1999년도 대한전기학회 정기총회 및 추계학술대회 논문집 1999. 11. 20

웨이브렛 패킷을 이용한 신호의 압축기법에 관한 연구

라경택*, 이진하*, 이영석**, 서보혁*
*경북대 대학원 전기공학과, **영진전문대 전기계열

Research of Compression Technique of Signal Using Wavelet Packet

Kyung-Taek La*, Jin-Ha Lee*, Young-Seog Lee**, Bo-Hyeok Seo*
Dept. of Electrical Eng. Kyungpook Nat. Univ.*, Dept. of Electrical Eng. Yeungjin Junior College**

Abstract - It is extremely important to compress a very large multimedia data under the currently used communication networks because of the forthcoming high fast multimedia informations.

Wavelets are very popular and useful mathematical tool in the field of signal processing recently.

In this paper, signal compression methods using the wavelet packet transform are compared with the other compression method under same threshold condition. As the result, the compression of using the wavelet packet transform is much more effective than the other compressing methods.

1. 서 론

최근 정보의 고속화, 멀티미디어 시대의 도래로 엄청난 양의 데이터가 텍스트, 음성, 정지화상과 동영상 등의 통합형태로 통신망에 유·출입되고 저장된다. 디지털 기술의 발달로 기존의 아날로그 방식으로 저장, 전송 복원되던 신호를 디지털 방식으로 처리하기 시작하였다. 디지털 방식은 아날로그 방식에 비해 데이터의 조작이 자유로와 개선된 통신망 서비스를 제공할 수 있지만, 다량의 디지털 데이터 량의 저장과 전송에는 많은 어려움과 비용이 소모되기 때문에 실용화에 어려움이 많았다. 이러한 이유로 비용절감을 위한 정보량의 압축방법이 필요하며 부호 화된 신호의 호환성을 위해 표준화된 압축방법이 필수적이다.

현재 통신에서 많이 사용되고 있는 DCT는 주어진 신호가 정상 신호로 이루어 졌을 때는 잘 적용되지만, 시간 영역에서는 급격한 변화를 가지는 비정상 신호는 해석하는데 있어서 불리하게 된다. 이러한 해상도 문제로 인한 DCT의 단점을 보완할 수 있는 웨이브렛 변환과 웨이브렛 패킷 변환이 필요로 하게 된다. 웨이브렛 기저함수들은 긴 시간영역의 저주파 함수와 짧은 시간 영역의 고주파 함수로 되어 있는데, 이 기저함수의 확대·축소를 통해 다양한 해상도를 만들 수 있으므로 DCT에 비해 급격한 변화를 가지는 신호에 대한 해석이 정확하게 하는 이점이 있어서, 기존의 DCT의 단점을 보완하는 웨이브렛 변환에 대한 연구가 활발히 진행중이다.

본 논문에서는 이러한 압축기법들 중 웨이브렛 패킷 변환을 이용한 압축에 대한 효용성에 대해 검토한다. 특히 웨이브렛 패킷 변환을 이용하여 압축을 실행하는데 있어 가지치기 알고리듬과 성장 알고리듬을 적용하여, 두 알고리듬의 효율도 비교를 한다.

2. 본 론

2.1 웨이브렛 변환

웨이브렛 변환은 푸리에 변환이 이용해 $-\infty \sim +\infty$ 주파수를 갖는 정현파로 분해하는 것과 같이 (1)식을 이용해 신호를 이동시키고 크기가 조절된 어미 웨이브렛으로 분해하는 것이다.

$$W_f(a,b) = \int_{-\infty}^{\infty} f(t)\psi(a,b)dt \qquad (1)$$

여기서, a는 스케일, b는 위치 또는 이동이며, ψ는 어미 웨이브렛이다. 위의 식에서 보는 바와 같이 웨이브렛 변환의 기저 함수는 어미 웨이브렛 $\psi(t)$를 시간 축에서 b만큼 이동시키고, a만큼 스케일링하여 얻어낸다.

웨이브렛 변환의 기본 개념은 임의의 함수 $f(x)$를 시간-주파수 공간에서 동시에 지역성을 갖는 웨이브렛 기저함수의 일차결합으로 표현하는 것이다. 기저 함수는 식 (2)의 원형 함수의 확대/축소와 변이에 의해 만들어진다.

$$\psi(a,b) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \qquad a,b \in R^2,\ a \neq 0 \qquad (2)$$

2.2 웨이브렛 패킷 변환

웨이브렛 패킷 변환은 신호에 대한 웨이브렛 변환의 일반화된 것이다.

웨이브렛 변환에서 신호는 저역통과 필터의 출력인 평균치와 고역통과 필터의 출력인 차분치로 나누어지고, 다시 평균치는 다음 레벨의 평균치와 차분치로 나누어지는 형태를 반복한다.

한편 웨이브렛 패킷 변환에서는 평균치뿐만 아니라 차분치도 같은 형태로 나눌 수 있다. 그림 1은 웨이브렛 패킷 변환의 형태를 보여 준다.

웨이브렛 패킷 해석은 그림 1의 신호 S는 다음 식 (3)과 같이 표현할 수 있다.

$$S = A_1 + DD_2 + AAD_3 + DAD_3 \qquad (3)$$

이것은 웨이브렛 변환에서는 불가능한 것이지만, 웨이브렛 패킷 변환에서는 이런 가능한 부호화를 통해 최적의 표현을 찾을 수 있음을 보여준다.

주어진 신호를 가장 잘 적절하게 분해하기 위해 각각의 트리 노드에서 얻어진 엔트로피에 근거한 정보량을 계산함으로서 최적의 tree를 탐색하여 얻을 수 있다.

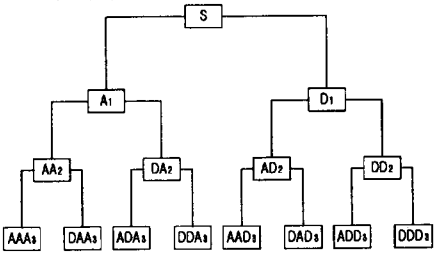

그림 1 웨이브릿 패킷 변환 Tree

웨이브렛 패킷 분해는 많은 기저를 발생하는데, 본 논문에서는 세넌 엔트로피에 근거하여 최적 트리를 결정한다. 식 (4)은 세넌 엔트로피 함수를 보여주고 있다.

$$M(x) = -\sum |x_i|^2 \log |x_i|^2 \quad (4)$$

웨이브렛 패킷을 사용하여 신호를 압축을 할 때, 다음과 같이 신호를 압축한다.

1) 분해(Decomposition of signal)
2) 최적 tree 계산(Computation the best tree)
3) 웨이브렛 패킷 계수의 경계치
(Thresholding of wavelet packet coefficient)
4) 복원(Reconstruction of signal)

3. 알고리듬

3.1 가지치기 알고리듬

가지치기 알고리듬은 최적 기저 또는 최적 트리 알고리듬으로 Coifman과 Wickauser에 의해 제안되었다. 주어진 신호와 부가성 성능지수에 대한 웨이브렛 패킷 라이브러리 중 최소 값을 갖는 기저를 선택하는 알고리듬이다. 우선 모든 가능한 변환 계수를 구하여 전체 트리로 확장을 한다. 그 다음 마지막 노드에서 처음으로 거슬러 올라가면서 트리의 가지치기를 한다. 이 가지치기는 노드와 그 자식 노드의 값을를 비교함으로써 수행된다. 가지치기 알고리듬은 다음과 같다.

1) 가능한 모든 웨이브렛 계수를 구하고, 전체 필터 뱅크 트리로 확장한다.

2) 각 노드에서 계수의 값를 구하고, 계수 값 트리 $\mathbf{M} = \{M_{ij}\}$를 생성한다.

3) $j = J$ 레벨에서 i번째 노드에 대해 $S_{ij} = 1$로 선택 tree를 초기화한다. 기저는 마지막 트리 레벨의 노드까지 수행되고 난 뒤에 선택된 것이다. 그리고 나머지는 $S_{ij} = 0$으로 세팅된다.

4) 모든 노드에 대해 M_{ij}을 계산하고 S_{ij}을 초기화한 후에 값을 비교/선택단계를 수행한다.

5) $\mathbf{S}$에서 1로 표시되는 노드들을 선택하고, $S_{ij} = 0$로 표시되는 가지들은 잘라버린다. 즉, 선택된 노드들 중에서 처음 노드와 가까운 노드를 선택하고 나머지는 잘라버린다.

3.2 성장 알고리듬

성장 알고리듬은 웨이브렛 패킷 기저 선택 알고리듬으로, 최상위에서 터미널 노드까지 걸쳐 실행되는 트리 구조의 기저 라이브러리에서 기저를 선택하는 방법이다. 가지는 자식들의 값이 현재보다 클 때까지 계속 자란다.

이 알고리듬은 전체 트리로 확장시킬 필요는 없고, 주어진 신호와 값에 대한 최적의 기저를 찾는 다고 볼 수 없다. 이 알고리듬은 웨이브렛 패킷 계수와 값을 비교 근사적으로 구하기 때문에 가지치기 알고리듬과는 달리 모든 변환 계수들을 가지는 행렬을 계산하고 저장할 필요가 없다.

알고리듬은 신호 자체를 변환하는 기저인 최상위 노드의 선택에서 시작하고, 이것으로부터 반복적인 알고리듬이 각 계산된 노드에 대해 적용 수행된다. 트리 레벨 j에서 i번째 노드에 연관되는 계수들을 $y_i^{(j)}$라고 하고, 이것의 값을 $M_{ij} = M(y_i^{(j)})$라고 할 때 먼저 최상위 노드를 기저로 선택하고 자기 자신의 신호를 변환시킨다. 여기서 적용되는 알고리듬은 다음과 같다.

1) 시작 노드에서 두 개의 자식 가지를 계산한다. 이것은 부모 노드에서 2-채널 필터링으로 얻을 수 있다.

2) 이 두 개의 자식 가지들에 대한 값을 계산한다.

3) $M_{ij} > M_{2i,j+1} + M_{2i+1,j+1}$ 이라면, $y_{2i}^{(j+1)}$의 노드는 새로운 어미 노드가 되며 단계 1)을 반복한다.(왼쪽가지 반복) 또는 $y_{2i+1}^{(j+1)}$의 노드는 새로운 어미 노드가 되며 단계 1)을 반복한다.(오른쪽 가지 반복)

이 외의 경우는 반복을 중단한다.

4. 모의 실험

검증 방법은 입력받은 음성신호를 현재 데이터 압축에 많이 사용하고 있는 DCT와 웨이브렛 변환, 그리고 웨이브렛 패킷 변환을 통하여 각각의 변환 결과로 얻은 계수를 임계치를 두고, 이를 다시 복원한 후 SNR(Sigmal-to-Noise Ratio)로 비교한다.

여기서 웨이브렛 변환과 웨이브렛 패킷 변환에서 기저 함수는 데이터 압축에 유용한 Daubechy 4 웨이브렛 함수를 사용하였다. 그리고, 웨이브렛 패킷 변환에서는 가지치기 알고리듬과 성장 알고리듬을 각각 적용하여 보았다.

기저 계수를 줄이기 위해 사용된 임계치는 Hard threshold로 아래 식 (5)과 같다.

$$d_{jk}^{hard} = \begin{cases} 0, & d_{jk} < \lambda \\ d_{jk}, & d_{jk} \geq \lambda \end{cases} \quad (5)$$

본 논문에서는 신호를 일반화 시키기 위해 임계치 λ를 0.1을 사용하였다.

표 1. 검증 시뮬레이션 결과 (단위 :dB)

	DCT	WT	WPK(P)	WPK(G)
Data 1	29.3224	30.4080	30.5858	30.4433
Data 2	24.8539	25.3929	25.9752	25.4288
Threshold	0.1000	0.1000	0.1000	0.1000

Data 1 : 16 ㎑ 16 bit precision
Data 2 : 12 ㎑ 8 bit precision
DCT : Dicrete Cosine Transform
WT : Wavelet Transform
WPK(P) : Wavelet Packet Transform(Pruning algorithm)
WPK(G) : Wavelet Packet Transform(Growth algorithm)

표 2. 컴퓨터 CPU time (단위 : sec)

	DCT	WT	WPK(P)	WPK(G)
Compression	1.98	1.07	3184.16	17.53
Reconstruction	2.15	2.37	214.43	12.49

표 1에서 보는 바와 같이 DCT에 비해 웨이브렛 패킷 변환이 더욱 개선된 음질을 보여줌을 알 수 있다. 또한 웨이브렛 패킷 변환에서도 가지치기 알고리듬으로 최적 기저를 계산한 경우가 양호한 결과를 보여준다.

그러나 표 2에서 보면, 이를 계산하는데 경과되는 CPU 시간은 웨이브렛 패킷 변환에서 확연히 많이 차지함을 알 수 있다. 특히 가지치기 알고리듬을 사용한 웨이브렛 패킷 변환의 경우가 많은 계산을 요구함을 알 수 있다.

다음의 그림에서는 Data 2에 대해 각각의 알고리듬을 적용한 결과를 보여 주고 있다.

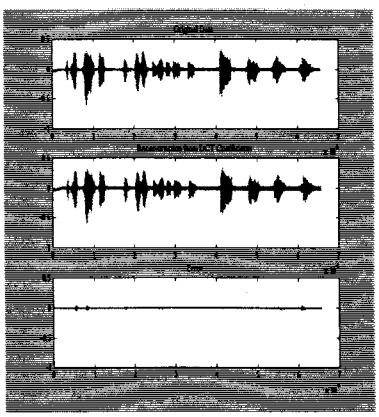

그림 2 Data 2의 DCT 변환 결과

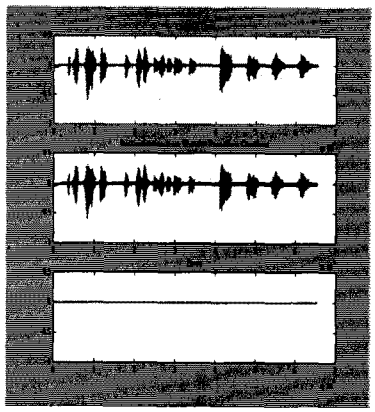

그림 3 Data 2의 웨이브릿 변환 결과

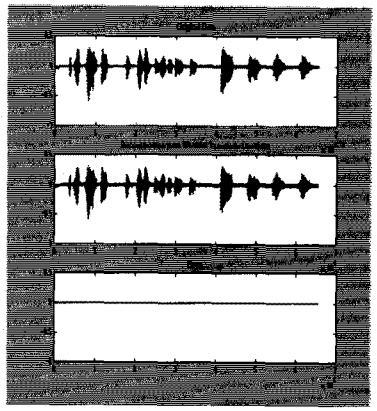

그림 4 Data 2의 웨이브릿 패킷 변환 결과(가지치기 알고리듬)

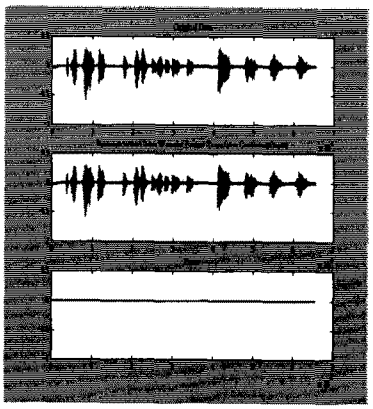

그림 5 Data 2의 웨이브릿 패킷 변환 결과(성장 알고리듬)

5. 결 론

본 논문에서는 단음절 음성 데이터와 하나의 단문장 음성 데이터를 가지고 DCT, 웨이브렛 변환, 웨이브렛 패킷 변환을 거치고 Hard threshold 시켜 얻은 계수들을 복원시키는 방법으로 알고리듬들을 검토하였다. 또한 웨이브렛 패킷 변환에서는 기저 선택 방법 중 가지치기 알고리듬과 성장 알고리듬의 두 가지 알고리듬을 적용시켜 보았다. 복원된 신호에 대해선 SNR을 이용하는 방법을 통해 평가를 해 보았다. 이 결과 웨이브렛 패킷 변환을 통한 경우에서 개선된 음질을 기대할 수 있었으며, 특히 가지치기 알고리듬을 이용한 경우에서 더욱 개선된 음질을 기대할 수 있었다.

그러나, 가지치기 알고리듬의 경우는 전체 트리의 계수를 비교/계산하여 기저를 선택하기 때문에 계산 량이 많아 다른 변환에 비해 소요시간을 많이 필요로 하는 단점이 나타났다.

그래서 DCT와 웨이브렛 변환에 비해 더 개선된 음질을 얻을 수 있다는 장점은 있지만, 웨이브렛 패킷 변환의 계산 소요시간이 너무 많이 요구됨에 따라 웨이브렛 패킷 변환의 속도 개선에 대한 연구가 필요하다.

[참 고 문 헌]

[1] P.P. Vaidyanathan "Multirate Systems and Filter Banks" *Prentice Hall.*

[2] B.S. Atal, V. Cuperman, A. Gersho "Advances In Speeching Coding" *Kluwer Academic Publishers*

[3] C. Sidney Burrus, Ramesh A. Gopinath, Haitao Guo "Introduction to Wavelets and Wavelet Transforms" *Prentice Hall*

[4] Y. T. Chan "Wavelet Basics" *Kluwer Academic Publishers*

[5] Albert Cohen, Jelena Kovacevic "Wavelets: The Mathematical Background" *Proceeding of the IEEE vol. 84, NO. 4, April 1996*

[6] W. Chong, J. Kim "Speech and Image Compression by DCT, Wavelet, and Wavelet Packet" *Proceeding of the 1997 International Conference on Information, communication, and Signal Pricessing, -Vol. 3, pp.1353~1357*

[7]Boon-Lum Lim, Zi-Lu Ying " Performance Analysis of Audio Signal Compression Based on Wavelet and Wavelet Packet Transform" *Proceeding of the 1997 International Conference on Information, communication, and Signal Pricessing,*

[8] Raquel Gomez-Sanchez, Diego Andina, Joaquín Torres "Speech Compression with Wavelet Packets" *IEEE Int. Conf. on Electronics, Circuits and Systems, Lisboa, Portugal, Vol. 3, pp. 117-120, Sept. 1998.*

[9] Ronald R. Coifman, Mladen Victor Wickerhauser "Entropy-Based Algorithm for Best Basis Selsction" *IEEE Transactions on Information Theory, Vol. 38, NO. 2, March 1992*

[10] Pierrick Philippe, Francois Moreau de Saint-Martin, Michel Lever "Wavelet Packet Filterbank for Low Time Delay Audio Coding" *IEEE Transaction on Speech and Audio Processing, vol. 7, NO. 3, May 1999*