고속 리버스 자켓 변환과 그의 역변환

정회원 이 승 래*, 성 굉 모*

Fast Reverse Jacket Transform and Its Inverse Transform

Seung-Rae Lee*, Koeng-Mo Sung* *Regular Members*

요 약

본 논문에서는 고속 리버스 자켓 역변환 (inverse fast Reverse Jacket transform, 간략히 IFRJT)을 제안하며 이 방법은 역변환을 explicit하게 표현한다. 이 알고리즘의 장점은 중앙가중치 하다마드 변환보다 더 빠르고 쉽게 주어진 행렬의 역을 구한다는 점이다. 우리는 얼마나 간단히 IFRJT를 얻을 수 있는지를 예제를 통해 보여준다.

ABSTRACT

In this paper, we propose an inverse fast Reverse Jacket transform (IFRJT) and express the inverse in explicit form. The advantage of this algorithm is to get the inverse of a given matrix more easier and faster than using the center-weighted Hadamard transform (CWHT). We illustrate how simple the IFRJT can be achieved and provide an example.

Ⅰ. 서 론

일반적으로 주어진 정칙행렬의 역을 구하는 효율적 알고리즘을 찾는 것은 그리 쉬운일이 아니다. 리버스 자켓 행렬의 특이한 현상은 그의 역행렬을 구할 때 본 행렬 (original matrix)의 요소들이 그룹을 지어 위치들이 이동한 형상으로 나타나 마치 등산복을 뒤집어 입은 것처럼 보인다 하여 그의 이름을 Reverse Jacket 행렬이라 칭하였다[4]. 이 행렬의 장점중에 하나는 그의 역행렬이 본 행렬과 비교할 때 동일한 구조를 갖고 있으며 쉽게 구한다는 것이다[4-6]. 리버스 자켓 행렬을 분해 (decomposition)하여 개발한 고속 리버스 자켓 변환 (fast Reverse Jacket transform, 간략히 FRJT)은 중앙가중치 하다마드 변환 (Center-weighted Hadamard transform, 간략히 CWHT)[2] 보다 훨씬 더 빠른 알고리즘이다[5].

본 논문에서는 고속 리버스 자켓 역변환을 Kronecker product나 direct sum 연산자등만을 사용하여 explcit하게 표현한다. 이 역변환 역시 CWHT의 역변환보다 더 효율적이다.

본 논문은 다음과 같이 구성되었다. 제 2장에서는 개선된 리버스 자켓 행렬의 정의와 역행렬을 구하는 문제를 해결한 정리 그리고 고속 푸리에 변환과의 관계를 나타내는 정리등을 학습한다. 제3장에서는 이 행렬을 분해하여 얻은 고속 리버스 자켓 변환과 그의 역변환을 소개하며 제4장에서는 IFRJT와 ICWTH의 효율성을 비교한다. 끝으로, 제5장에서는 결론과 향후 전망에 대하여 논의한다.

Ⅱ. 리버스 자켓 행렬과 예비학습

이 장에서는 역행렬이 뒤집어지는 기하학적 구조를 갖는 리버스 자켓 행렬을 구성한다. 이 행렬은 하다마드 행렬의 요소들을 일반화한다. 안팎으로 뒤집어 입을 수 있는 등산복처럼 리버스 자켓의 역행렬을 구할 때 적어도 두 개의 행렬요소의 위치가 바뀐다. 예를 들어, 중앙 원으로부터 바깥쪽으로, 그리고 바깥쪽에서 안쪽으로 뒤바뀌는 형태가 된다. 이러한 기이한 현상으로 말미암아 이 행렬을 리버스 자켓 행렬이라 칭하였다[4].

* 서울대학교 전기공학부 (srlee@acoustics.snu.ac.kr)
논문번호: K01019-0109, 접수일자: 2000년 1월 9일

주어진 2×2 기본행렬 $R_2 \in \square^{2\times 2}$에 대하여 가역 (invertible)인 4×4 행렬을 구성한다.

I_{2^k}와 O_{2^k}, $j \in \square \cup 0$은 각각 $(2^j \times 2^j)$ 단위행렬과 영행렬이며, $\square$, $\square$은 각각 실수와 자연수의 집합을 칭한다.

정의 1: [5] a, b, c, d를 영이 아닌 실수 또는 복소수라 하자. 리버스 자켓 행렬은 귀납적으로 (recursively) 다음과 같이 정의된다.

$$R_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} S_{2^k} \\ S_{2^k} R_{2^k} Z_{2^k} & -J_{2^k} R_{2^k} J_{2^k} \end{bmatrix} \tag{2.1}$$

여기서 초기조건은 $1 \le k$에 대하여

$$R_2 := \begin{bmatrix} a & b \\ c & -d \end{bmatrix}, Z_{2^k} := \begin{bmatrix} I_{2^{k-1}} & O_{2^{k-1}} \\ O_{2^{k-1}} & -I_{2^{k-1}} \end{bmatrix},$$

$$S_{2^k} := \begin{bmatrix} O_{2^{k-1}} & I_{2^{k-1}} \\ I_{2^{k-1}} & O_{2^{k-1}} \end{bmatrix}, \text{ 그리고 } J_{2^k} := \begin{bmatrix} O_{2^{k-1}} & I_{2^{k-1}} \\ -I_{2^{k-1}} & O_{2^{k-1}} \end{bmatrix}$$

이다. (2.2)

정리 2: [5] R_2가 정칙행렬이라 하자. 리버스 자켓 행렬

$$R_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} S_{2^k} \\ S_{2^k} R_{2^k} Z_{2^k} & -J_{2^k} R_{2^k} J_{2^k} \end{bmatrix} \tag{2.3}$$

이면 $R_{2^{k+1}}$는 정칙행렬이며

$$R_{2^{k+1}}^{-1} = \begin{bmatrix} \check{R}_{2^k}^{-1} & Z_{2^k} \check{R}_{2^k}^{-1} S_{2^k} \\ S_{2^k} \check{R}_{2^k}^{-1} Z_{2^k} & -J_{2^k} \check{R}_{2^k}^{-1} J_{2^k} \end{bmatrix} \tag{2.4}$$

이다. 이 때,

$$\check{R}_{2^k} = R_{2^k} + Z_{2^k} R_{2^k} S_{2^k} J_{2^k} R_{2^k}^{-1} J_{2^k} S_{2^k} R_{2^k} Z_{2^k}, \ 1 \le k$$

RJM과 FFT 행렬과의 관계는 다음의 요약에서 알 수 있다.

행렬 U_{2k} $(2 \le k)$를 다음과 같이 정의한다.

$$U_{2^k} := [I_{2^{k-1}} \oplus (S_2 \otimes I_{2^{k-2}})]$$

여기서 $S_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$이며 $\oplus$direct sum 연산자이다. R_2가 정칙행렬이라는 가정아래 행렬

$$W_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} \\ R_{2^k} Z_{2^k} & Z_{2^k} R_{2^k} Z_{2^k} \end{bmatrix} \tag{2.5}$$

를 정의하면 $W_{2^{k+1}}$는 정칙행렬이며

$$W_{2^{k+1}}^{-1} = \begin{bmatrix} \hat{R}_{2^k}^{-1} & Z_{2^k} \hat{R}_{2^k}^{-1} \\ \hat{R}_{2^k}^{-1} Z_{2^k} & Z_{2^k} \hat{R}_{2^k}^{-1} Z_{2^k} \end{bmatrix} \tag{2.6}$$

이다. 이 때,

$$\hat{R}_{2^k} = R_{2^k} - Z_{2^k} R_{2^k} Z_{2^k} R_{2^k}^{-1} Z_{2^k} R_{2^k} Z_{2^k}, \ 1 \le k$$

더 나아가서,

$$R_{2^{k+1}} = U_{2^{k+1}} W_{2^{k+1}} U_{2^{k+1}} \tag{2.7}$$

가 성립하는 직교행렬 $U_{2^{k+1}}$ $(1 \le k)$이 존재한다.

Ⅲ. 고속 리버스 자켓 변환과 그의 역변환

이 장에서는 고속 하다마드 변환이나 고속 중앙가중치 하다마드 변환을 일반화한 알고리즘을 소개하기로 한다[5]. 이 알고리즘을 위해 우선 두 개의 치환행렬을 정의하자.

$$P_4 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$P_{2^k} := P_4 \otimes I_{2^{k-2}}, \ 2 \le k,$$

$$Q_{2^k} := [I_{2^{k-1}} \oplus (S_2 \otimes I_{2^{k-2}})] \tag{3.1}$$

여기서 $\otimes$는 Kronecker product를 $\oplus$는 direct sum을 각각 나타낸다.

N차원 벡터 x에 관한 리버스 자켓 변환 y를 다음 관계식으로 표현한다.

$$y = R_N \cdot x \tag{3.2}$$

이 때, R_N은 N×N리버스 자켓 행렬이다. 한편 그의 역변환은

$$x = R_N^{-1} \cdot y \tag{3.3}$$

$\otimes$와 $\oplus$등을 이용한 행렬분해를 통해서 개발한

고속 리버스 자켓 변환 (FRJT) 알고리즘은 다음과 같은 수식으로 표현된다.

$$R_{2^k} = P_{2^k}^T \underbrace{(H_{2^{k-1}} \oplus H_{2^{k-1}})}_{\text{스테이지 3}} \underbrace{((U_A \otimes I_{2^{k-2}}) \oplus (U_B \otimes I_{2^{k-2}}))}_{\text{스테이지 2}} \underbrace{(H_2 \otimes I_{2^{k-1}})}_{\text{스테이지 1}} Q_{2^k}^T \tag{3.4}$$

여기서 $U_A := \mathrm{diag}(a, b)$ 이고 $U_B := \mathrm{diag}(c, d)$ 이다.

[5]에서 밝혔듯 a=b=c=1일 때 $N \log_2 N$ 덧셈과 $\frac{N}{4}$의 곱셈이 소요된다.

이제 고속 리버스 자켓 변환의 역변환 (Inverse of fast Reverse Jacket Transform, 간략히 IFRJT)을 잘 알려진 Kronecker product와 direct sum 연산자 계산 공식들을 이용하여 쉽게 유도할 수 있다.

$$\begin{aligned}
R_{2^k}^{-1} &= Q_{2^k}(H_2 \otimes I_{2^{k-1}})^{-1}((U_A \otimes I_{2^{k-2}}) \oplus (U_B \otimes I_{2^{k-2}}))^{-1}(H_{2^{k-1}} \oplus H_{2^{k-1}})^{-1} P_{2^k} \\
&= Q_{2^k}(\frac{1}{2^{k-1}}(H_2 \otimes I_{2^{k-1}}))((U_A \oplus U_B)^{-1} \otimes I_{2^{k-2}})(\frac{1}{2^{k-1}} H_{2^{k-1}} \oplus \frac{1}{2^{k-1}} H_{2^{k-1}}) P_{2^k} \\
&= Q_{2^k}(\frac{1}{2^{k-1}}(H_2 \otimes I_{2^{k-1}}))((U_A^{-1} \oplus U_B^{-1}) \otimes I_{2^{k-2}})(\frac{1}{2^{k-1}}(H_{2^{k-1}} \oplus H_{2^{k-1}})) P_{2^k} \\
&= \frac{1}{N} Q_{2^k}(H_2 \otimes I_{2^{k-1}})((U_A^{-1} \oplus U_B^{-1}) \otimes I_{2^{k-2}})(H_{2^{k-1}} \oplus H_{2^{k-1}}) P_{2^k}
\end{aligned} \tag{3.5}$$

예제 : 기본행렬을 $R_2 = \begin{bmatrix} 2 & -1 \\ 1 & -4 \end{bmatrix}$라 하자.

N=4일 경우에 IFRJT를 적용하면 다음과 같다.

$$R_4 = \begin{bmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \end{bmatrix} \begin{bmatrix} 1&1&0&0 \\ 1&-1&0&0 \\ 0&0&1&1 \\ 0&0&1&-1 \end{bmatrix} \begin{bmatrix} 2&0&0&0 \\ 0&-1&0&0 \\ 0&0&1&0 \\ 0&0&0&4 \end{bmatrix} \begin{bmatrix} 1&0&1&0 \\ 0&1&0&1 \\ 1&0&-1&0 \\ 0&1&0&-1 \end{bmatrix} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{bmatrix}$$

$$= \begin{bmatrix} 2&-1&-1&2 \\ 1&-4&4&-1 \\ 1&4&-4&-1 \\ 2&1&1&2 \end{bmatrix}$$

$$R_4^{-1} = \frac{1}{4} \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \end{bmatrix} \begin{bmatrix} 1&0&1&0 \\ 0&1&0&1 \\ 1&0&-1&0 \\ 0&1&0&-1 \end{bmatrix} \begin{bmatrix} 1/2&0&0&0 \\ 0&-1&0&0 \\ 0&0&1&0 \\ 0&0&0&1/4 \end{bmatrix} \begin{bmatrix} 1&1&0&0 \\ 1&-1&0&0 \\ 0&0&1&1 \\ 0&0&1&-1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} 1&0&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1/2&1&1&1/2 \\ -1&-1/4&1/4&1 \\ -1&1/4&-1/4&1 \\ 1/2&-1&-1&1/2 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 2&4&4&2 \\ -4&-1&1&4 \\ -4&1&-1&4 \\ 2&-4&-4&2 \end{bmatrix}$$

Remark: i) 기본행렬을 $R_2 = \begin{bmatrix} 1 & 1 \\ 1 & -j \end{bmatrix}$라 하면 정의 1에 따라 R_4를 구한 후 (2.7)에 의해 $W_4 = U_4 R_4 U_4$ 이므로 4-점 FFT 행렬을 쉽게 구할 수 있어 FFT에 FRJT를 적용할 수 있다[5-6].

ii) FFT에 반하여 FRJT를 사용하는 장점은 다음 두 가지이다. 첫 째는, 행렬분해를 통해 FFT보다 더 간단한 구조를 가지고 있으며, 둘 째는, 데이터 열 이나 주파수 요소의 중앙부분을 특별히 강조할 필요가 있을 때 FFT보다 더 적합하다[2-4].

iii) 4-점 FRJT는 N 점 FFT에 응용될 수 있다[6].

Ⅳ. IFRJT와 ICWHT과의 비교

IFRJT의 효율성을 Inverse CWHT (ICWHT)와 비교하면 다음과 같다. 중앙 가중치 변환 (CWHT)을 실현하기 위한 가중치 계수 행렬 (weighted coefficient matrix) R^c는 다음과 같이 정의되었다[2-3].

$$R^C = H \cdot [WH] \tag{4.1}$$

여기서 H 는 하다마드 행렬을, [WH]는 가중치 하다마드 행렬을 칭한다. 가중치 하다마드 행렬은 분명히 리버스 자켓 행렬의 특별한 경우이다. 그래서 가중치 계수 행렬 R^c 는

$$R_N^C = H_N \cdot R_N \tag{4.2}$$

으로 표현될 수 있다. 하다마드와 리버스 자켓 행렬의 확장법을 이용하면

$$\begin{aligned}
R_N^C &= (H_{N/2} \otimes H_2)(R_{N/2} \otimes H_2) \\
&= (H_{N/2} R_{N/2}) \otimes (H_2 H_2) = R_{N/2}^C \otimes 2I_2
\end{aligned} \tag{4.3}$$

이 된다. 여기서 I_2는 2×2 단위행렬이다.

$H_N^{-1} = \frac{1}{N} H_N^T$ 이므로 (4.2)로부터 우리는 다음 식을 얻는다.

$$R_N = \frac{1}{N} H_N R_N^C \tag{4.4}$$

(4.4)의 역은

$$R_N^{-1} = N(R_N^C)^{-1} H_N^{-1}, = (R_N^C)^{-1} H_N^T \tag{4.5}$$

이 R_N^C는 각 행에서 영이 아닌 요소가 많아야 둘인 sparse 행렬이다. 따라서 이 행렬은 비교적 단순

하여 역을 구하기에 쉽다. 그러나 역을 구하기 위해서는 (4.5)에서 알 수 있듯, R_N^C의 역을 직접 계산해야 하거나 $(R_N^C)^{-1} = R_N^{-1}(H_N^T)^{-1} R_N$ (또는 [WH])의 역을 계산해야만 하는 불합리성을 내포한다. 다시 말하자면, R_N^C의 확장법 (4.3)을 이용하더라도 적어도 R_N^C의 역을 구하여야만 한다. 그러나 이를 위해 특별히 explicit한 알고리즘이 제시되지 못하였다[2-3]. 이러한 점을 고려하지 않고 단지 연산량만을 비교하더라도 덧셈에서는, $N=2^k$, a=b=c=1인 경우에 $N\log_2 N < k\cdot N + \frac{N}{2}$ 으로, 곱셈에서는 $\frac{N}{4} < \frac{3N}{2}$ 로 각각 IFRJT가 ICWHT보다 훨씬 더 효율적이다.

V. 결 론

본 논문에서는 고속 리버스 자켓 역변환 (IFRJT)을 제안하였고 이 방법은 역변환을 Kronecker product나 direct sum 연산자 등만을 사용하여 explcit하게 표현하였다. 이 알고리즘의 장점은 중앙 가중치 하다마드 역변환 (ICWHT)보다 더 빠르고 쉽게 주어진 행렬의 역을 구한다는 점이다. 우리는 얼마나 간단히 IFRJT를 얻을 수 있는지를 예제를 통해 보여 주었으며 IFRJT와 ICWHT의 효율성을 비교하였다.

참 고 문 헌

[1] O. ERSOY, Fourier-Related Transforms, *Fast Algorithms and Applications*, Prentice Hall International Editions, 1997.

[2] M. H. LEE, *The Center-weighted Hadamard Transform*, IEEE Trans. on Circuits and Systems-II, Vol. 36, No. 9, pp. 1247-1249, 1989.

[3] M. H. LEE, A New Reverse Jacket Transform and Its Fast Algorithm, IEEE Trans. on Circuits and Systems-II, Vol. 47, No. 1, pp. 39-47, 2000.

[4] S. -R. LEE AND M. H. LEE, *On the Reverse Jacket Matrix for Weighted Hadamard Transform*, IEEE Trans. on Circuits and Systems-II, Vol. 45, No. 3, pp. 436-441, 1998.

[5] S.-R. LEE AND K.-M. SUNG, *고속 리버스 자켓 변환과 그의 응용*, 한국통신학회논문지에 제출, 2001.

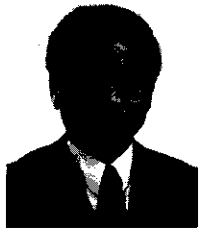
[6] S.-R. LEE AND K.-M. SUNG, *4-점 리버스 자켓 변환를 이용한 N-점 고속 리버스 자켓 변환*, 한국통신학회논문지, 제26권 제 4호, 2001.

[7] R. K. R. YARLAGADDA AND J. E. HERSHEY, *Hadamard Matrix Analysis and Synthesis*, Kluwer Academic Publishers, 1997.

이 승 래(Seung-Rae Lee) 정회원

1991년 : 독일 아헨공대 수학과 학사 (VorDiplom)
1993년 : 독일 아헨공대 수학과 석사 (Diplom Mathematiker)
1997년 : 독일 아헨공대 수학과 이학박사 (Dr. rer. Nat.)
1995년 7월~1996년 12월 : 독일 두이스부르그 대학 수학과 강사
1998년 9월~2000년 3월 : 서울대학교 제어계측신기술연구센터 Post Doc. 연구원
2000년 4월~2000년 11월 : 성균관대학교 전기전자 및 컴퓨터공학부 연구조교수
2000년 12월~현재 : 서울대학교 음향공학연구실 BK21 Post Doc. 연구원
<주관심 분야> 미분게임, 최적제어이론, 신호처리, 부호이론, 음향공학

성 굉 모(Koeng-Mo Sung) 정회원



1971년 : 서울대학교 공과대학 전자공학과 학사
1977년 : 독일 아헨공대 전자공학과 석사 (Diplom Ingenieur)
1982년 : 독일 아헨공대 전자공학과 공학박사 (Dr. Ing.)
1983년 7월~현재 : 서울대학교 전기공학부 교수
1997년 7월~1998년 9월 : 뉴미디어 통신공동연구소장
1998년 9월~2000년 9월 : 서울대학교 전기공학부 학부장
2000년 1월~현재 : 한국음향학회 회장
<주관심 분야> ultrasonics, musical acoustics, underwater acoustics, electro-acoustics, speech recognition and synthesis