IEEE TRANSACTIONS ON MEDICAL IMAGING*, vol. 25 (2005), no. 6, 804-813

[4] Ahmed O.A., New denoising scheme for magnetic resonance spectroscopy signals, *Proceedings - 23rd Annual Conference IEEE/EMBS Oct.25-28 2001 Istanbul TURKEY*, vol.4 (2001), 2157- 2160

[5] Blahut R.E., Fast Algorithms for Signal Processing, *Cambridge University Press*, (2010), 68-112

[6] Andreatto B., Tariov A., A fast algorithm for multiresolution discrete Fourier transform, *Electrical Review* (2012), no. 11a/2012, 66-69

[7] Baba T., Time-Frequency Analysis Using Short Time Fourier Transform , *The Open Acoustics Journal,* 5 (2012), 32-38

[8] Nawab S.H., Quatieri T.F., Lim J.S., Signal reconstruction from short-time Fourier transform magnitude, *Acoustics, Speech and Signal Processing, IEEE Transactions,* vol 31(1983), no. 4, 986-998

[9] Balart R., Matrix reformulation of the Gabor transform, *Optical Engineering,* vol 31(1992), no 6, 1235-1242

[10] Orr R. S., The Order of Computation for Finite Discrete Gabor Transforms, *Signal Processing, IEEE Transactions,* vol. 41(1993), no. 1, 122-130

[11] Stewart D. F., Potter L. C., Ahalt S. C., Computationally attractive real Gabor transforms, *Signal Processing, IEEE Transactions,* vol. 43(1995), no. 1, 122-130

[12] Stockwell R.G., Mansinha L., Lowe R.P., Localization of complex spectrum: the S transform, *IEEE Transactions on Signal Processing*, 144 (1996), 998-1001

[13] Chu P.C., The S-transform for obtaining localized spectra, *Marine Technological Society Journal*, 29 (1996), 28- 38

[14] Pinnegar P.C., Mansinha L., The S-transform with windows of arbitrary and varying shape, *Geophysics*, vol. 69 (2003), no. 1, 381-385

[15] Tariov A., Gliszczyński M., Vectorized S Transform algorithms for multiprocessor platform, *PAK* (2011), no. 11, 1401-1403

***Authors**: Remigiusz Końca, West Pomeranian University of Technology in Szczecin, Faculty of Computer Science and Information Technology, Żołnierska 49, 71-210 Szczecin, E-mail: rkonca@wi.zut.edu.pl;*