한국음향학회지 제29권 제1호 pp. 41-47 (2010)

켑스트럼 기반 혼성영역 피치변경법의 처리시간 단축에 관한 연구

On a Processing Time Reduction of Cepstrum-Based Pitch Alteration in Time-Frequency Hybrid Domain

조 왕 래*, 김 종 국*, 배 명 진*
(Wang-Rae Jo*, Jong-Kuk Kim*, Myung-Jin Bae*)

*숭실대학교 정보통신공학과
(접수일자: 2009년 10월 16일; 수정일자: 2009년 11월 18일; 채택일자: 2009년 12월 4일)

음성변환을 위한 피치변경법은 시간영역법과 주파수영역법, 혼성영역법이 많이 사용되고 있으며 시간-주파수 혼성영역법은 스펙트럼 왜곡이 적고 명료성과 자연성이 우수하다는 장점이 있는 반면 영역변환을 위한 처리시간이 매우 길다는 단점을 가지고 있었다. 본 논문에서는 시간-주파수 혼성 영역 피치변경법의 처리시간을 단축하는 방법을 제안하였다. 음성신호를 켑스트럼으로 변경하는 과정에서 사용되는 FFT와 IFFT의 비트-재정렬 과정을 생략함으로써 처리시간을 단축하는 방법이다. 이를 적용함으로써 기존의 켑스트럼 피치변경법과 같은 음성품질을 유지하면서도 처리시간은 86.26%로 단축할 수 있었다.
핵심용어: 음성 변환, 피치 변경법, 켑스트럼 분석, 비트-재정렬
투고분야: 음성처리 분야 (2.4)

The pitch alteration technique for voice conversion is classified in time domain, frequency domain and hybrid domain. The Hybrid domain method has a merit of clearness and natural-ness of pitch altered speech but has the major drawback of long processing time.

In this paper, we proposed a new method that can reduce the processing time of pitch alteration in time-frequency hybrid domain. We omitted the bit-reversing process of FFT and IFFT in changing the processing domain. Therefore we can reduce the processing time by 86.26% to the conventional method with same quality.
Keywords: Voice Conversion, Pitch Alteration Technique, Cepstrum Analysis, Bit-Reversing
ASK subject classification: Speech Signal Processing (2.4)

I. 서 론

최근에 인간과 컴퓨터와의 자연스러운 통신을 위해 음성 언어를 이용한 휴먼 인터페이스 기술에 대한 관심이 점점 커지고 있다. 따라서 맨-머신 인터페이스를 위한 음성 응용 제품들이 증가하고 있으며 편리하고 빠른 인터페이스를 위한 기술 개선에 대한 요구가 더욱 많아지고 독특해졌다. 음성 언어를 이용한 인터페이스를 실현하기 위해서는 기계가 음성을 이해하고 기계가 음성을 생성하는 음성 인식, 음성 합성 기술이 필요하다. 특히 음성 합성 기술은 현재 상용화되어 서비스에 응용될 정도로 많은 기술적 발전이 있었다.

음성 변환 (voice conversion)기술은 현재의 음성 합성 방법만으로 해결하지 못한 음성 생성 기술이나 운율 제어 기술 등의 한계를 극복하기 위한 연구가 진행되고 있다 [1]. 음성 변환은 합성음에 감정정보를 포함시켜 정감도를 높이거나 특정화자의 음색으로 변환하는 기술이며 [2] 원시 화자 (source speaker)의 음성을 목표 화자 (target speaker)가 발성한 것처럼 발성 음성을 변환하는 과정이다. 원시 화자로부터 목표 화자의 음성으로 변환해 주기 위해서는 음운 및 운율의 변환이 이루어져야 한다 [3].

음성 변환을 위해 고려되어야할 화자의 개인성 요소는 크게 음향학적 요소와 운율적 요소로 나눌 수 있다. 음향

책임저자: 조 왕 래 (wrjo@naver.com)
156-743 서울시 동작구 상도동 숭실대학교 정보통신공학과
전화: 02-824-0906; 팩스:

학적 화자의 개인성 요소는 발성기관의 해부학적 구조의 차이, 발성기관을 이용한 조음 방법의 차이, 성대에서의 여기 신호의 특성 등에 의해 나타나는 포만트 주파수, 포만트 대역폭, 스펙트럼 경사와 성문 파형 (glottal waveform) 등이 있으며 운율적 개인성 요소에는 기본 주파수 궤적, 음소별 지속시간, 휴지기, 에너지 등이 있다 [3].

완전한 음성 변환을 위해서는 이러한 요소들의 변환이 모두 이루어져야 한다. 그러나 운율 요소의 변환은 화자의 발성습관을 모델링 하여야 한다는 점에서 매우 어려운 작업이며, 현재의 음성변환 기술들도 음향학적 요소의 변환에 주력하고 있는 실정이다. 일반적으로 음성 변환을 위해 여러 음향학적 요소를 포함하는 스펙트럼 포락(spectrum envelope)의 변환과, 개인성 요소에 가장 큰 영향을 미치는 운율 요소인 피치 주기 값만을 변환시키고 있다 [1][3].

지금까지 제안된 피치 변경법은 처리영역에 따라 시간영역법, 주파수영역법, 시간-주파수 혼성영역법으로 나눌 수 있다. 시간-주파수 혼성영역법은 켑스트럼의 특징을 이용하여 성도큐퍼런시와 피치펄스 사이의 켑스트럼 값이 거의 영 (zero)이 되는 부분에 영값 (zero value)을 삽입하거나 삭제함으로써 피치를 변경하는 방법이다 [4]. 이 방법은 스펙트럼 왜곡이 적고 명료성과 자연성이 우수하다는 장점이 있지만 음성신호를 시간영역에서 켑스트럼 영역으로 변환하고 큐퍼런시상에 영값을 삽입하거나 삭제하여 피치를 변경한 후 다시 시간영역으로 변환해야 하기 때문에 영역변환을 위한 처리시간이 매우 길다는 단점을 가지고 있었다.

본 논문에서는 켑스트럼 영역의 피치 변경법의 처리시간 단축 방법을 제안하였다. 음성신호를 켑스트럼 영역으로 변환하기 위한 과정에 수반되는 FFT와 IFFT 과정의 비트-재정렬 과정을 생략함으로써 처리시간을 단축하는 방법을 사용하였다.

II. 켑스트럼 분석법

음성신호는 시간영역에서 여기성분과 여파기성분의 컨벌루션으로 식 (1)과 같이 나타낼 수 있으며 주파수 영역에서 음성 스펙트럼은 식 (2)와 같이 여기 스펙트럼과 여파기 스펙트럼의 곱으로 나타낼 수 있다.

$$s(n) = e(n) * h(n) \tag{1}$$

$$S(K) = E(K) \cdot H(K) \tag{2}$$

여기에서 $s(n)$은 음성신호, $e(n)$은 여기신호를 나타내고 $h(n)$은 성도필터의 주파수 응답을 나타내며, $S(K)$, $E(K)$, $H(K)$는 각각의 푸리에 변환을 나타낸다.

이러한 스펙트럼을 로그형태로 나타내면 곱의 형태에서 합의 형태로 변환되기 때문에 여기성분과 여파기성분을 쉽게 분리할 수 있다. 이를 다시 시간영역으로 역변환하면 음성신호의 켑스트럼이 구해진다.

$$\begin{aligned} \hat{S}(K) &= \log[S(K)] \\ &= \log[E(K) \cdot H(K)] \\ &= \log[E(K)] + \log[H(K)] \\ &= \hat{E}(K) + \hat{H}(K) \end{aligned} \tag{3}$$

$$\hat{s}(n) = \hat{e}(n) + \hat{h}(n) \tag{4}$$

여기에서 $\hat{S}(K)$, $\hat{E}(K)$, $\hat{H}(K)$는 $S(K)$, $E(K)$, $H(K)$의 로그 스펙트럼을 나타내며, $\hat{s}(n)$, $\hat{e}(n)$, $\hat{h}(n)$은 로그 스펙트럼을 IFFT하여 얻은 켑스트럼을 말한다.

음성신호의 로그 스펙트럼과 켑스트럼을 그림 1에 나타내었다. 그림 1 (b)와 같이 켑스트럼의 낮은 큐퍼런시 영역에는 여파기 모델에 관한 정보가 들어 있고, 높은 큐퍼런시 영역에는 여기 모델에 관한 정보가 들어있다 [4]. 따라서 식 (5)와 같은 리프터 (lifter)를 이용하면 성도 여파기의 특성을 구할 수 있다.

$$\begin{aligned} l(n) &= 1, & |n| < n_0 \\ &= 0, & |n| \geq n_0 \end{aligned} \tag{5}$$

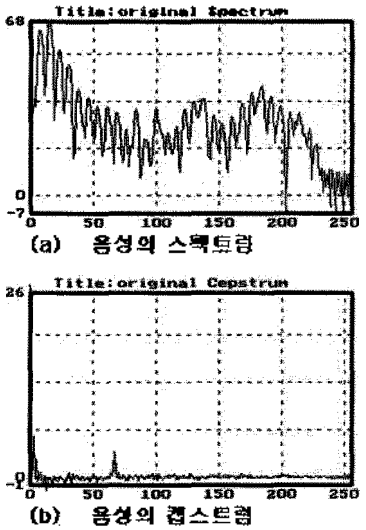

그림 1. 음성신호의 스펙트럼과 켑스트럼
Fig. 1. Spectrum and Cepstrum of Speech.

여기서 n_0는 피치주기 N_p보다 작게 선택된다. 이렇게 구해진 여파기 스펙트럼은 음성신호의 공명 특성을 나타내며 포만트 스펙트럼과 같아진다. 또한 캡스트럼상의 성도 여파기 특성은 큐퍼런시가 증가함에 따라 급속히 감소하는 특성을 갖는다 [5].

한편, 음성 캡스트럼에 식 (6)과 같은 리프터 (lifter)를 적용하면 음성신호의 여기 특성을 구할 수 있다.

$$\begin{aligned} l(n) &= 0, \quad |n| < n_0 \\ &= 1, \quad |n| \geq n_0 \end{aligned} \tag{6}$$

Ⅲ. 캡스트럼 피치변경법의 처리시간 단축

캡스트럼 분석에 의해 음성을 합성할 때는 낮은 큐퍼런시의 캡스트럼을 여파기 특성으로 취하고 높은 큐퍼런시의 캡스트럼을 여기 특성으로 취하여 이들을 컨볼루션함으로써 음성신호를 합성한다. 이때 여기 특성을 변경하여 피치를 변경할 수 있게 된다.

캡스트럼의 특징은 대부분의 캡스트럼 값이 영 (zero) 큐퍼런시 부근에 존재하며 이들 값은 큐퍼런시 증가에 따라 급속히 감소하여 피치주기 부근에서는 거의 영이 된다. 피치를 변경하기 위해서는 캡스트럼 값이 거의 영이 되는 부분에 변경하려는 주기만큼의 영 캡스트럼을 삽입하거나 삭제하면 된다. 이러한 방법은 여파기 특성에는 거의 영향을 주지 않으면서 여기 특성만을 변경시키기 위해 영값을 삽입하거나 삭제하기 위한 위치의 선정이 매우 중요하다. 현재 분석중인 음성구간의 피치를 사전에 알고 있다면 피치주기 근방에 영값을 삽입하거나 삭제하는 것이 바람직하다. 그러나 분석중인 창함수내에서 시간에 따라 피치 주기가 변화하고 있는 경우에는 피치주기 근방의 캡스트럼 필스가 일정 폭을 유지하게 되어 영값을 삽입하거나 삭제하기 위한 위치의 선정에 어려움이 따르게 된다. 잘못된 위치 선정은 합성음질에 큰 열화를 초래하게 된다.

음성신호의 캡스트럼을 구하는 방법은 FFT (Fast Fourier Transform)를 이용하거나 LPC (Linear Prediction Coefficients) 분석을 이용할 수 있으며 전자를 FFT 캡스트럼이라 하고 후자를 LPC 캡스트럼이라 한다 [6]. FFT 캡스트럼은 호모몰픽 디컨볼루션의 특성 시스템을 이용한 분석방법으로 그림 2에 나타낸 바와 같이 입력된 음성신호의 FFT를 구하고 로그 연산 후 다시 IFFT를 적용함으로써 구할 수 있다 [6].

FFT는 DFT (Discrete Fourier Transform)를 계산하는데 있어 결과는 같으면서도 연산수를 줄여 계산속도를 높이는 방법이다. 일반적인 DFT의 계산식은 식 (7), 식 (8)과 같다.

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} \tag{7}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, W_N^{-kn} \tag{8}$$

계산량을 살펴보면 N개의 샘플을 DFT하는데 각 n에 대하여 N번의 복소수 곱셈이 필요하게 되어 결과적으로 N^2에 비례하는 계산량이 필요하게 된다. 그러나 N개의 샘플을 FFT하는 경우에는 같은 결과를 내면서 계산량은 $N \times \log_2 N$에 비례하도록 줄일 수 있다.

FFT는 그림 3과 같이 DIT (decimation in time)와 DIF (decimation in frequency) 각각에 대해 정상 순서의 입력을 사용한 경우와 비트-재정렬된 입력을 사용하는 방법이 있다. FFT 알고리즘에 가장 많이 사용되는 Cooley-Tukey 알고리즘은 DIF 방법을 사용하며, IFFT의 경우에는 FFT와 같은 방법을 사용하면서 단지 계수들의 켤레 복소수 (complex conjugate)를 사용하고 루틴의 끝에서 1/N 스케일링 (scaling)을 수행하는 것만이 다르다 [7]. 그러나 FFT는 계산하고자하는 데이터 샘플수가 $N = 2^v$ (v는 정수)가 되어야 한다는 것과 그림 3(a)에 나타낸 바와 같이 입력배열의 순서가 0~7까지 정상적인 순서로 입력되어도 출력배열은 0, 4, 2, 6, 1, 5, 3, 7로 출력되어 입력배열과 출력배열의 순서가 서로 일치하지 않는다는 단점이 있다. 따라서 FFT 수행 전이나 수행 후에 배열의 순서를 재정렬해 주어야만 한다. 이를 비트-재정렬 (bit-reversing)이라 하며 계산량에 있어 큰 오버헤드로 작용하게 된다. 이러한 오버헤드는 적은 샘플수를 갖는 데이터에 대한 FFT 연산이 DFT에 비해 큰 이점이 없도록 하며 캡스트럼 분석과 같이 시간-주파수 영역 변환이 잦은 연산의 처리속도에 큰 영향을 미치게 된다 [3].

본 논문에서는 캡스트럼의 특징을 이용해 피치를 변경할 때 영역변경에 수반되는 FFT와 IFFT의 비트-재정렬

그림 2. FFT 캡스트럼 분석과정
Fig. 2. Process of FFT Cepstrum.

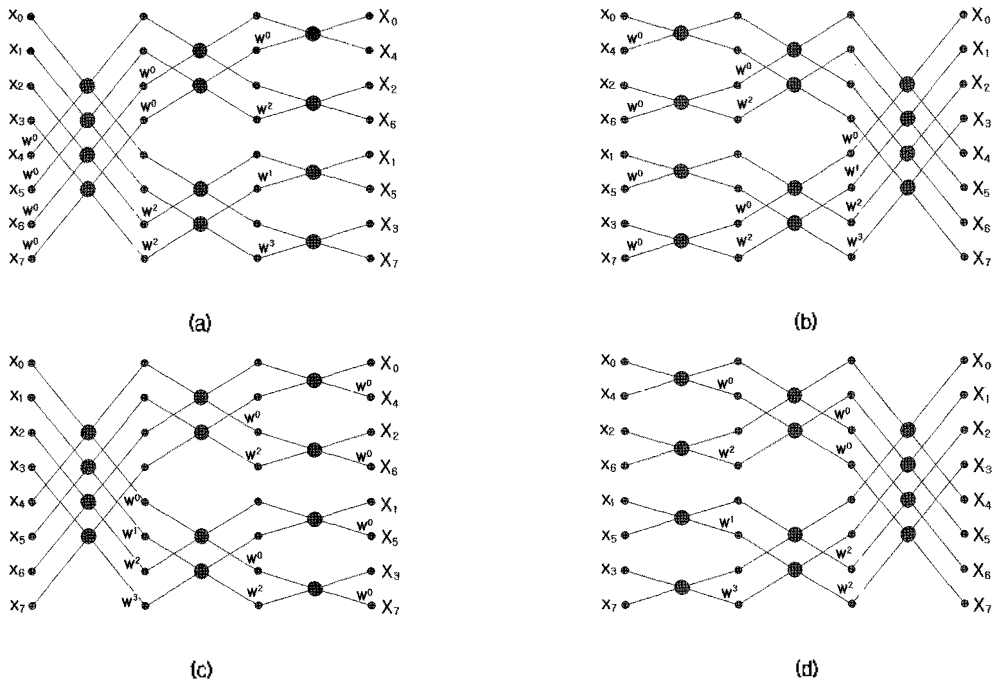

그림 3. 8-point FFT의 흐름도
(a) 정상순서 입력의 DIT 흐름도 (b) 비트-재정렬 입력의 DIT 흐름도
(c) 정상순서 입력의 DIF 흐름도 (d) 비트-재정렬 입력의 DIF 흐름도

Fig. 3. Flow graphs for 8 point FFT.
(a) DIT flow graph with normal ordered inputs (b) DIT flow graph with bit-reversed ordered inputs
(c) DIF flow graph with normal ordered inputs (d) DIF flow graph with bit-reversed ordered inputs

과정을 생략함으로써 피치 변경 시간을 단축하는 새로운 방법을 제안하였다. 그림 4에 음성 신호를 켑스트럼으로 변환하기 위한 기존의 변환방법과 제안한 변환방법을 블록도로 나타내고 8 포인트 음성 데이터의 처리과정을 예로 들어 배열 순서의 변화과정을 나타내었다. 그림 4 (a)에 나타낸 것처럼 기존의 방법에서는 FFT에는 정상순서 입력의 DIF 연산후에 출력배열의 순서를 다시 정렬해주는 비트-재정렬을 수행하고 로그 연산 후 다시 IFFT하기 위해 정상순서 입력의 DIF 연산과 비트-재정렬을 수행한다. 그림 4 (a)의 배열 (b)와 배열 (e)와 같이 FFT와 IFFT의 버터플라이 연산 (DIF) 후에는 배열의 순서가 바뀌므로 정상순서로 재정렬 해 주어야만 한다. 제안한 방법은 그림 4 (b)와 같이 정상순서 입력의 DIF 연산으로 FFT를 수행하고 비트-재정렬 하지 않고 로그 연산을 수행한다. 이를 그림 3 (d)에 나타낸 비트-재정렬 입력의 DIF 연산으로 IFFT 하면 정상 순서의 출력을 얻을 수 있게 된다. 따라서 FFT와 IFFT에서 처리시간의 오버헤드로 작용하는 비트-재정렬과정을 생략할 수 있게 된다.

켑스트럼 피치 변경법은 음성 신호를 앞에서 설명한 방법을 사용하여 켑스트럼으로 변환하고 '0' 큐퍼런시와 피치펄스 사이의 에너지가 작은 부분에 '0'값을 삽입하거

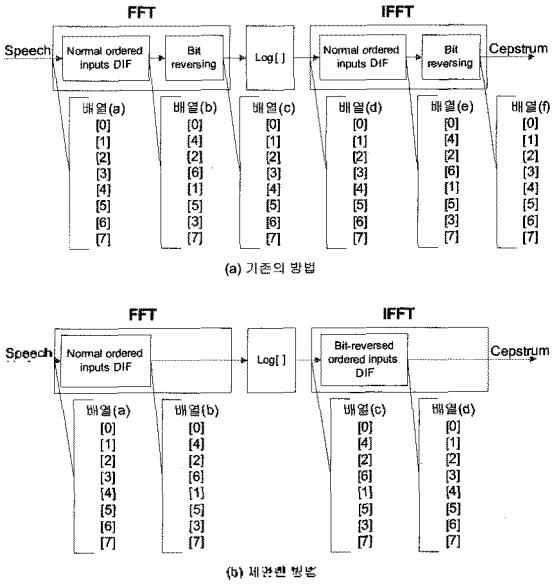

그림 4. 켑스트럼 변환과정과 배열순서의 변화

Fig. 4. Process of Cepstrum transform and change of array index.

나 삭제하여 피치를 변경하고 다시 음성 신호로 변환하는 방법이다. 음성 신호를 켑스트럼으로 변환하기 위해서 FFT와 IFFT를 사용하고 피치 변경후 다시 음성 신호로 변환하기 위해서도 FFT와 IFFT가 사용된다. 제안한 피

치변경법의 처리과정을 그림 4에 나타내었다. 입력된 음성을 프레임 단위로 해밍윈도우를 취하고 FFT하여 주파수 영역으로 변환한다. 그 결과에 로그연산을 적용하여 로그스펙트럼을 구한 후에 IFFT하여 켑스트럼 영역으로 변환한다. 켑스트럼상의 '0' 쿼퍼런시와 피치펄스 사이에서 변경하고자 하는 피치값만큼 영값을 삽입하거나 삭제하여 피치를 변경한다. 피치가 변경된 켑스트럼을 다시 음성신호로 복원하기 위하여 FFT하고 지수함수를 적용하여 IFFT를 취하면 피치가 변경된 음성 신호가 생성된다. 참고로 음영으로 나타낸 바와 같이 켑스트럼 변환을 위한 FFT, IFFT과정에서 비트-재정렬과정은 한 프레임당 4회가 발생하게 되어 처리시간에 큰 영향을 주게 되며 제안한 알고리즘에서는 비트-재정렬과정을 모두 생략하게 되어 처리시간이 많이 단축될 수 있다.

IV. 실험 및 결과고찰

제안한 켑스트럼 피치 변경법의 성능을 평가하기 위하여 기존의 방법과 제안한 방법을 Lenovo/T-61p 노트북 (Intel Core2 Duo 2.6 GHz)에서 C++로 구현하였다. 기존의 방법이나 제안한 방법에 사용된 FFT 알고리즘과 IFFT 알고리즘은 일반적인 신호처리에 많이 사용되는 Cooley-Tukey 알고리즘을 사용하였다 [7].

먼저 FFT 알고리즘의 전체 처리시간에서 비트-재정렬 과정이 차지하는 비율을 알아보기 위하여 128포인트, 256포인트, 512포인트 FFT의 전체 처리시간과 비트-재정렬 처리시간을 측정하였다. 각각에 대하여 1,000회씩 측정하여 평균값을 구하여 비교하였다. 표 1에 나타낸 바와 같이 128포인트 FFT의 경우 전체 처리 시간의 13.21 %에 해당하는 시간이 비트-재정렬에 소요됨을 알 수 있다.

두 번째로 제안한 방법을 켑스트럼 피치변경법에 적용하여 기존의 방법과의 처리시간을 비교하였다. 기존의 방법은 정상순서 입력의 DIF 계산을 수행한 후 비트-재정렬하는 방법으로 FFT와 IFFT를 수행한 경우이고 제안한 방법은 FFT에는 정상순서 입력의 DIF를 사용하고 IFFT에는 비트-재정렬된 입력의 DIF를 사용하여 비트-재정렬 과정을 생략한 경우이다.

그림 5에 나타낸 바와 같이 한 프레임의 음성신호에 대해 해밍윈도우를 취하고 FFT하여 주파수 영역으로 변환하였다. 여기에 로그연산을 적용하여 로그스펙트럼을 구하고 IFFT하여 켑스트럼 영역으로 변환하였다. 켑스트럼상에서 영값 삽입이나 삭제를 통하여 피치를 변경하고 다시 FFT, 지수함수, IFFT를 수행하여 피치가 변경된 음성신호를 재구성하였다.

기존의 켑스트럼 피치변경법과 제안한 켑스트럼 피치변경법의 계산시간을 비교하기 위하여 128샘플, 256샘플, 512샘플을 한 프레임으로 처리하여 계산시간을 비교하였다. 표 2에 나타난 바와 같이 256샘플에 대한 켑스트럼 피치변경의 경우 제안한 방법이 2612.60 μs가 소요되어 기존의 방법 3028.63 μs에 비하여 86.26 %로 단축됨을 알 수 있었다. 그림 6에는 제안한 켑스트럼 피치변경법에 의해 피치주기를 120 % 변경한 경우의 스펙트럼과 켑스트럼을 나타내었다.

다음으로는 기존의 켑스트럼 피치변경법과 제안한 켑스트럼 피치변경법을 샘플 문장에 적용하여 처리시간을 측정하였다. 사용된 샘플문장은 /인수네 꼬마는 천재소년을 좋아한다./라는 3초간의 문장으로 8 kHz로 샘플링하고 16비트로 양자화하였다. 피치변경을 위한 프레임은 256샘플로 처리하였고 128샘플씩 오버랩하여 처리하였다. 표 3에 나타낸 바와 같이 피치를 90 %로 압축하는

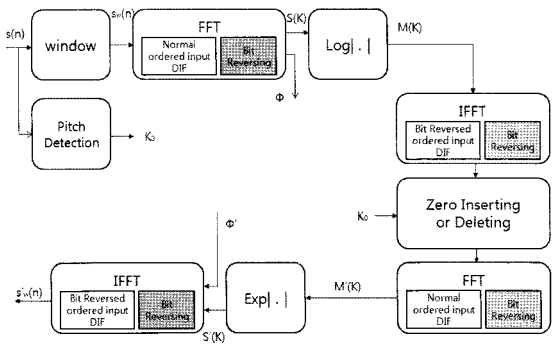

그림 5. 제안한 알고리즘의 블록도
Fig. 5. Block diagram of proposed algorithm.

표 1. FFT 처리시간과 비트-재정렬시간의 비
Table 1. The ratio of FFT time vs. Bit-reversing time.

	처리 시간(μs)		비트-재정렬 시간 비율 (A/B)
	전체처리시간 (A)	비트-재정렬 시간 (B)	
128points	399.90	44.91	13.21%
256points	617.69	76.56	12.39%
512points	1306.73	173.32	13.26%

표 2. 단일 프레임데이터 처리시간의 비교
Table 2. Comparison of processing time of one frame data.

	처리 시간(μs)		처리시간 단축율(B/A)
	기존방법(A)	제안한방법(B)	
128points	1304.43	1118.94	85.78%
256points	3028.63	2612.60	86.26%
512points	6539.18	5625.65	86.05%

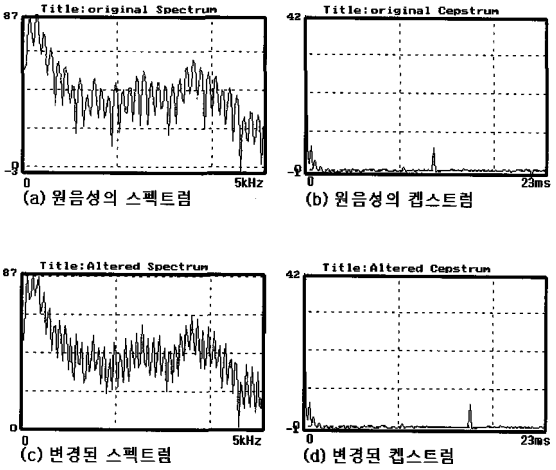

그림 6. 피치주기를 120%로 신장한 경우
Fig. 6. Case of Pitch Expanded signal by 120 %.

표 3. 문장 처리시간의 비교
Table 3. Comparison of processing time of one sentence.

	처리 시간(μs)		처리시간 단축율(B/A)
	기존방법(A)	제안한방법(B)	
피치압축(90%)	540894.5	464588.6	85.89%
피치신장(120%)	554830.3	478523.7	86.25%

경우 기존의 방법이 540,894.5 μs의 처리시간이 소요되는데 비하여 제안한 방법은 464,588.6 μs가 소요되어 기존 방법의 85.89 %로 처리시간이 단축됨을 알 수 있다. 그림 7에는 피치변경된 음성파형을 나타내었다.

V. 결 론

최근 음성을 이용한 휴먼 인터페이스 기술이 많이 활용되면서 음성변환 기술에 대한 연구가 진전되고 있다. 음성변환을 위한 피치변경법은 시간영역법과 주파수영역법, 혼성영역법이 많이 사용되고 있으며 시간-주파수 혼성영역법은 스펙트럼 왜곡이 적고 명료성과 자연성이 우수하다는 장점이 있는 반면 영역변환을 위한 처리시간이 매우 길다는 단점을 가지고 있었다. 본 논문에서는 시간-주파수 혼성 영역 피치변경법의 처리시간을 단축하는 방법을 제안하였다. 음성신호를 켑스트럼으로 변경하는 과정에서 사용되는 FFT와 IFFT의 버트-재정렬 과정을 생략함으로써 처리시간을 단축하는 방법이다. 이를 적용함으로써 기존의 켑스트럼 피치변경법과 같은 음성품질을 유지하면서도 256샘플 처리시간은 86.26 %로 단축할 수 있었다.

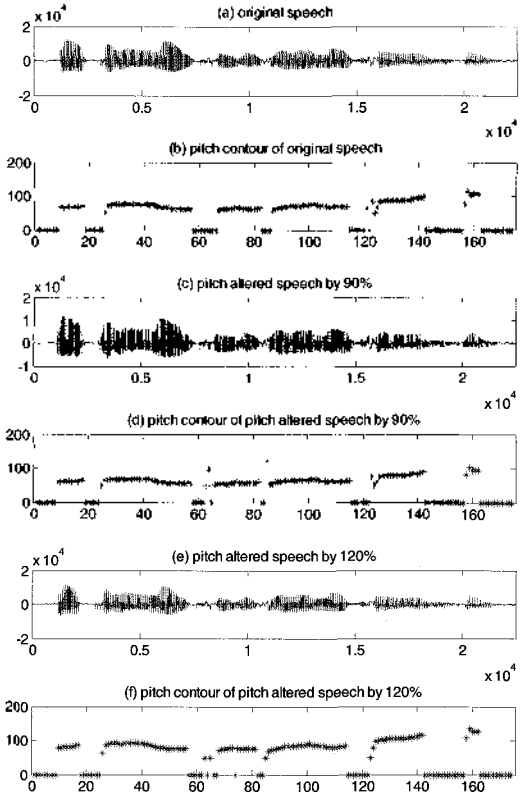

그림 7. 피치 변경 결과 파형
Fig. 7. The result waveform of pitch alteration.

참 고 문 헌

1. M. J. Bae, "The TTS Speech Synthesis Techniques," *Proceedings of Korea Inst. Commu. Sciences*, vol. 11, no. 9, pp. 67-78, 1994.
2. M. J. Bae, "On a Voice Color Change in the Fairy Tale Narration System with Parent's Voice Color," *J. Acoust. Soc. Korea*, vol. 16, no. 8, pp. 131-135, 1997.
3. 김종국, *음성변환을 위한 운율구에 기반한 피치 궤적 변환 기술에 관한 연구*, 숭실대학교 박사학위논문, 2005.
4. M. J. Bae, S. H. Lee, "On a Cepstral Technique for Pitch Control in the High Quality Text-to-Speech Type System," *39'th Midwest symposium on circuits and Systems, Proceeding of MWSCAS'96*, pp. 803-806, 1996.
5. 전선도, 강철호, "잡음에 강한 음성 인식을 위한 성문 가중 캡스트럼에 관한 연구," *한국음향학회지, 제18권, 제5호*, pp. 78~82, 1999.
6. 정혜경, 김유진, 정재호, "켑스트럼으로부터 변환된 로그 스펙트럼을 이용한 포먼트 평활화 켑스트럴 평균 차감법," *한국음향학회지, 제21권, 제4호*, pp. 361~373, 2002.
7. Embree, Paul M. & Bruce Kimble, *C Language Algorithms for Digital Signal Processing*, Prentice-Hall, 1991.

저자 약력

•조 왕 래 (Wang-Rae Jo)

1996년: 숭실대학교 정보통신공학과 (학사)
1998년: 숭실대학교 전기공학과 (석사)
2003년 ~ 현재: 디비정보통신 책임연구원
※ 주관심 분야: 음성 신호처리

•김 종 국 (Jong-Kuk Kim)

한국음향학회지 제 26권 제8호 참조

•배 명 진 (Myung-Jin Bae)

현재: 숭실대학교 정보통신전자공학부 교수
한국음향학회지 제 26권 제4호 참조