(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G10L 19/00 (2006.01)

(45) 공고일자 2007년08월03일
(11) 등록번호 10-0745387
(24) 등록일자 2007년07월27일

(21) 출원번호	10-2000-7014271	(65) 공개번호	10-2001-0052915
(22) 출원일자	2000년12월15일	(43) 공개일자	2001년06월25일
심사청구일자	2005년02월04일		
번역문 제출일자	2000년12월15일		
(86) 국제출원번호	PCT/US2000/010477	(87) 국제공개번호	WO 2000/63882
국제출원일자	2000년04월19일	국제공개일자	2000년10월26일

(81) 지정국 국내특허 : 캐나다, 일본, 대한민국, 멕시코, 미국,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀란드, 사이프러스,

(30) 우선권주장 60/130,016 1999년04월19일 미국(US)

(73) 특허권자 에이티 앤드 티 코포레이션
미국 뉴욕 10013-2412 뉴욕 애비뉴 오브 디 아메리카즈 32

(72) 발명자 캐필로우데이빗에이.
미국,뉴저지07922,버클리헤이츠,워싱톤스트리트122

(74) 대리인 이범래
이병호
장훈

(56) 선행기술조사문헌
US 5615298 US 5907822

심사관 : 장호근

전체 청구항 수 : 총 30 항

(54) 패킷 손실 또는 프레임 삭제 은폐를 실행하는 방법 및 장치

(57) 요약

본 발명은 전용 또는 표준 FEC 처리를 갖지 않는 음성 코더(speed coder)에서 패킷(packet) 손실 또는 프레임 삭제 은폐(Frame Erasure Concealment, FEC)를 실행하기 위한 방법 및 장치에 관한 것이다. 디코더를 갖춘 수신기는 인코더로부터 전송된 압축 음성 정보의 부호화 프레임들을 수신한다. 수신기에 있는 손실 프레임 검출기는 부호화 프레임이 전송시 손실되거나 변형되었나, 또는 삭제되었는지를 결정한다. 부호화 프레임이 삭제되지 않았으면, 부호화 프레임은 디코더에

의해 복호화되고 임시 메모리는 디코더의 출력으로 업데이트된다. 소정의 지연 기간이 적용되어 오디오 프레임이 출력된다. 부호화 프레임이 삭제된 것으로 손실 프레임 검출기가 결정하면, FEC 모듈은 신호에 프레임 은폐 처리를 적용한다. FEC 처리는 삭제된 프레임에 대해 자연스럽게 들리는 합성 음성을 만든다.

대표도

도 1

특허청구의 범위

청구항 1.

FEC(Frame Erasure Concealment) 능력을 갖지 않는 음성 코더를 위한 프레임 삭제 은폐(FEC)를 실행하는 방법에 있어서,

인코더로부터 전송되는 압축된 음성 정보의 부호화 프레임들을 수신하는 단계;

부호화 프레임이 전송 중 삭제, 손실, 또는 오염되었는지를 결정하는 단계;

상기 결정 단계에서 상기 부호화 프레임이 삭제되지 않았다고 결정하면, 디코더에서 상기 수신된 부호화 프레임들을 복호화 프레임들로 복호화하는 단계;

상기 결정 단계에서 상기 부호화 프레임이 삭제되었다고 결정하면, 이전의 복호화 프레임의 데이터를 사용하여 상기 삭제된 프레임을 위한 합성 프레임을 생성하는 단계;

상기 복호화 및 합성 프레임들의 데이터로 제 1 메모리를 업데이트하는 단계; 및

상기 복호화 및 합성 프레임들을 오디오 신호로서 출력하는 단계;를 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 2.

제 1 항에 있어서,

지연 메모리를 사용하여 미리 결정된 시간 기간만큼 상기 오디오 신호의 출력을 지연시키는 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 3.

제 1 항에 있어서,

하나 또는 그 이상의 삭제된 프레임들의 시퀀스 내의 제 1 삭제된 프레임에서 상기 생성 단계는,

상기 삭제된 프레임의 피치 기간을 추정하는 단계;

상기 제 1 메모리로부터 제 2 메모리로 가장 최근 피치 기간들 중 하나 또는 그 이상을 복사하는 단계;

상기 제 1 메모리로부터 제 3 메모리로 가장 최근 피치 기간의 일부분을 복사하는 단계;

상기 제 2 메모리 내에 저장된 피치 기간의 일부분으로 상기 제 3 메모리를 오버랩 가산하는 단계;

상기 제 2 메모리의 가장 최근 부분 내에 상기 피치 기간의 오버랩 가산된 부분을 저장하는 단계;

상기 업데이트 단계에서 상기 제 1 메모리를 업데이트하고, 상기 출력 단계에서 출력하기 위해, 상기 제 2 메모리에 저장된 가장 최근 피치 기간들 중 하나 또는 그 이상에 기초하여 합성 프레임을 생성하는 단계; 및

상기 생성 단계에서 사용된 상기 제 2 메모리 내에 최후의 위치를 오프셋 포인터에 저장하는 단계;를 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 4.

제 3 항에 있어서,

상기 오버랩 가산된 부분으로 상기 지연 메모리를 업데이팅하는 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 5.

제 3 항에 있어서,

하나 또는 그 이상의 후속되는 부호화된 프레임들이, 합성 프레임이 출력된 후에 삭제되었는지를 결정하는 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 6.

제 5 항에 있어서,

하나 또는 그 이상의 후속되는 부호화된 프레임들이 삭제되었다면, 상기 방법은,

연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과하는 지를 결정하는 단계;

상기 결정 단계에서 연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과하지 않는다고 결정하면, 상기 합성 프레임을 생성하는데 사용된 피치 기간들의 수를 변경하는 단계; 및

상기 결정 단계에서 연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과한다고 결정하면, 현재 피치 기간들의 수로 상기 합성 프레임 생성을 지속하는 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 7.

제 6 항에 있어서,

상기 연속적으로 삭제된 프레임의 수가 미리 결정된 길이를 초과하지 않는다면, 상기 변경 단계는,

피치 기간의 부분을 위하여 이전의 프레임으로부터 상기 합성 신호의 생성을 지속하고, 제 4 메모리로 그 결과를 배치하는 단계;

제 2 메모리 내에 사용된 피치 기간들의 수를 증가시키는 단계;

상기 제 2 메모리 내에 저장된 피치 기간의 부분으로 상기 제 3 메모리를 오버랩 가산하고, 상기 제 2 메모리의 가장 최근 부분에 그 결과를 저장하는 단계;

상기 오프셋 포인터가 상기 제 2 메모리의 새로운 부분을 지시할 때까지, 상기 오프셋 포인터를 피치 기간들만큼 이동하는 단계;

상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써, 합성 프레임의 데이터를 생성하는 단계;

상기 생성 단계에 사용된 상기 제 2 메모리 내의 최후의 위치를 오프셋 포인터 내에 저장하는 단계;

상기 제 5 메모리 내의 피치 기간의 부분으로 상기 제 4 메모리를 오버랩 가산하고, 그 결과를 상기 제 5 메모리 내에 저장하는 단계;

상기 삭제된 프레임들이 미리 결정된 길이를 초과하는지를 결정하는 단계; 및

마지막으로 언급된 결정 단계에서 상기 삭제된 프레임이 미리 결정된 길이를 초과한다고 결정하면, 상기 삭제 부분의 길이에 기초하여 상기 제 5 메모리를 감소시키는(attenuating) 단계로서, 상기 제 5 메모리는 상기 업데이팅 단계에서 상기 제 1 메모리를 업데이팅하고 상기 출력 단계에서 출력하기 위해 사용되는, 상기 감소 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법

청구항 8.

제 6 항에 있어서,

상기 연속적으로 삭제되는 프레임들의 수가 미리 결정된 길이를 초과한다면, 상기 지속 단계는,

상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써, 합성 프레임의 데이터를 생성하는 단계;

상기 생성 단계에 사용된 상기 제 2 메모리의 최후의 위치를 오프셋 포인터 내에 저장하는 단계;

상기 삭제된 프레임들이 미리 결정된 길이를 초과하는지를 결정하는 단계; 및

마지막으로 언급된 결정 단계에서 상기 삭제된 프레임들이 미리 결정된 길이를 초과한다고 결정하면, 상기 삭제 부분의 길이에 기초하여 상기 제 5 메모리를 감소시키는 단계로서, 상기 제 5 메모리는 업데이팅 단계에서 상기 제 1 메모리를 업데이팅하고 상기 출력 단계에서 출력하기 위해 사용되는, 상기 감쇠 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 9.

제 5 항에 있어서,

상기 후속하여 부호화된 프레임들이 삭제되지 않는다면, 상기 방법은,

FEC 프로세스를 사용하여 제 1 비삭제 부호화 프레임을 처리하는 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 방법.

청구항 10.

제 9 항에 있어서,

상기 처리 단계는,

상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써, 합성 프레임의 데이터를 생성하는 단계;

상기 삭제된 프레임이 미리 결정된 길이를 초과하는지를 결정하는 단계;

상기 결정 단계에서 상기 삭제된 프레임이 미리 결정된 길이를 초과하는 것으로 결정하면, 상기 삭제 부분의 길이에 기초하여 상기 제 5 메모리를 감소시키는 단계;

제 1 비삭제 복호화 프레임의 시작부로 상기 제 5 메모리의 부분을 오버랩 가산하고, 상기 제 5 메모리 내에 그 결과를 저장하는 단계; 및

상기 제 5 메모리로 상기 복호화 프레임의 비오버랩 가산된 영역을 복사하는 단계로서, 상기 제 5 메모리는 상기 업데이팅 단계에서 상기 제 1 메모리를 업데이트하고 상기 출력 단계에서 출력하는데 사용되는, 상기 복사 단계;를 더 포함하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 11.

제 1 항에 있어서,

상기 합성 프레임들을 생성하기 위해 사용된 피치 기간들의 수는, 시간의 함수로서 증가하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 12.

제 1 항에 있어서,

상기 생성 단계는 인코더를 사용하여, 디코더의 상태 변수들이 상기 합성 프레임들을 추적하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 13.

제 1 항에 있어서,

고주파수 입력 신호들에 대하여, 하나 또는 이상의 피치 기간들이 상기 합성 프레임들을 생성하기 위해 사용되는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 14.

제 3 항에 있어서,

상기 디코더가 피치 추정을 유지하지 않는다면, 상기 추정 단계는 상기 피치 기간 추정을 위한 자동 상관을 사용하고, 상기 디코더가 피치 추정을 유지하면, 상기 디코더의 피치 추정이 사용되는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 15.

제 10 항에 있어서,

상기 오버랩 가산된 길이는 상기 삭제된 프레임들의 길이에 따라 증가하는, 프레임 삭제 은폐(FEC) 실행 방법.

청구항 16.

삭제

청구항 17.

FEC(Frame Erasure Concealment) 능력을 갖지 않는 음성 코더를 위한 프레임 삭제 은폐(FEC)를 실행하는 장치에 있어서,

인코더로부터 전송되는 압축된 음성 정보의 부호화 프레임을 수신하고, 부호화 프레임이 전송 중 소실(missing)되거나, 손실 또는 오염되었는지를 결정하는 손실 프레임 검출기;

상기 손실 프레임 검출기가 상기 부호화 프레임이 삭제되지 않았다고 결정하면, 상기 수신된 부호화 프레임을 복호화 프레임으로 복호화하는 디코더;

상기 손실 프레임 검출기가 상기 부호화 프레임이 삭제되었다고 결정하면, 이전의 복호화 프레임의 데이터를 사용하여 상기 삭제된 프레임을 위한 합성 프레임을 생성하는 FEC 모듈;

상기 복호화 및 합성 프레임의 데이터로 업데이트되는 제 1 메모리; 및

상기 복호화 및 합성 프레임을 오디오 신호로서 출력하는 출력 장치;를 포함하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 18.

제 17 항에 있어서,

지연 메모리를 사용하여 미리 결정된 시간 기간만큼 상기 오디오 신호의 출력을 지연시키는 지연 모듈;을 더 포함하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 19.

제 17 항에 있어서,

하나 또는 그 이상의 삭제된 프레임들의 시퀀스 내의 제 1 삭제된 프레임에서, 상기 FEC 모듈은 상기 삭제된 프레임의 피치 기간을 추정하고, 상기 제 1 메모리로부터 제 2 메모리로 하나 또는 그 이상의 가장 최근 피치 기간들을 복사하고, 상기 제 1 메모리로부터 제 3 메모리로 가장 최근 피치 기간의 부분을 복사하고, 상기 제 2 메모리 내에 저장된 상기 프레임들의 부분으로 상기 제 3 메모리 내에 저장된 상기 피치 기간의 부분을 오버랩 가산하고, 상기 제 2 메모리 내에 저장된 상기 피치 기간의 가장 최근 부분 내에 상기 피치 기간의 오버랩 가산된 부분을 저장하고, 상기 제 1 메모리를 업데이팅하고 상기 출력 장치를 사용하여 출력하기 위하여, 상기 제 2 메모리 내에 저장된 하나 또는 그 이상의 가장 최근 피치 기간들에 기초하여 합성 프레임을 생성하며, 상기 제 2 메모리에서 사용된 최후의 위치를 오프셋 포인터로서 저장하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 20.

제 19 항에 있어서,

상기 FEC 모듈은 상기 오버랩 가산된 부분으로 상기 지연 메모리를 업데이트하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 21.

제 17 항에 있어서,

상기 손실 프레임 검출기는, 합성 프레임이 출력된 후에, 하나 또는 그 이상의 후속하여 부호화된 프레임들이 삭제되었는지를 결정하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 22.

제 21 항에 있어서,

상기 손실 프레임 검출기가, 하나 또는 그 이상의 후속하여 부호화된 프레임들이 삭제되었다고 결정하면, 상기 FEC 모듈은 연속적으로 삭제된 프레임들이 미리 결정된 길이를 초과하는지를 결정하고, 상기 연속적으로 삭제된 프레임들이 미리 결정된 길이를 초과하지 않는다고 결정되면, 상기 합성 프레임을 생성하는데 사용된 피치 기간들의 수를 변경하고, 연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과한다고 결정되면, 현재 피치의 수로 상기 합성 프레임 생성을 지속하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 23.

제 22 항에 있어서,

상기 FEC 모듈이, 연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과하지 않는다고 결정하면, 상기 FEC 모듈은 피치 기간의 부분을 위해 이전의 프레임으로부터 상기 합성 신호의 생성을 지속하고, 그 결과를 제 4 메모리에 배치하고, 제 2 메모리에 사용된 상기 피치 기간들의 수를 증가시키고, 상기 제 2 메모리 내에 저장된 피치 기간의 부분으로 상기 제 3 메모리를 오버랩 가산하고 상기 제 2 메모리의 가장 최근 부분 내에 그 결과를 저장하고, 상기 제 2 메모리의 새로운 부분을 지시할 때까지 피치 기간만큼 오프셋 포인터를 이동시키고, 상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써 합성 프레임의 데이터를 생성하고, 상기 제 5 메모리 내의 피치 기간의 부분으로 상기 제 4 메모리를 오버랩 가산하고 상기 제 5 메모리 내에 그 결과를 저장하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과하는지를 결정하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과한다고 결정되면, 상기 삭제 부분의 길이에 기초하여 상기 제 5 메모리를 감소시키며, 상기 제 5 메모리는 상기 제 1 메모리를 업데이팅하고 상기 출력 장치를 사용하여 출력하기 위해 사용되는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 24.

제 22 항에 있어서,

상기 FEC 모듈이, 연속적으로 삭제된 프레임들의 수가 미리 결정된 길이를 초과한다고 결정하면, 상기 FEC 모듈은 상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써 합성 프레임의 데이터를 생성하고, 상기 제 2 메모리 내에 사용된 최후의 위치를 오프셋 포인터 내에 저장하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과하는지를 결정하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과한다고 결정되면 상기 삭제 부분의 길이에 기초하여 제 5 메모리를 감소시키며, 상기 제 5 메모리는 상기 제 1 메모리를 업데이팅하고 상기 출력 장치를 출력하기 위해 사용되는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 25.

제 21 항에 있어서,

상기 후속하여 부호화된 프레임이 삭제되지 않는다면, 상기 FEC 모듈은 FEC 프로세스를 사용하여 제 1 비삭제 부호화 프레임을 처리하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 26.

제 25 항에 있어서,

상기 FEC 모듈은 상기 제 2 메모리로부터 제 5 메모리로 데이터를 복사함으로써 합성 프레임의 데이터를 생성하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과하는지를 결정하고, 상기 소실된 프레임들이 미리 결정된 길이를 초과한다고 결정되면, 상기 삭제 부분의 길이에 기초하여 상기 제 5 메모리를 감소시키고, 제 1 비삭제 복호화 시작 프레임의 시작부로 상기 제 5 메모리의 부분을 오버랩 가산하고 상기 제 5 메모리 내에 상기 결과를 저장하고, 상기 제 5 메모리로 상기 복호화 프레임의 비오버랩 가산된 영역을 복사하고, 상기 제 5 메모리는 상기 제 1 메모리를 업데이팅하고 상기 출력 장치를 사용하여 출력하기 위해 사용되는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 27.

제 17 항에 있어서,

상기 합성 프레임들을 생성하는데 사용된 피치 기간들의 수는, 시간의 함수로서 증가하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 28.

제 17 항에 있어서,

상기 FEC 모듈은 인코더를 사용하여, 상기 디코더의 상태 변수가 상기 합성 프레임을 추적하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 29.

제 17 항에 있어서,

고주파수 입력 신호들을 위하여, 하나 또는 그 이상의 피치 기간들은 상기 합성 프레임들을 생성하는데 사용되는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 30.

제 19 항에 있어서,

디코더가 피치 추정을 유지하지 않는다면, 상기 FEC 모듈은 상기 피치 기간 추정을 위한 자동 상관을 사용하고, 상기 디코더가 피치 추정을 유지하면, 상기 디코더의 피치 추정이 사용되는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 31.

제 26 항에 있어서,

상기 오버랩 가산된 길이는 상기 소실된 프레임들의 길이에 따라 증가하는, 프레임 삭제 은폐(FEC) 실행 장치.

청구항 32.

삭제

명세서

기술분야

본 정규 출원은 여기서 참고로 포함되고 1999년 4월 19일 출원된 미국 가출원 60/130,016호의 이점을 주장한다. 또한, 다음의 문서들, ITU-T Recommendation G.711 - Appendix 1, "G.711를 갖는 패킷 손실 은폐를 위한 고품질의 저복잡성 알고리즘(A high quality low complexity algorithm for packet loss concealment with G.711)"(9/99) 및 전기 통신에 대한 미국 표준 규격 - ITU-T Recommendation G.711과 함께 사용되는 패킷 손실 은폐(T1.521-1999)는 여기서 참고로 포함된다.

본 발명은 패킷(packet) 손실 또는 프레임 삭제 은폐(Frame Erasure Concealment; FEC)의 실행에 관한 것으로, 특히, G.711 음성 코더와 같이 내장(built-in) 또는 표준 FEC를 갖지 않는 음성 코더를 사용한 FEC의 실행에 관한 것이다.

배경기술

패킷 손실 또는 프레임 삭제 은폐(FEC) 기술들은 입력 신호가 전송기에 부호화되고 패킷화되어 네트워크 상으로 보내져, 프레임을 복호화하고 출력을 플레이 아웃하는 수신기에서 수신되는 오디오 시스템에서의 전송 손실들을 감춘다. IUT-T's G.723.1, G.728, 및 G.729와 같이 많은 표준 CELP(Code Excited Linear Prediction)-기반의 음성 코더들이 그들의 표준들에 내장 또는 제안된 FEC 알고리즘들을 갖는 반면, 예를 들어, G.711을 위한 그러한 표준은 현재 존재하지 않는다.

발명의 상세한 설명

본 발명은 내장 또는 표준 FEC 프로세스를 갖지 않는 음성 코더를 위한 패킷 손실 또는 프레임 삭제 은폐(FEC)를 실행하기 위한 방법 및 장치와 관련된다. 디코더를 갖는 수신기는 인코더로부터 전송된 압축된 음성 정보의 부호화 프레임들을 수신한다. 수신기에서 손실된 프레임 검출기는 부호화 프레임이 전송시에 손실 또는 오염되었거나, 삭제되었는지를 결정한다. 부호화 프레임이 삭제되지 않는다면, 부호화 프레임은 디코더에 의해 복호화되고, 임시 메모리는 디코더의 출력과 함께 업데이트된다. 미리 결정된 지연 기간은 적용되고, 이어서 오디오 프레임이 출력된다.

손실 프레임 검출기가, 부호화 프레임이 소거되었음을 결정하면, FEC 모듈은 상기 신호에 프레임 은폐 프로세스를 적용한다. FEC 프로세싱은 삭제된 프레임을 위하여 자연스러운 사운딩 합성 음성을 생산한다.

본 발명은 음성 코딩 시스템 처리에 대하여 패킷 손실 또는 프레임 삭제 은폐(FEC)를 실행하는 방법 및 장치에 관한 것이다. 부호화 프레임이 삭제될 때, 프레임 은폐 처리는 신호에 적용된다. 이 처리는 손실 음성을 합성하기 위해 피치 파형들의 복제를 채용하지만, 기존의 기술과는 달리, 처리는 그 수가 삭제의 길이와 함께 증가하는 다수의 피치 파형들을 복제한다. 이 FEC 처리는 삭제 프레임들에 대하여 자연스럽게 들리는 합성 음성을 생성함으로써 종래의 기술에서 진일보 하였다.

삭제

실시예

최근에는 POTS(Plain-Old-Telephony Service)를 지원하기 위하여 보장된 서비스 품질 없이 패킷(packet) 네트워크들 상에서 G.711을 사용하는 것에 많은 관심이 있어 왔다. 이들 네트워크들 상에서 프레임 삭제들(또는 패킷 손실들)이 일어날 때, 은폐 기술들이 필요로 되거나 호출의 질이 심각하게 저하된다. 고품질의 저복잡성 프레임 삭제 은폐(Frame Erasure Concealment(FEC)) 기술은 개발되었고, 이하에 상세히 설명된다.

FEC를 갖춘 오디오 시스템의 예시적인 블록도는 도 1에 도시된다. 도 1에서, 인코더(110)는 입력 오디오 프레임을 수신하고, 코드화된 비트-스트림(bit-stream)을 출력한다. 비트-스트림은 프레임들이 손실되었는지를 결정하는 손실 프레임 검출기(115)에 의해 수신된다. 손실 프레임 검출기(115)가 프레임이 손실된 것으로 결정하면, 손실 프레임 검출기(115)는 소실된 프레임들을 재구성하기 위하여 FEC 알고리즘이나 처리를 적용하도록 FEC 모듈(130)에 신호를 전한다.

따라서, FEC 처리는 입력 신호가 전송기에서 부호화되어 패킷화되고, 네트워크를 통해 전달되며, 프레임이 손실되었는지를 결정하는 손실 프레임 검출기(115)에서 수신되는 오디오 시스템에서 전송 손실을 감춘다. 도 1에서는 손실 프레임 검출기(115)가 예상되는 프레임이 사용되기에 너무 늦게 도착되었는지의 여부를 결정하는 방식을 갖는다고 가정한다. IP 네트워크 상에서, 이것은 일반적으로 전송된 프레임 내의 데이터에 시퀀스 넘버나 시간스탬프(timestamp)를 부가함으로서 구현된다. 손실 프레임 검출기(115)는 어떤 프레임들도 도착하는 프레임들의 시퀀스 넘버들을 손실하지 않았다면 예상되었을 수 있는 시퀀스 넘버들과 비교한다. 프레임이 예상될 때, 손실 프레임 검출기(115)가 도착된 것으로 검출하면, 이것은 디코더(120)에 의해 복호화되고 오디오의 출력 프레임이 출력 시스템에 주어진다. 프레임이 손실되면, FEC 모듈(130)은 오디오를 대신하는 합성 프레임을 생성함으로서 소실된 오디오 프레임을 감추는 처리를 적용한다.

G.723.1, G.728, 및 G.729와 같이, 많은 표준 ITU-T CELP-기반의 음성 코더들이 디코더들에서 음성 재생을 모델화한다. 따라서, 디코더들은 디코더에서 직접 FEC 처리를 완성하기에 충분한 상태 정보를 갖는다. 이들 음성 코더들은 그들의 표준들의 일부로서 지정된 FEC 알고리즘들 또는 처리들을 갖는다.

비교에 의해서, G.711은 음성 재생을 모델화하지 않는 샘플-대-샘플(sample by sample) 부호화 구조이다. 코더에는 FEC에서 보조하기 위한 상태 정보가 없다. 그 결과로, G.711를 가진 FEC 처리는 코더와 무관하다.

G.711 코더와 사용되는 시스템의 예시적인 블록도는 도 2에 도시된다. 도 1에서와 같이, G.711 인코더(210)는 비트-스트림 데이터를 부호화하여 손실 프레임 검출기(215)에 전송한다. 다시, 손실 프레임 검출기(215)는 도착 프레임들의 시퀀스 넘버들을 프레임이 손실되지 않았다면 예상될 수 있는 시퀀스의 넘버들과 비교한다. 프레임이 예상될 때 도착하면, 이것은 디코더(220)에 의해 복호화되기 위해 전달되고, 이어서 이력 버퍼(history buffer)(240)로 출력되어 그 신호를 저장한다. 프레임이 손실되면, 손실 프레임 검출기(215)는 오디오를 대신하는 합성 프레임을 생성함으로서 소실된 오디오 프레임을 감추는 처리를 적용하도록 FEC 모듈(230)에 알린다.

그러나, 소실된 프레임들을 감추기 위해, FEC 모듈(230)은 어느 신호가 소실된 프레임에 있어야 하는가를 추정하기 위해 이력 버퍼(240)에 의해 제공되는 복호화 출력 신호의 과거 이력을 사용하는 G.711 FEC 처리를 적용한다. 부가하여, 삭제된 프레임과 삭제되지 않은 프레임 사이의 매끄러운 전이를 보장하도록, 지연 모듈(250)은 또한 소정의 시간 기간, 예를 들면 3.75 msec 만큼 시스템의 출력을 지연시킨다. 이 지연은 합성 삭제 신호가 삭제의 시작시 실제 출력 신호와 천천히 혼합되는 것을 허용한다.

FEC 모듈(230)과 각각의 이력 버퍼(240) 및 지연 모듈(250) 블록 사이의 화살표들은 저장된 이력이 합성 신호를 생성하기 위해 FEC 처리에서 사용됨을 의미한다. 부가하여, FEC 모듈(230)의 출력은 삭제 동안 이력 버퍼(240)를 업데이트하기 위해 사용된다. FEC 처리는 단지 G.711의 복호화 출력에만 의존하므로, 그 처리가 어떤 음성 코더가 존재하지 않을 때도 작용하게 될 것임을 주목하여야 한다.

입력 신호가 FEC 모듈(230)에서 FEC 처리에 의해 처리되는 방법의 그래픽 예가 도 3에 도시된다.

도면에서 상단의 파형은 남성 화자로부터 나온 음성 영역에서 20 msec 삭제가 일어날 때 시스템으로의 입력을 도시한다. 그 아래 파형에서는 FEC 처리가 갭(gap)에 합성 음성을 생성함으로서 소실된 세그먼트들을 은폐하였다. 비교를 위해, 삭제되지 않은 원래의 입력 신호가 또한 도시된다. 이상적인 시스템에서는 은폐된 음성은 원래의 음성과 똑같이 들린다. 도면으로부터 볼 수 있는 바와 같이, 합성 파형은 소실된 세그먼트들에서 원래 파형과 매우 흡사하다. 이후에는 "입력" 파형으로부터 "은폐" 파형이 생성되는 방법이 논의된다.

FEC 모듈(230)에 의해 사용되는 FEC 처리는 이력 버퍼(240)에 저장된 음성에 유사한 특징들을 갖는 합성 음성을 생성함으로서 소실된 프레임을 은폐한다. 기본적인 개념은 다음과 같다. 신호가 발성된 것이면, 신호가 유사 주기적(quasi periodic)이고 국부적으로 정적이라 가정한다. 피치를 추정하고 이력 버퍼(240)에서 최종 피치 기간을 수회 반복한다. 그러나, 삭제가 길거나 피치가 짧으면(주파수가 높으면), 똑같은 피치 기간을 너무 여러 번 반복하는 것은 자연스러운 음성과 비교해 너무 조화적인(harmonic) 출력을 만들게 된다. 삑하는 소리 및 통하는 소리와 같이 들을 수 있는 조화적인 아티팩트들을 방지하기 위해, 이력 버퍼(240)로부터 사용되는 피치 기간들의 수는 삭제의 길이가 증대됨에 따라 증가된다. 짧은 삭제들은 합성 신호를 생성하기 위해 이력 버퍼(240)로부터의 최종 또는 마지막 몇 개의 피치 기간들만을 사용한다. 긴 삭제들은 또한 이력 버퍼(240)에서 더 뒤에 있는 피치 기간들을 사용한다. 긴 삭제들에서, 이력 버퍼(240)로부터의 피치 기간들은 원래 음성에서 일어난 것과 똑같은 순서로 재생되지 않는다. 그러나, 테스트에서, 긴 삭제들에서 생성된 합성 음성 신호가 아직까지 자연스러운 사운드를 만드는 것으로 밝혀졌다.

삭제가 길면 길수록, 합성 신호가 실제 신호에서 멀어질 가능성이 더 높아진다. 특정한 종류의 사운드를 너무 오래 간직함으로써 야기되는 아티팩트들을 방지하기 위해, 합성 신호는 삭제가 더 길어짐에 따라 감쇠된다. 기간 10 msec 이하의 삭제들 동안에는 감쇠가 필요없다. 10 msec 이상의 삭제에서는 합성 신호가 추가 10 msec 당 20%의 비율로 감쇠된다. 60 msec를 넘어서, 합성 신호는 0(묵음)으로 설정된다. 이는 합성 신호가 원래 신호에 너무 비슷하지 않아 평균적으로 60 msec 이후에 소실된 음성을 은폐하도록 계속 시도하는 것은 유리하기 보다 불리해지기 때문이다.

상이한 소스들로부터의 신호들 사이에 전이가 이루어질 때마다, 전이가 클릭들처럼 들을 수 있는 불연속들, 또는 출력 신호로의 부자연스러운 아티팩트들을 도입시키지 않는 것이 중요하다. 이러한 전이는 여러 위치에서 일어난다:

1. 합성 신호의 시작과 최종 양호한 프레임의 끝부분 사이의 경계에서 삭제의 시작시.

2. 합성 신호와 삭제 이후 처음 양호한 프레임 내의 신호의 시작 사이의 경계에서 삭제의 종료시.

3. 이력 버퍼(240)로부터 사용되는 피치 기간들의 수가 신호 변화를 증가시키도록 변화될 때마다.

4. 이력 버퍼(240)의 반복 부분들 사이의 경계들에서.

매끄러운 전이들을 보장하기 위해, 모든 신호 경계들에서는 오버랩 가산(OLA, Overlap Add)이 실행된다. OLA는 한 엣지에서 오버랩되는 두 신호들을 매끄럽게 조합하는 방법이다. 신호들이 오버랩되는 영역에서, 신호들은 윈도우들(window)에 의해 가중화되고, 함께 부가(혼합)된다. 윈도우들은 특정한 샘플에서 가중치의 합이 1이 되도록 설계된다. 즉, 신호들의 전체적인 합에는 이득이나 감쇠가 적용되지 않는다. 부가하여, 윈도우는 신호가 좌측에서 가중치 1로 시작되어 점차 0으로 페이드(fade)되고, 우측에서 가중치 0으로 시작되어 점차 가중치 1로 페이드되도록 설계된다. 따라서, 오버랩 윈도우의 좌측 영역에서는 좌측 신호만이 존재하는 한편, 오버랩 윈도우의 우측 영역에서는 우측 신호만이 존재한다. 오버랩 영역에서, 신호는 점차적으로 좌측 신호에서 우측 신호로 전이된다. FEC 처리에서, 삼각형 윈도우들은 가변 길이 윈도우들 계산하는 복잡성을 낮게 유지하는데 사용되지만, 해닝 윈도우들(Hanning window)와 같은 다른 윈도우들이 대신에 사용될 수 있다.

도 4는 삭제가 끝난 이후 시작되는 실제 음성과 OLA 처리된 20-msec 삭제의 종료 시 합성 음성을 도시한다. 한 예에서, OLA 가중화 윈도우는 5.75 msec의 삼각형 윈도우이다. 상단 신호는 삭제하는 동안 생성된 합성 신호이고, 그 아래의 오버랩 신호는 삭제 이후의 실제 음성이다. OLA 가중화 윈도우들은 신호 아래 도시된다. 여기서는 삭제 동안 실제 신호에서의 피치 변화로 인해, 합성 신호와 실제 신호의 피크들(peak)이 정합되지 않고, OLA 없이 신호들을 조합하도록 시도하는 경우 도입되는 불연속은 "OLA 없이 조합"이라 칭하여진 그래프에 도시된다. "OLA 없이 조합"된 그래프는 OLA 윈도우의 시작까지 합성 신호를 복사하고 지속 기간 동안 실제 신호를 복사함으로서 생성되었다. OLA 동작들의 결과는 경계들에서의 불연속들이 얼마나 매끄러워졌는 가를 도시한다.

앞의 논의는 설명된 처리가 정적 발성된 음성과 작용되는 것에 관심을 두지만, 음성이 신속하게 변하거나 발성된 것이 아니면, 음성은 주기적인 구조를 갖지 않을 수 있다. 그러나, 이들 신호들은 이후 설명될 바와 같이, 똑같은 방법으로 처리된다.

먼저, 본 실시예에서의 피치 추정에서 허가되는 가장 작은 피치는 5 msec이고, 이는 200 Hz의 주파수에 대응한다. 일부 고주파수 여성 및 어린이 화자들이 200 Hz 이상의 기본 주파수를 갖는 것으로 알려져 있으므로, 윈도우가 비교적 크게 유지되도록 이를 200 Hz로 제한한다. 이 방법에서는 10 msec 삭제 프레임 내에서, 선택된 피치 기간이 최대 2회 반복된다. 고주파수 화자인 경우에는 피치 추정기가 실제 피치 기간의 배수를 돌려주기 때문에 이것이 실제로 출력을 저하시키지 않는다. 또한, 음성을 너무 자주 반복시키지 않음으로써, 처리는 비주기적 음성 이외의 합성 주기적 음성을 생성하지 않는다. 두 번째로, 삭제가 더 길어짐에 따라 합성 음성을 생성하는데 사용되는 피치 기간들의 수가 증가되므로, 긴 삭제 동안 주기성이 주어지지 않는 신호에 충분한 변동이 부가된다.

음성의 시간 스케일링(time scaling)을 위한 WSOLA(Waveform Similarity Overlap Add) 처리는 또한 동일한 처리가 주기적 및 비주기적 음성 신호를 모두 시간 스케일 조장하는데 사용될 수 있도록 큰 고정 크기의 OLA 윈도우를 사용함을 주목하여야 한다.

상기에는 FEC 처리의 개요가 주어졌고, 이후에는 각 단계가 상세히 논의된다.

이 논의를 위해, 예를 들면, 프레임은 10 msec의 음성을 포함하고, 샘플링 비율은 8 kHz인 것으로 가정한다. 따라서, 삭제들은 80개 샘플들(8000 * .010 = 80)의 증가로 일어날 수 있다. FEC 처리는 다른 프레임 크기들 및 샘플링 비율들에 쉽게 적용될 수 있음을 주목하여야 한다. 샘플링 비율을 변화시키기 위해서는 단지 msec로 주어진 시간 기간을 .001로 곱하고, 그 샘플링 비율로 적절한 버퍼 크기들을 정한다. 예를 들면, 이력 버퍼(240)는 음성의 마지막 48.75 msec를 포함한다. 8 kHz에서, 이는 버퍼가(48.75 * .001 * 800) = 390 샘플의 길이임을 의미한다. 16 kHz 샘플링에서는 이것이 2배, 즉 780 샘플이 된다.

몇 가지 버퍼 크기는 처리에서 만날 것으로 예상되는 최저 주파수를 기반으로 한다. 예를 들면, 설명되는 처리에서는 8 kHz 샘플링에서 보게 될 최저 주파수가 66 2/3 Hz인 것으로 가정한다. 이는 15 msec(1/(66 2/3) = .015)의 최대 피치 기간을 만든다. 이력 버퍼(240)의 길이는 최저 주파수 기간의 3.25 배이다. 그래서, 이력 버퍼(240)는 15 * 3.25 = 48.75 msec이다. 16 kHz 샘플링에서 입력 필터들이 50 Hz(20 msec 기간) 만큼 낮은 주파수를 허용하면, 이력 버퍼(240)는 20 * 3.25 = 65 msec로 연장되어야 한다.

프레임 크기도 또한 변화될 수 있다; 10 msec는 G.729와 같이 몇 가지 표준 음성 코더들에 의해 사용되는 프레임 크기이므로 디폴트(default)로 선택되었고, 또한 몇 가지 무선 시스템들에서도 사용된다. 프레임 크기를 변화시키는 것은 직접적이다. 원하는 프레임 크기가 10 msec의 배수이면, 처리는 변화지 않고 유지된다. 단순히, 삭제 처리의 프레임 크기를 10 msec로 남겨두고, 이를 프레임 당 여러 번 호출한다. 원하는 패킷 프레임 크기가 5 msec와 같이 10 msec의 약수이면, FEC 처리는 기본적으로 변하지 않고 유지된다. 그러나, 피치 버퍼에서 기간들의 수가 증가되는 비율은 10 msec 내의 프레임들의 수를 기반으로 수정되어야 한다. 12 msec와 같이, 10 msec의 배수들 또는 약수들이 아닌 프레임 크기도 또한 수용될 수 있다. FEC 처리는 합리적으로 피치 버퍼에서 사용되는 피치 기간들의 수에 대한 증가 비율을 변화시키는 것을 용인하고 있다. 10 msec 마다 1회 이외에 12 msec 마다 1회로 기간들의 수를 증가시키는 것은 그다지 차이가 없다.

도 5는 도 2의 실시예에 의해 실행되는 FEC 처리의 블록도이다. 주요 동작 중 일부를 실시하는데 필요한 서브단계들은 7, 도 12, 및 도 16에서 더 상세히 설명되고, 이후 논의된다. 다음 논의에서는 몇 가지 변수가 값 및 버퍼를 유지하는데 사용된다. 이들 변수는 다음과 같이 요약된다:

[표 1]

변수 및 그 내용

변수	타입	설명	코멘트
B	Array	피치 버퍼	범위[-P*3.25:-1]
H	Array	내력 버퍼	범위[-390:-1]
L	Array	최종 1/4 버퍼	범위[-P*.25:-1]
O	Scalar	피치 버퍼내의 오프셋	
P	Scalar	피치 추정	40 <= P < 120
P4	Scalar	1/4 피치 추정	P4 = P >> 2
S	Array	합성된 음성	범위[0:79]
U	Scalar	사용된 파장	1 <= U <= 3

도 5의 흐름도에 도시된 바와 같이, 처리는 단계(505)에서 시작되어, 다음 프레임이 손실 프레임 검출기(215)에 의해 수신된다. 단계(510)에서, 손실 프레임 검출기(215)는 프레임이 삭제되는지를 결정한다. 프레임이 삭제되지 않으면, 단계(512)에서, 프레임은 디코더(220)에 의해 복호화된다. 이어서, 단계(515)에서, 복호화된 프레임은 FEC 모듈(230)에 의해 사용되도록 이력 버퍼(240)에 저장된다.

이력 버퍼 업데이트 단계에서, 이 버퍼(240)의 길이는 예상되는 최장 피치 기간의 길이의 3.25배이다. 8 kHz 샘플링에서, 최장 피치 기간은 15 msec 또는 120 샘플이므로, 이력 버퍼(240)의 길이는 48.75 msec 또는 390 샘플이다. 그러므로, 각 프레임이 디코더(220)에 의해 복호화된 이후에, 이력 버퍼(240)는 최근 음성 이력을 포함하도록 업데이트된다. 이력 버퍼(240)의 업데이트는 도 6에 도시된다. 이 도면에 도시된 바와 같이, 이력 버퍼(240)는 최근 음성 샘플들을 우측에 포함하고 가장 오래된 음성 샘플들을 좌측에 포함한다. 가장 최근 프레임의 복호화 음성이 수신되면, 이는 우측에서 버퍼(240)로 쉬프트되고, 가장 오래된 음성에 대응하는 샘플은 좌측에서 버퍼를 벗어나 쉬프트된다(도 6b를 참고).

부가하여, 단계(520)에서는 지연 모듈(250)이 최장 피치 기간의 1/4 만큼 음성 출력을 지연시킨다. 8 kHz 샘플링에서, 이는 120 * 1/4 = 30 샘플 또는 3.75 msec이다. 이 지연은 FEC 모듈(230)이, FEC 모듈(230)에 의해 생성된 합성 신호와 삭제 이전의 실제 신호 사이의 매끄러운 전이를 보장하기 위하여 삭제의 시작에서 FEC 모듈(230)이 1/4 파장 OLA를 실행하는 것을 허용한다. 출력은 프레임을 복호화한 이후 다음 프레임이 삭제되는지를 알지 못하기 때문에 지연되어야 한다.

단계(525)에서는 오디오가 출력되고, 단계(530)에서는 프레임들이 더 있는가를 결정한다. 프레임이 더 이상 없으면, 처리는 종료된다. 프레임이 더 있으면, 처리는 다음 프레임을 구하도록 단계(505)로 되돌아간다.

그러나, 단계(510)에서, 손실 프레임 검출기(215)가 수신된 프레임이 삭제되는 것으로 결정하면, 처리는 FEC 모듈(230)이 제 1 삭제 프레임을 은폐시키는 단계(535)로 진행되고, 그 처리는 이후 도 7에서 상세히 설명된다. 제 1 프레임이 은폐된 이후에, 단계(540)에서는 손실 프레임 검출기(215)가 다음 프레임을 구한다. 단계(545)에서, 손실 프레임 검출기(215)는 다음 프레임이 삭제되는가 여부를 결정한다. 다음 프레임이 삭제되지 않으면, 단계(555)에서, FEC 모듈(230)은 삭제 이후 제 1 프레임을 처리하고, 그 처리는 이후 도 16에서 상세히 설명된다. 제 1 프레임이 처리된 이후에, 처리는 단계(530)로 복귀되어, 손실 프레임 검출기(215)는 프레임들이 더 있는 지를 결정한다.

단계(545)에서, 손실 프레임 검출기(215)가 다음 또는 이어지는 프레임이 삭제되는 것으로 결정하면, FEC 모듈(230)은 도 12에서 상세히 설명되는 처리에 따라 제 2 및 이어지는 프레임을 은폐한다.

도 7은 처음 10 msec의 삭제를 은폐하도록 취해지는 단계를 설명한다. 이후에는 그 단계가 상세히 조사된다.

도 7에서 볼 수 있는 바와 같이, 단계(705)에서, 삭제의 시작시 제 1 동작은 피치를 추정하는 것이다. 이를 위해, 표준화된 자기-상관관계(auto-correlation)가 40에서부터 120샘플까지의 탭(tap) 지연에서 20 msec(160 샘플) 윈도우로 이력 버퍼(240) 신호 상에서 실행된다. 8 kHz 샘플링에서는 이러한 지연이 5 내지 15 msec의 피치 기간 또는 200에서부터 66 2/3 Hz까지의 기본 주파수에 대응한다. 자기-상관관계의 기간에서 탭은 피치 추정 P이다. H가 이 이력을 포함하고, -1(삭제 이전의 우측 샘플)에서 -390(삭제가 시작되기 이전에 390개 샘플의 샘플)으로 인덱스된다고 가정하면, 탭 j에 대한 자기 상관관계는 수학적으로 다음과 같이 표시될 수 있다:

$$Autocor(j) = \frac{\sum_{i=1}^{160} H[-i]H[-i-j]}{\sqrt{\sum_{k=1}^{160} H^2[-k-j]}}$$

그래서, 자기 상관관계의 피크 또는 피치 추정은 다음과 같이 표시될 수 있다:

삭제

$$P = \{\max_j(Autocor(j)) \mid 40 \le j \le 120\}$$

상술된 바와 같이, 5 msec 또는 40 샘플로 허용된 최저 피치 기간은 단일 피치 기간이 10 msec 삭제 프레임에서 최대 2회 반복되기에 충분하게 크다. 이는 발성되지 않은 음성에서 아티팩트들을 방지하고, 또는 높은 피치의 화자에서 부자연스러운 조화적인 아티팩트들을 방지한다.

도 3의 삭제에 대해 표준화된 자기-상관관계를 계산하는 그래프의 예가 도 8에 도시된다.

"이력(history)"이라 칭하여진 파형은 삭제 직전에 이력 버퍼(240)의 내용이다. 점선 수평선은 신호의 참고 부분, 즉 삭제 직전의 20 msec 음성인 이력 버퍼(240) H[-1]:H[-160]을 도시한다. 직선 수평선은 40 샘플(상단선, 5 msec 기간, 200 Hz 주파수)에서부터 120 샘플(하단선, 15 msec 기간, 66.66 Hz 주파수)까지의 탭에서 지연된 20 msec 윈도우이다. 상관관계의 출력은 또한 윈도우들의 위치들과 정렬되어 그려진다. 상관관계에서 점선 수직선은 곡선의 피크로, 추정된 피치를 나타낸다. 이 선은 삭제의 시작으로부터 한 기간 뒤이다. 이 경우, P는 56 샘플로, 7 msec의 피치 기간 및 142.9 Hz의 기본 주파수에 대응한다.

자기-상관관계의 복잡성을 낮추기 위해, 2가지 특수한 과정들이 사용된다. 이들 숏컷들(shortcut)이 출력을 현저하게 변화시키지 않지만, 처리의 전체적인 운행 시간 복잡성에 큰 영향을 갖는다. FEC 처리에서 대부분의 복잡성은 자기-상관관계에 있다.

먼저, 모든 탭에서 상관관계를 계산하기보다는 추려진 신호에 대략적인 피크의 추정이 먼저 결정되고, 이어서 미세한(fine) 탐색이 대략적인 피크 주변에서 실행된다. 대략적인 추정을 위해, 2:1로 간추려진 신호에 작용하고 모든 다른 탭마다 조사만을 하는 새로운 함수로 상기의 Autocor 함수를 수정한다:

$$Autocor_{rough}(j) = \frac{\sum_{i=1}^{80} H[-2i]H[-2i-j]}{\sqrt{\sum_{k=1}^{80} H^2[-2k-j]}}$$

$$P_{rough} = 2\{\max_j(Autocor_{rough}(2j)) \mid 20 \le j \le 60\}$$

대략적인 추정을 사용하면, 범위 $P_{rough} - 1 \le j \le P_{rough} + 1$에서만 원래의 탐색 처리가 반복된다. j가 40 내지 120 샘플 사이의 원래 범위에 머물도록 확인하는데 주의한다. 샘플링 비율이 증가되면, 데시메이션 인자(decimation factor)도 증가되어, 처리의 전체적인 복잡성이 대략 일정하게 유지됨을 주목한다. 44.1 kHz로 샘플링된 음성에 8:1의 데시메이션 인자들로 테스트가 실행되어 양호한 결과를 얻었다. 도 9는 $Autocor_{rough}$의 그래프를 Autocor의 그래프와 비교한다. 도면에서 볼 수 있는 바와 같이, $Autocor_{rough}$는 Autocor에 양호하게 근접하고, 복잡성은 8 kHz 샘플링에서 거의 4의 계수만큼 - 모든 다른 탭만이 조사되기 때문에 2의 계수, 또한 소정의 탭에서 모든 다른 샘플만이 조사되기 때문에 2의 계수만큼 - 감소된다.

제 2 과정은 Autocor 및 $Autocor_{rough}$에서 에너지 계산의 복잡성을 낮추도록 실행된다. 각 단계에서 전체적인 합을 계산하기보다는 운행되는 에너지의 합이 유지된다. 즉:

$$Energy(j) = \sum_{k=1}^{160} H^2[-k-j]$$

then:

$$Energy(j+1) = \sum_{k=1}^{160} H^2[-k-j-1] = Energy(j) + H^2[-j-161] - H^2[-j-1]$$

그래서, 제 1 에너지 항이 계산된 이후에 FEC 처리의 각 단계에서는 에너지 항을 업데이트하는데 단지 2회 곱산 및 2회 합산만이 필요하다.

이제는 피치 추정 P를 갖고, 파형은 삭제 동안 생성되기 시작한다. 다시 도 7의 흐름도를 참고로, 단계(710)에서, 가장 최근 3.25 파장들(3.25 * P 샘플)은 이력 버퍼(240) H에서 피치 버퍼 B로 복사된다. 피치 버퍼의 내용은 최근 1/4 파장을 제외하고, 삭제 기간 동안 일정하게 유지된다. 한편, 이력 버퍼(240)는 삭제 동안 합성 음성으로 계속하여 업데이트된다.

단계(715)에서, 이력 버퍼(240)로부터의 최근 1/4 파장(.25 * P 샘플)은 최종 1/4 버퍼(last quater buffer) L에 저장된다. 이 1/4 파장은 수회의 OLA 동작들 동안 필요하다. 편의상, 이력 버퍼(240)에서와 같이, B 및 L 버퍼를 액세스하는데는 똑같은 음의 인덱스 구조가 사용될 것이다. B[-1]은 삭제 도착 이전의 최종 샘플이고, B[-2]는 그 이전의 샘플이다. 합성 음성은 0 이상으로 인덱스 처리되는 합성 버퍼 S에 배치될 것이다. 그래서, S[0]은 제 1 합성 샘플이고, S[1]은 제 2 합성 샘플이다.

도 3에서 삭제 동안 피치 버퍼 B 및 최종 1/4 버퍼 L의 내용은 도 10에 도시된다. 이전 섹션에서 기간 P는 56 샘플로 계산되었다. 따라서, 피치 버퍼는 3.25 * 56 = 182 샘플 길이이다. 최종 1/4 버퍼는 .25*56=14 샘플 길이이다. 도면에서, 수직선들은 삭제의 시작으로부터 뒤로 매 P 샘플마다 배치된다.

삭제의 처음 10 msec 동안에는 피치 버퍼로부터 최종 피치 기간만이 사용되므로, 단계(720)에서는 U = 1이다. 음성 신호가 진짜 주기적이고 피치 추정이 단지 추정이 아니라 정확한 진짜 값이면, 파형을 피치 버퍼 B에서 합성 버퍼 S로 직접 복사할 수 있고, 합성 신호는 매끄럽고 연속적이 된다. 즉, S[0] = B[-P]이고, S[1] = B[-P+ 1]이다. 피치가 10 msec 프레임 보다 짧으면, 즉 P < 80이면, 단일 피치 기간은 삭제된 프레임에서 1회 이상 반복된다. 본 예에서는 P = 56이므로, 복사가 S[56]까지 이른다. 샘플 56 부근에서 샘플 대 샘플 복사 시퀀스는: S[54] = B[-2], S[55] = B[-1], S[56] = B[-56], S[57] = B[-55] 등.

실제로, 피치 추정은 정확하지 않고 신호는 진짜 주기적이지 않을 수 있다. (a) 실제 신호와 합성 신호 사이의 경계 및(b) 기간이 반복되는 경계에서 불연속을 방지하기 위해서는 OLA가 요구된다. 두 경계에서는 모두 실제 음성의 종료 B[-1]로부터 한 기간 이전의 음성 B[-P]까지 매끄러운 전이를 원한다. 따라서, 단계(725)에서, 이는 B[-P] 이전의 1/4 파장을 이력 버퍼(240)의 최종 1/4 파장, 또는 L의 내용과 오버랩 가산(OLA)함으로서 이루어질 수 있다. 그래프에서, 이는 피치 버퍼에서 최종 1 1/4 파장을 취하여, 이를 한 파장 우측으로 쉬프트하고, 1/4 파장 오버랩 영역에서 OLA를 실행하는 것과 동일하다. 단계(730)에서, OLA의 결과는 이력 버퍼(240)에서 최종 1/4 파장으로 복사된다. 추가 기간의 합성 파형을 생성하기 위해, 피치 버퍼는 추가 파장만큼 쉬프트되고, 추가 OLA가 실행된다.

도 11은 처음 2회 반복되는 동안의 OLA 동작을 도시한다. 이 도면에서, 모든 파형과 교차하는 수직선은 삭제의 시작부이다. 짧은 수직선은 피치 마커들(marker)이고, 삭제 경계로부터 P 샘플 떨어진 곳에 위치한다. 파형들 "피치 버퍼"와 "P 만큼 우측으로 쉬프트" 사이의 오버랩 영역은 "P 만큼 우측으로 쉬프트"와 "2P 만큼 우측으로 쉬프트" 사이의 오버랩 영역과 정확하게 똑같은 샘플에 대응한다. 그러므로, 1/4 파장 OLA만이 한번 계산될 필요가 있다.

단계(735)에서, OLA를 먼저 계산하고 피치 버퍼의 최종 1/4 파장에 그 결과를 배치함으로서, 진짜 주기적 신호에 대해 합성 파형을 생성하는 처리가 사용될 수 있다. 샘플 B[-P]에서 시작되어, 단순히 피치 버퍼로부터의 샘플들을 합성 버퍼로 복사하고, 피치 버퍼의 끝부분에 이르면, 피치 버퍼 포인터를 다시 피치 기간의 시작으로 보낸다. 이 기술을 사용해, 임의의 기간의 합성 파형이 생성될 수 있다. 도 11의 "OLA로 조합" 파형에서 삭제 시작의 좌측까지의 피치 기간은 피치 버퍼의 업데이트 내용에 대응한다.

"OLA로 조합" 파형은 단일 기간의 피치 버퍼가 불연속들 없이 기간 P인 주기적 신호를 생성함을 설명한다. 이력 버퍼(240)에서 단일 파장으로부터 생성되는 이 합성 음성은 처음 10 msec의 삭제를 은폐하는데 사용된다. OLA의 효과는 "피치 버퍼" 및 "OLA들과의 조합" 파형들에서 삭제가 시작되기 직전의 1/4 파장을 비교함으로서 관찰될 수 있다. 단계(730)에서, "OLA들과의 조합" 파형의 이러한 1/4 파장은 이력 버퍼(240)에서 최종 1/4 파장을 대신한다.

삼각형 윈도우들과의 OLA 동작은 또한 수학적으로 표시될 수 있다. 먼저, 변수 P4는 샘플들에서 피치 기간의 1/4인 것으로 정의된다. 그래서, P4 = P >> 2이다. 본 예에서, P는 56이므로, P4는 14이다. OLA 동작은 범위 $1 \le i \le P4$에서 다음과 같이 표시될 수 있다:

$$B[-i] = \frac{i}{P4} L[-i] + \left(\frac{P4 - i}{P4}\right) B[-i - P]$$

OLA의 결과는 피치 버퍼 및 이력 버퍼(240)에서의 최종 1/4 파장을 모두 대신한다. 이력 버퍼(240)를 대신함으로서, 이력 버퍼(240)는 또한 3.75 msec 만큼 출력을 지연시키므로, 이력 버퍼(240)가 업데이트될 때, 1/4 파장 OLA 전이가 출력될 것이다. 처음 10 msec의 삭제 동안 출력 파형은 도 3의 "은폐" 파형에서 처음 두 점선들 사이의 영역에서 관찰될 수 있다.

단계(740)에서, 프레임에 대한 합성 음성을 생성의 종료 시, 현재 오프셋은 피치 버퍼에 변수 O로 저장된다. 이 오프셋은 합성 파형이 다음 프레임의 실제 또는 합성 신호의 OLA를 위해 다음 프레임으로 계속되는 것을 허용한다. O는 또한 삭제가 10 msec를 넘어 확장되면 적절한 합성 신호 위상이 유지되게 허용한다. 80 샘플 프레임 및 P = 56인 본 예에서, 삭제의 시작 시 오프셋은 -56이다. 56 샘플 이후에, 이는 다시 -56이 된다. 추가 80 - 56 = 24 샘플 이후에, 오프셋은 -56 + 24 = -32이므로, O는 제 1 프레임의 종료 시 -32이다.

단계(745)에서, 합성 버퍼가 S[0]에서부터 S[79]까지 채워진 이후에, S는 이력 버퍼(240)를 업데이트하기 위해 사용된다. 단계(750)에서, 이력 버퍼(240)는 또한 3.75 msec 지연을 부가한다. 이력 버퍼(240)의 취급은 삭제 프레임들 및 비삭제 프레임들 동안 똑같다. 이 지점에서, 도 5의 단계(535)에 대한 제 1 프레임 은폐 동작은 종료되고, 처리는 도 5에서 단계(540)로 진행된다.

도 5의 단계(550)에 도시된 바와 같이, FEC 모듈(230)이 10 msec를 넘어서 나중 프레임들을 은폐하도록 동작하는 방법의 상세한 내용은 도 12에 상세히 도시된다. 제 2 및 나중 삭제 프레임들 동안 합성 신호를 생성하기 위해 사용되는 기술은 비록 신호에 일부 변화를 가하는데 일부 추가 작업이 행해질 필요가 있더라도 제 1 삭제 프레임과 아주 유사하다.

단계(1205)에서, 삭제 코드는 제 2 또는 제 3 프레임이 삭제되는 지를 결정한다. 제 2 및 제 3 삭제 프레임들 동안, 피치 버퍼로부터 사용되는 피치 기간들의 수는 증가된다. 이는 신호에서 더 큰 변화를 만들고, 합성 출력이 너무 조화적으로(harmonic) 발성되는 것을 방지한다. 다른 모든 전이들과 같이, OLA는 피치 기간들의 수가 증가될 때 경계를 평활화할(smooth) 필요가 있다. 제 3 프레임(30 msec의 삭제)을 넘으면, 피치 버퍼는 3 파장의 길이로 일정하게 유지된다. 이 3 파장은 삭제 기간 동안 모두 합성 음성을 생성한다. 그래서, 도 12에서 좌측에 있는 브랜치는 단지 제 2 및 제 3 삭제 프레임에만 취해진다.

다음 단계(1210)에서는 피치 버퍼에서 사용되는 파장의 수가 증가된다. 즉, U = U+ 1로 설정된다.

제 2 또는 제 3 삭제 프레임이 시작될 때, 단계(1215)에서는 이전 프레임으로부터의 합성 신호가 추가 1/4 파장 동안 현재 프레임의 시작까지 계속된다. 예를 들면, 제 2 프레임이 시작될 때, 본 실시예의 합성 신호는 도 13에 도시된 바와 같이 나타난다. 이 1/4 파장은 피치 버퍼로부터 더 오래된 파장을 사용하는 새로운 합성 신호와 오버랩 가산된다.

제 2 삭제 프레임이 시작될 때, 파장의 수는 2로 증가되어, U = 2가 된다. 1 파장 피치 버퍼와 같이, OLA는 2-파장 피치 버퍼가 그 자체를 반복하는 경계에서 실행된다. 피치 버퍼 B의 끝에서 뒤로 U 파장들에서 끝나는 1/4 파장인 이 시간은 단계(1220)에서 최종 1/4 버퍼 L의 내용과 오버랩 가산된다. 이 OLA 연산자는 범위 $1 \le i \le P4$에서 다음과 같이 표시될 수 있다:

$$B[-i] = \frac{i}{P4} L[-i] + \left(\frac{P4 - i}{P4}\right) B[-i - PU]$$

이전 버전의 식과 다른 점은 단지 우변에서 B를 인덱스 처리하는데 사용되는 상수 P가 PU로 변환된 것이다. 2-파장 피치 버퍼의 생성은 도 14에 그래프로 도시된다.

도 11에서와 같이, 삭제 시작의 우측에 있는 "OLA로 조합" 파형의 영역은 2-기간 피치 버퍼의 업데이트 내용이다. 짧은 수직선은 피치 기간을 나타낸다. "OLA로 조합" 파형에서 연속적인 피크의 면밀한 조사는 피크가 삭제 시작 이전에 뒤로 1 및 2 파장을 피크에서 교체하는 것으로 나타낸다.

제 2 프레임에서 합성 출력이 시작될 때, 새로운 피치 버퍼로부터의 신호를 도 13에서 생성된 1/4 파장과 병합시켜야 한다. 새로운 피치 버퍼로부터의 합성 신호는 사용되는 버퍼 중 가장 오래된 부분으로부터 전해지는 것이 바람직하다. 그러나, 새로운 부분은 유사한 부분의 파형으로부터 주어짐을 주의하여야 한다. 그렇지 않으면, 이들을 혼합할 때, 들을 수 있는 아티팩트들이 생성될 것이다. 다른 말로 하면, 정확한 위상을 유지하는 것을 원하지만, 이들을 혼합할 때, 파형이 파과적으로 간섭될 수 있다.

이는 단계(1225)(도 12)에서 피치 버퍼의 사용 부분에서 가장 오래된 파장을 나타낼 때까지 이전 프레임의 종료 시 저장된 오프셋 O에서 기간 P를 감산함으로서 이루어진다.

예를 들면, 제 1 삭제 프레임에서, 피치 버퍼 B에 대해 유효한 인덱스는 -1에서부터 -P까지였다. 그래서, 제 1 삭제 프레임으로부터 저장된 O는 이 범위 내에 있어야 한다. 제 2 삭제 프레임에서, 유효 범위는 -1에서부터 -2P까지이다. 그래서, O가 범위 $-2P \le O < -P$에 있을 때까지 O로부터 P를 감산한다. 또는, 보다 일반적으로, 범위 $-UP \le O < -(U-1)P$에 있을 때까지 O로부터 P를 감산한다. 본 실시예에서는 제 1 삭제 프레임의 종료 시 P = 56이고 O = -32이다. -32에서 56을 감산하면, -88이 산출된다. 따라서, 제 2 프레임에서 제 1 합성 샘플은 B[-88]로부터 주어지고, 다음은 B[-87]로부터 주어진다.

제 2 삭제 프레임의 시작 시 1 및 2 기간 피치 버퍼로부터 합성 신호를 OLA 혼합하는 것이 도 15에 도시된다.

O에서 P를 감산함으로서, 적절한 파형 위상이 유지되고, "1P 피치 버퍼" 및 "2P 피치 버퍼" 파형에서 신호의 피크가 정렬됨을 주목하여야 한다. "OLA 조합" 파형은 또한 제 2 삭제 프레임의 시작 시 다른 피치 버퍼 사이의 매끄러운 전이를 도시한다. 도 15의 "OLA 조합" 파형에서 제 2 프레임이 출력될 수 있기 이전에 1회 이상의 동작이 요구된다.

단계(1230)(도 12)에서는 피치 버퍼로부터의 1/4 파장을 임시 버퍼에 복사하기 위해 새로운 오프셋이 사용된다. 단계(1235)에서, 1/4 파장은 오프셋에 가산된다. 이어서, 단계(1240)에서, 임시 버퍼는 출력 버퍼의 시작과 OLA 동작되고, 그 결과는 출력 버퍼의 처음 1/4 파장에 배치된다.

단계(1245)에서, 오프셋은 출력 버퍼에서 신호의 나머지를 생성하는데 사용된다. 피치 버퍼는 10 msec 프레임의 지속 기간 동안 출력 버퍼에 복사된다. 단계(1250)에서는 현재 오프셋이 피치 버퍼에 변수 O로 저장된다.

제 2 및 나중의 삭제 프레임들 동안, 단계(1255)에서는 합성 신호가 선형 기울기(linear ramp)로 감쇠된다. 합성 신호는 60 msec를 넘어서 0으로 설정되거나 묵음이 될 때까지 점차적으로 페이드(fade)된다. 삭제가 길어질수록, 은폐된 음성은 진짜 신호로부터 발산될 가능성이 더 높다. 사운드가 짧은 시간 기간 동안 고립되어 자연스럽게 들리더라도, 특정한 종류의 사운드를 너무 오래 간직하면, 은폐 처리의 출력에서 부자연스러운 가청(unnatural audible) 아티팩트들이 만들어질 수 있다. 합성 신호에서 이들 아티팩트들을 방지하기 위해서는 저속 페이드 아웃(fade out)이 사용된다. G.723.1, G.728, 및 G,729와 같이, 모든 표준 음성 코더들에서 발견되는 은폐 처리들에는 유사한 동작이 실행된다.

FEC 처리는 제 2 프레임에서 시작되어, 10 msec 프레임 당 20%로 신호를 감쇠한다. S, 즉 합성 버퍼가 감쇠 이전의 합성 신호를 포함하고, F가 연속적인 삭제 프레임의 수이면(제 1 삭제 프레임에 대해 F = 1, 제 2 삭제 프레임에 대해 F = 2), 감쇠는 다음과 같이 표시될 수 있다:

$$S'[i] = [1 - .2(F-2) - \frac{.2i}{80}]S[i]$$

범위 $0 \le i \le 79$에서 $2 \le F \le 6$이다. 예를 들어, 제 2 삭제 프레임의 시작 시 샘플에서 F = 2이므로, F-2 = 0이고, .2/80 = .0025이므로, S'[0] = 1.S[0], S'[1] = 0.9975S[1], S'[2] = 0.995S[2], 또한 S'[79] = 0.8025S[79]이다. 제6 삭제 프레임을 넘으면, 출력은 간단히 0으로 설정된다.

단계(1255)에서 합성 신호가 감쇠된 이후에, 이는 단계(1260)에서 이력 버퍼(240)로 주어지고, 단계(1265)에서는 출력이 3.75 msec 만큼 지연된다. 오프셋 포인터 O는 또한 제 2 프레임의 종료 시 피치 버퍼에서 그 위치로 업데이트되므로, 합성 신호는 다음 프레임에서 계속될 수 있다. 이어서, 처리는 다음 프레임을 구하도록 단계(540)로 되돌아간다.

삭제가 2 프레임을 넘어 계속되면, 제 3 프레임에 대한 처리는 피치 버퍼에서 기간들의 수가 1에서 2 대신에 2에서 3으로 증가되는 것을 제외하고 제 2 프레임에서와 똑같다. 본 예의 삭제는 2 프레임에서 종료되지만, 제 3 프레임 이상에서 사용되는 3-기간 피치 버퍼가 도 17에 도시된다. 제 3 프레임을 넘으면, 피치 버퍼에서 기간들의 수는 3으로 고정되어 유지되므로, 도 12의 우측에 있는 경로만이 취해진다. 이 경우, 오프셋 포인터 O는 간단히 피치 버퍼를 합성 출력에 복사하는데 사용되고, 오버랩 가산 동작들은 필요하지 않다.

삭제 이후 양호한 제 1 프레임에서 FEC 모듈(230)의 동작은 도 16에서 설명된다. 삭제의 종료 시, 삭제 동안 생성된 합성 음성과 실제 음성 사이에는 매끄러운 전이가 필요하다. 삭제가 단지 1 프레임 길이이면, 단계(1610)에서, 1/4 파장에 대한 합성 음성이 계속되고 실제 음성과의 오버랩 가산이 실행된다.

단계(1620)에서 삭제가 10 msec 보다 긴 것으로 FEC 모듈(230)이 결정하면, 합성 신호와 실제 신호 사이의 부정합 가능성이 더 커지므로, 단계(1630)에서는 합성 음성 생성이 계속되고, OLA 윈도우는 최대 10 msec까지 삭제 프레임 당 추가 4 msec 만큼 증가된다. 피치의 추정이 약간 벗어나거나, 실제 음성의 피치가 삭제 동안 변화되었으면, 합성 신호와 실제 신호 사이의 위상 부정합 가능성은 삭제 길이에 따라 증가된다. 더 긴 OLA 윈도우들은 더 천천히 합성 신호가 페이드 아웃(fade out)되게 하고, 실제 음성 신호가 페이드 인(fade in)되게 한다. 삭제가 10 msec 보다 더 길면, OLA가 실행되기 이전에 단계(1640)에서 이전 프레임에서 신호의 레벨과 정합하여 합성 신호를 감쇠시키는 것이 필요하다.

단계(1650)에서, OLA는 새로운 입력 프레임의 시작 시 출력 버퍼(합성 음성)의 내용에 실행된다. 입력 버퍼의 시작은 OLA의 결과와 대치된다. 상기의 예에서 삭제의 종료 시 OLA는 도 4에서 관찰될 수 있다. 상기 예에 대한 은폐 처리의 완전한 출력은 도 3의 "은폐" 파형에서 관찰될 수 있다.

단계(1660)에서, 이력 버퍼는 입력 버퍼의 내용으로 업데이트된다. 단계(1670)에서, 음성의 출력은 3.75 msec 만큼 지연되고, 처리는 다음 프레임을 구하도록 도 5의 단계(530)로 복귀한다.

작은 조정으로, FEC 처리는 G.726과 같이, 샘플들 또는 프레임들 사이의 상태 정보를 유지하고 은폐를 제공하지 않는 다른 음성 코더에 적용될 수 있다. FEC 처리는 삭제 동안 합성 파형을 생성하는데 이전 섹션에서 설명된 것과 똑같이 처리된다. 그러나, 코더의 내부 상태 변수가 FEC 처리에 의해 생성된 합성 음성을 추적하게 보장하도록 주의되어야 한다. 그렇지 않으면, 삭제가 끝난 이후, 디코더가 에러있는 상태를 사용하여 재시작될 때 출력에 아티팩트들 및 불연속들이 나타나게 된다. 삭제의 종료 시 OLA 윈도우가 돕지만, 더 실행되어야 한다.

도 18에 도시된 바와 같이, 삭제 기간 동안 디코더(1820)를 인코더(1860)로 변환하고, FEC 모듈(1830)의 합성 출력을 인코더(1860)의 입력으로 사용함으로서 더 나은 결과들이 구해질 수 있다.

이 방법으로, 디코더(1820)의 변수들의 상태가 은폐된 음성을 추적할 것이다. 전형적인 인코더와 다르게, 인코더(1860)는 단지 상태 정보만을 유지하기 위해 구동되고 그 출력은 사용되지 않음을 주목하여야 한다. 그래서, 구동 시간 복잡성을 현저히 낮추도록 이 방법이 취해질 수 있다.

상술된 바와 같이, 본 발명에 의해 제공되는 많은 이점 및 특성이 있다. 특별히, 프레임 삭제가 진행됨에 따라, 합성 신호를 생성하도록 신호 이력으로부터 사용되는 피치 기간들의 수는 시간의 함수로 증가된다. 이는 오랜 삭제들에서 조화적인 아티팩트들을 현저히 감소시킨다. 피치 기간들이 원래 순서로 재생되지 않더라도, 출력은 자연스럽게 들린다.

G.726, 및 샘플들 또는 프레임들 사이에 상태 정보를 유지하는 다른 코더들에서, 디코더는 은폐 처리의 합성 출력의 출력 시 인코더로 작용될 수 있다. 이 방법으로, 디코더의 내부 상태 변수들이 출력을 추적하게 되어, 삭제가 끝난 이후에 디코더에서 에러가 있는 상태의 정보에 의해 야기되는 불연속들을 방지 - 적어도 감소 - 시키게 된다. 인코더로부터의 출력은 결코 사용되지 않으므로(단지 상태 정보를 유지할 목적이므로), 간소화된 저복잡성 버전의 인코더가 사용될 수 있다.

본 실시예들에서 허용되는 최소 피치 기간(40 샘플, 또는 200 Hz)은 일부 여성 및 어린이 화자에 대해 예상되는 기본 주파수 보다 더 크다. 따라서, 고주파수 화자의 경우에는 1 이상의 피치 기간이 삭제의 시작 시에도 합성 신호를 생성하는데 사용된다. 높은 기본 주파수의 화자인 경우, 파형은 더 자주 반복된다. 합성 신호의 다중 피치 기간은 조화적인 아티팩트들을 덜 가능하게 한다. 이 기술은 또한 중단과 같은 신속한 전이의 영역뿐만 아니라, 발성된 것이 아닌 음성 세그먼트들 동안 신호가 자연스럽게 들리는데 도움을 준다.

삭제 이후 양호한 제 1 프레임의 종료 시 OLA 윈도우는 삭제의 길이를 포함하여 커진다. 삭제들이 더 길어지면, 다음 양호한 프레임이 도착할 때, 위상 정합들이 일어날 가능성이 더 높다. OLA 윈도우를 삭제의 함수로 연장시키는 것은 긴 삭제에서 위상 부정합에 의해 생성되는 순간적인 이상을 감소시키지만, 삭제가 짧은 경우 신호가 신속하게 회복되는 것을 허용한다.

본 발명의 FEC 처리는 또한 1/4 파장인 추정 피치의 부분이고 피치 피크들과 정렬되지 않는 가변 길이의 OLA 윈도우를 사용한다.

본 발명의 FEC 처리는 발성된 것과 발성된 것이 아닌 음성 사이를 구별하지 않는다. 그 대신에, 다음의 2가지 처리 특성 때문에 발성되지 않은 음성을 재생하는데 잘 실행된다:(A) 최소 윈도우 크기는 음성 중 발성된 것이 아닌 영역이 적당한 변화를 갖더라도 적절하게 크고, (B) 처리가 진행됨에 따라, 피치 버퍼의 길이가 증가되어, 조화적인 아티팩트들이 생기지 않도록 보장한다. 이미 공지된 시간-스케일링 기술 WSOLA에서도 발성된 음성 및 발성되지 않은 음성을 다르게 처리하는 것을 방지하는데 큰 윈도우들을 사용하는 방법이 주어짐을 주목하여야 한다.

삭제의 시작 시 OLA를 허용하는 지연을 부가하는 것이 본 발명의 처리 중 바람직하지 않은 특성이라 생각될 수 있지만, 삭제의 시작 시 실제 신호와 합성 신호 사이에 매끄러운 전이를 보장하는 것이 필요하다.

본 발명은 상술된 특정 실시예와 연관되어 설명되었지만, 당업자에게는 많은 변경들, 수정, 및 변화들이 명백한 것으로 밝혀진다. 따라서, 상술된 본 발명의 바람직한 실시예는 제한되는 것이 아니라, 설명하는 것으로 의도된다. 다음 청구항에서 정의되는 바와 같이, 본 발명의 의도 및 범위에서 벗어나지 않고 다양한 변화가 이루어질 수 있다.

도면의 간단한 설명

도 1은 예시적인 오디오 전송 시스템을 도시하는 도면.

도 2는 G.711 코더 및 FEC 모듈을 갖춘 예시적인 오디오 전송 시스템을 도시하는 도면.

도 3은 FEC 기술을 사용한 출력 오디오 신호를 도시하는 도면.

도 4는 삭제의 종료 시 오버랩-가산(overlap-add; OLA) 동작을 설명하는 도면.

도 5는 G.711 코더를 사용하여 FEC를 실행하는 예시적인 처리의 흐름도.

도 6은 이력 버퍼(history buffer)의 업데이트 처리를 도시하는 그래프.

도 7은 신호의 제 1 프레임을 은폐하는 예시적인 처리의 흐름도.

도 8은 자기 상관(auto-correlation)으로부터 피치(pitch) 추정을 도시하는 도면.

도 9는 미세한(fine) 피치 추정 대 대략적인(coarse) 피치 추정을 도시하는 도면.

도 10은 피치 및 최종 1/4(lastquarter) 버퍼에서의 신호들을 도시하는 도면.

도 11은 단일-기간 피치 버퍼를 사용한 합성 신호 생성을 도시하는 도면.

도 12는 제 2 또는 신호의 최종 삭제 프레임을 은폐하는 예시적인 처리의 흐름도.

도 13은 제 2 삭제 프레임에 이어지는 합성 신호들을 도시하는 도면.

도 14는 2-기간 피치 버퍼를 사용한 합성 신호 생성을 도시하는 도면.

도 15는 제 2 삭제 프레임의 시작에서 OLA를 도시하는 도면.

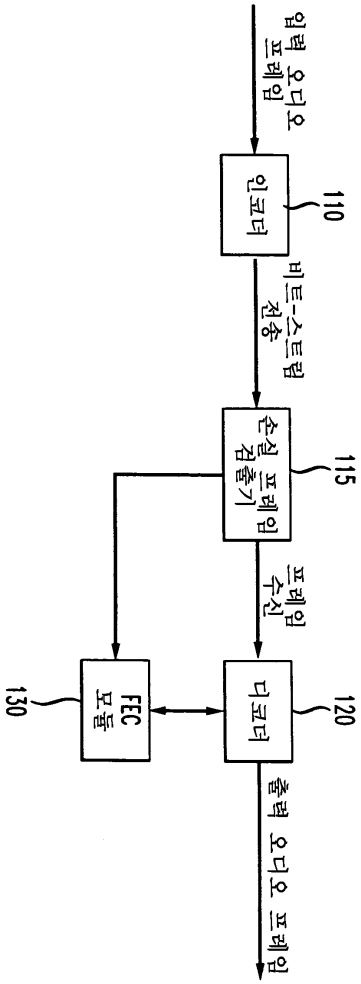
도 16은 삭제 이후에 제 1 프레임을 처리하는 예시적인 방법의 흐름도.

도 17은 3-기간 피치 버퍼를 사용한 합성 신호 생성을 도시하는 도면.

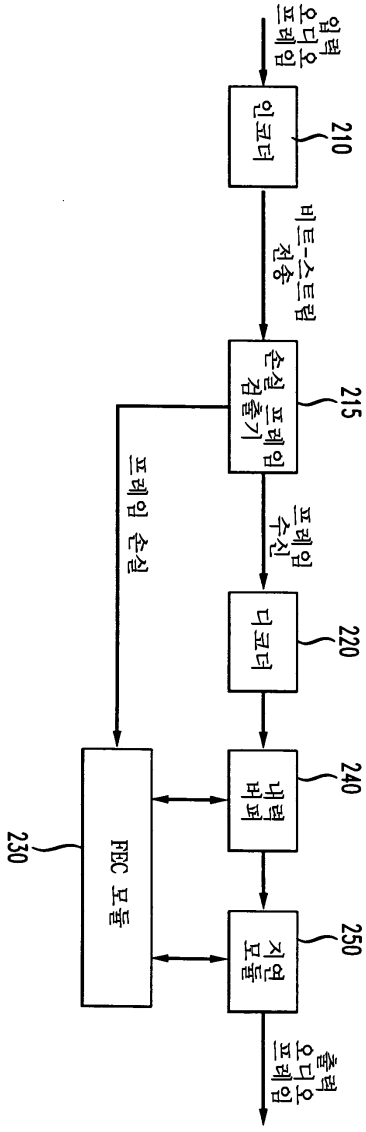
도 18은 다른 음성 코더들로 FEC 기술들을 사용하는 것을 도시하는 블록도.

도면

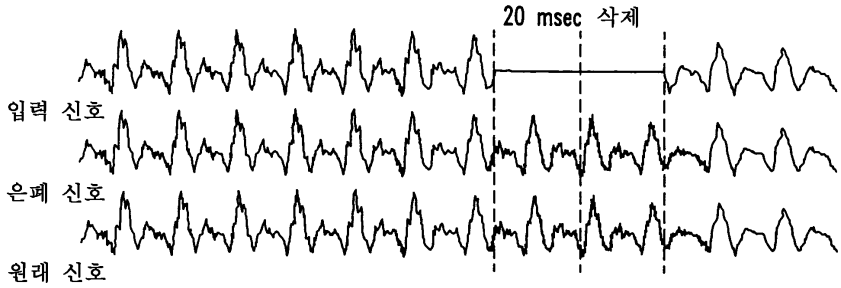
도면1


입력 오디오 프레임
인코더
110
비트-스트림 전송
손실 프레임 검출기
115
프레임 수신
FEC 모듈
130
디코더
120
출력 오디오 프레임

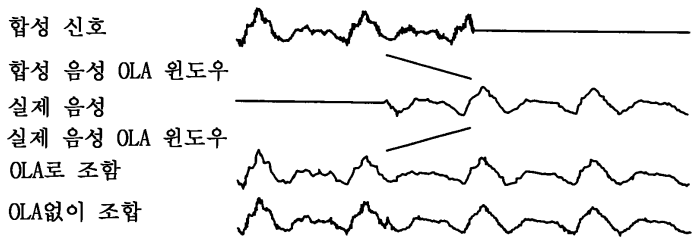
도면2


입력 오디오 프레임
인코더
210
비트-스트림 전송
손실 프레임 검출기
215
프레임 손실
프레임 수신
디코더
220
내력 버퍼
240
FEC 모듈
230
지연 모듈
250
출력 오디오 프레임

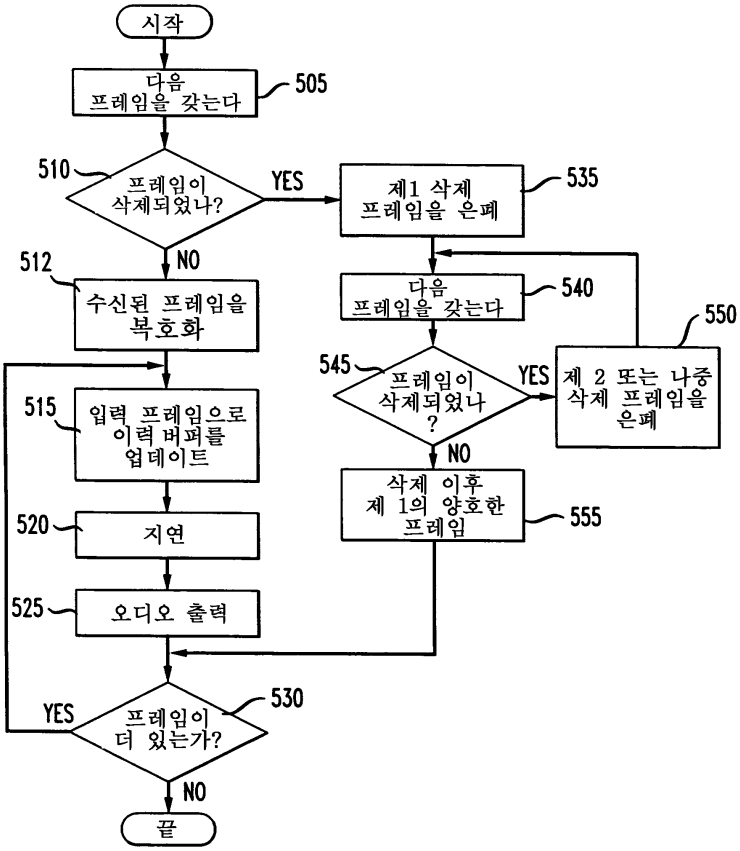
도면3


20 msec 삭제
입력 신호
은폐 신호
원래 신호

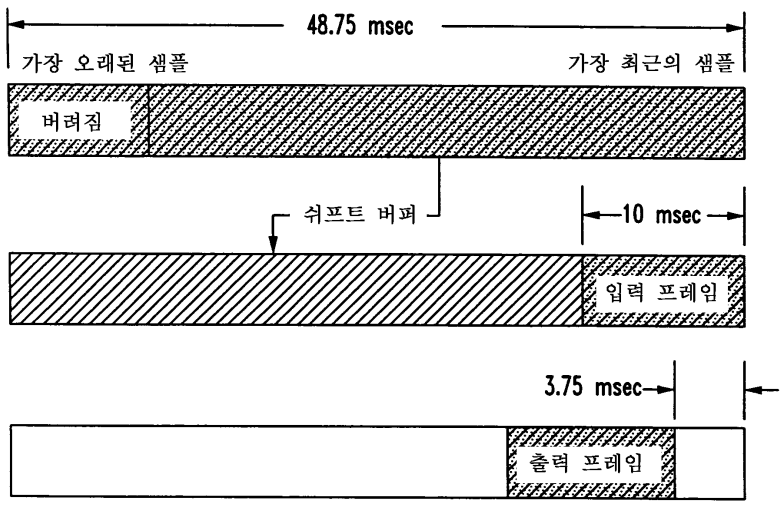
도면4


합성 신호
합성 음성 OLA 윈도우
실제 음성
실제 음성 OLA 윈도우
OLA로 조합
OLA없이 조합

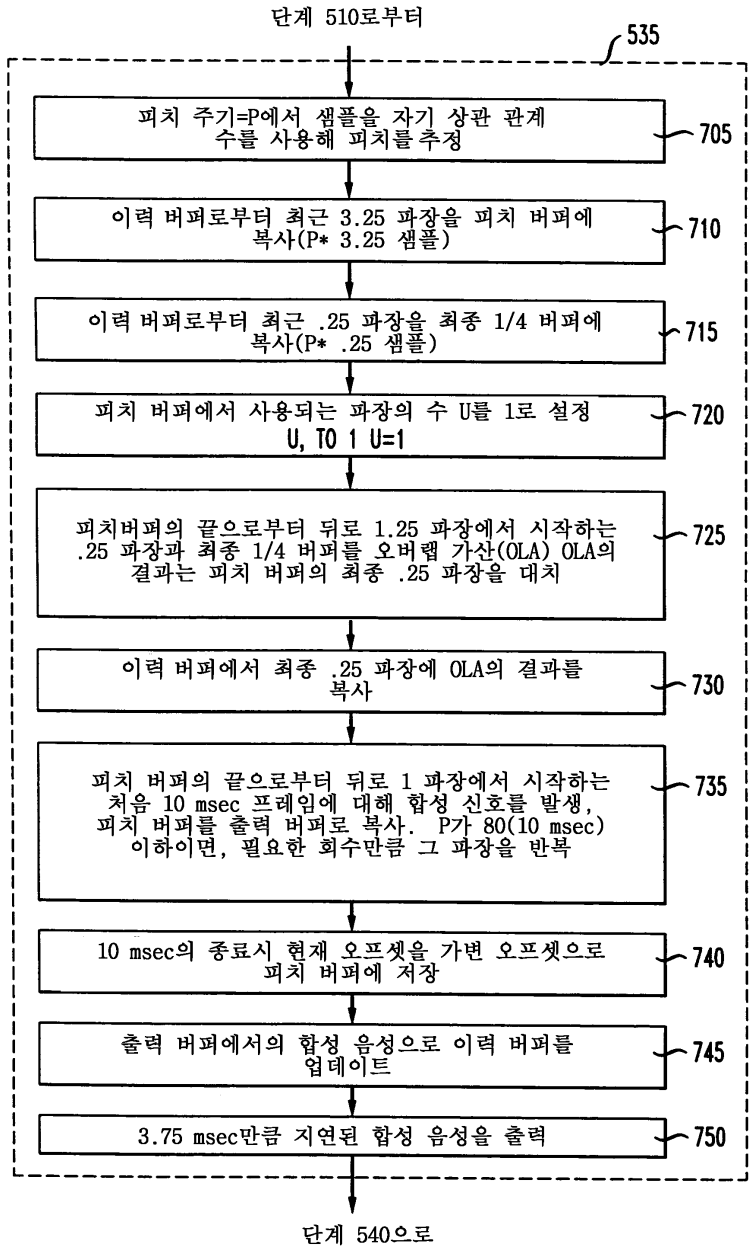
도면5


시작
다음 프레임을 갖는다
505
프레임이 삭제되었나?
510
YES
제1 삭제 프레임을 은폐
535
NO
512
수신된 프레임을 복호화
다음 프레임을 갖는다
540
550
545
프레임이 삭제되었나?
YES
제 2 또는 나중 삭제 프레임을 은폐
NO
515
입력 프레임으로 이력 버퍼를 업데이트
삭제 이후 제 1의 양호한 프레임
555
520
지연
525
오디오 출력
YES
프레임이 더 있는가?
530
NO
끝

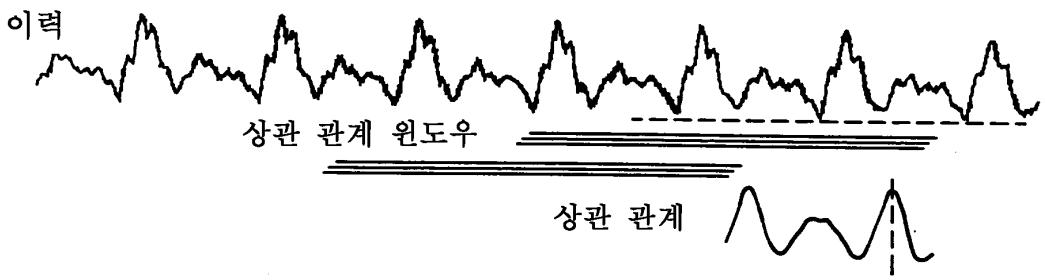
도면6


48.75 msec
가장 오래된 샘플
가장 최근의 샘플
버려짐
쉬프트 버퍼
10 msec
입력 프레임
3.75 msec
출력 프레임

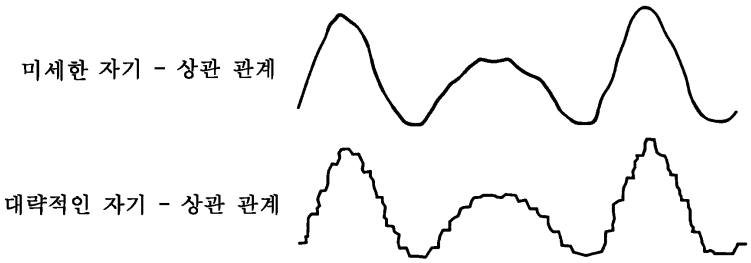
도면7


단계 510로부터
535
피치 주기=P에서 샘플을 자기 상관 관계 수를 사용해 피치를 추정
705
이력 버퍼로부터 최근 3.25 파장을 피치 버퍼에 복사(P* 3.25 샘플)
710
이력 버퍼로부터 최근 .25 파장을 최종 1/4 버퍼에 복사(P* .25 샘플)
715
피치 버퍼에서 사용되는 파장의 수 U를 1로 설정
U, TO 1 U=1
720
피치버퍼의 끝으로부터 뒤로 1.25 파장에서 시작하는 .25 파장과 최종 1/4 버퍼를 오버랩 가산(OLA) OLA의 결과는 피치 버퍼의 최종 .25 파장을 대치
725
이력 버퍼에서 최종 .25 파장에 OLA의 결과를 복사
730
피치 버퍼의 끝으로부터 뒤로 1 파장에서 시작하는 처음 10 msec 프레임에 대해 합성 신호를 발생, 피치 버퍼를 출력 버퍼로 복사. P가 80(10 msec) 이하이면, 필요한 회수만큼 그 파장을 반복
735
10 msec의 종료시 현재 오프셋을 가변 오프셋으로 피치 버퍼에 저장
740
출력 버퍼에서의 합성 음성으로 이력 버퍼를 업데이트
745
3.75 msec만큼 지연된 합성 음성을 출력
750
단계 540으로

도면8


이력
상관 관계 윈도우
상관 관계

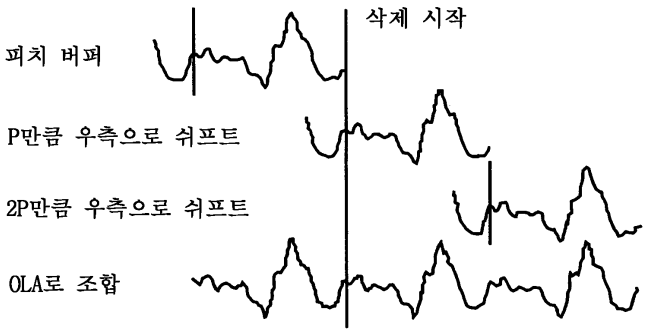
도면9



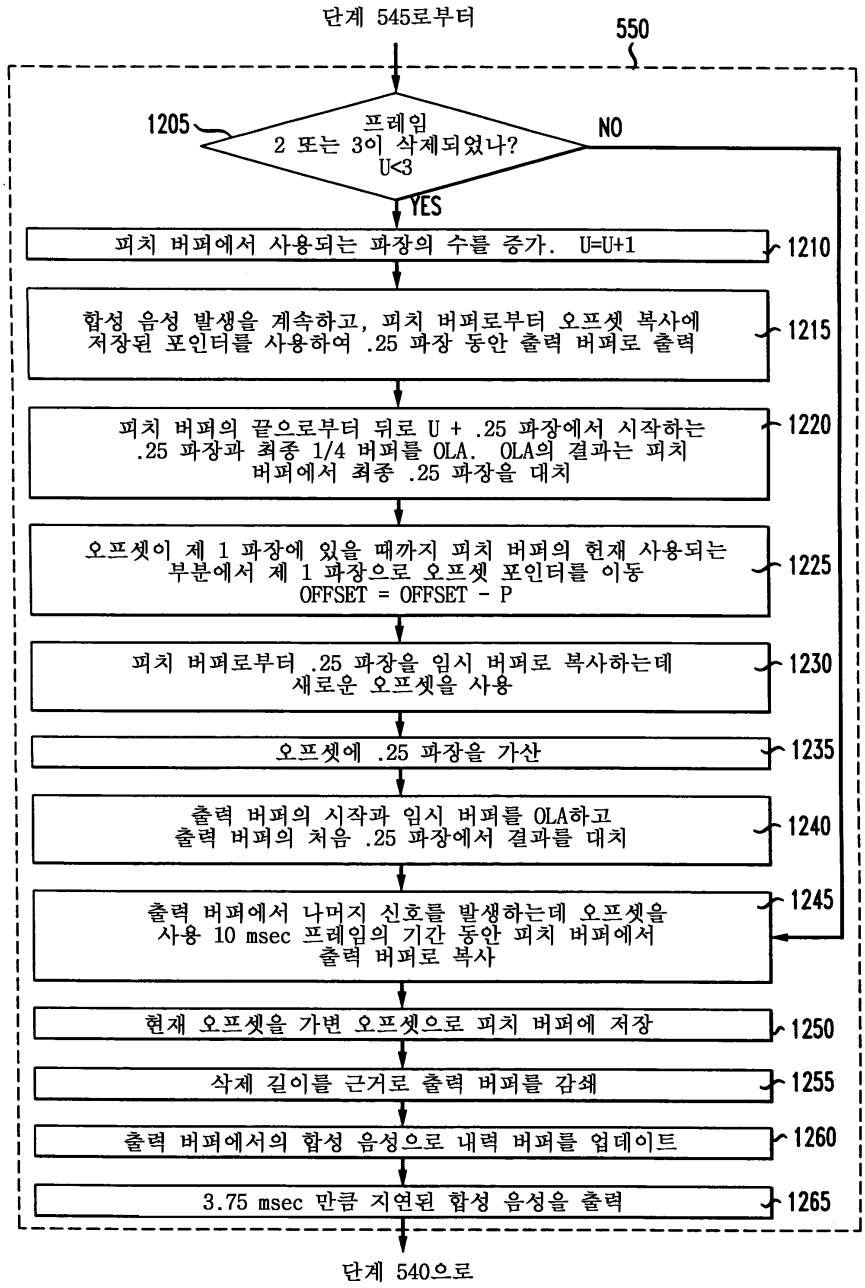
도면10



도면11



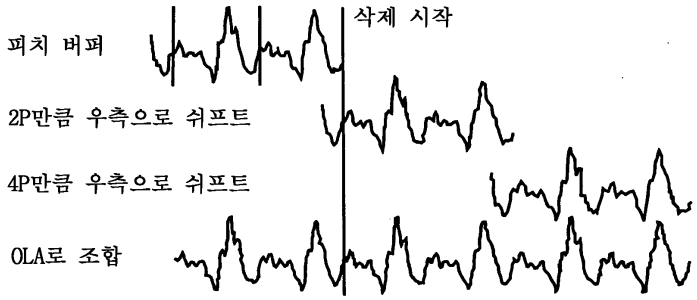
도면12


단계 545로부터
550
1205
프레임
2 또는 3이 삭제되었나?
U<3
NO
YES
피치 버퍼에서 사용되는 파장의 수를 증가. U=U+1
1210
합성 음성 발생을 계속하고, 피치 버퍼로부터 오프셋 복사에
저장된 포인터를 사용하여 .25 파장 동안 출력 버퍼로 출력
1215
피치 버퍼의 끝으로부터 뒤로 U + .25 파장에서 시작하는
.25 파장과 최종 1/4 버퍼를 OLA. OLA의 결과는 피치
버퍼에서 최종 .25 파장을 대치
1220
오프셋이 제 1 파장에 있을 때까지 피치 버퍼의 현재 사용되는
부분에서 제 1 파장으로 오프셋 포인터를 이동
OFFSET = OFFSET - P
1225
피치 버퍼로부터 .25 파장을 임시 버퍼로 복사하는데
새로운 오프셋을 사용
1230
오프셋에 .25 파장을 가산
1235
출력 버퍼의 시작과 임시 버퍼를 OLA하고
출력 버퍼의 처음 .25 파장에서 결과를 대치
1240
출력 버퍼에서 나머지 신호를 발생하는데 오프셋을
사용 10 msec 프레임의 기간 동안 피치 버퍼에서
출력 버퍼로 복사
1245
현재 오프셋을 가변 오프셋으로 피치 버퍼에 저장
1250
삭제 길이를 근거로 출력 버퍼를 감쇄
1255
출력 버퍼에서의 합성 음성으로 내력 버퍼를 업데이트
1260
3.75 msec 만큼 지연된 합성 음성을 출력
1265
단계 540으로

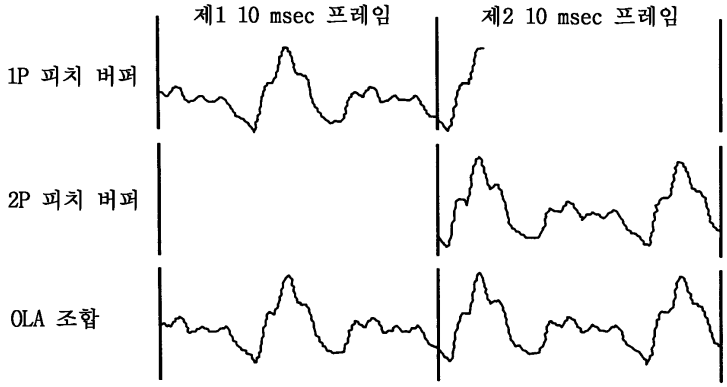
도면13


제1 10 msec 프레임
제2 10 msec 프레임

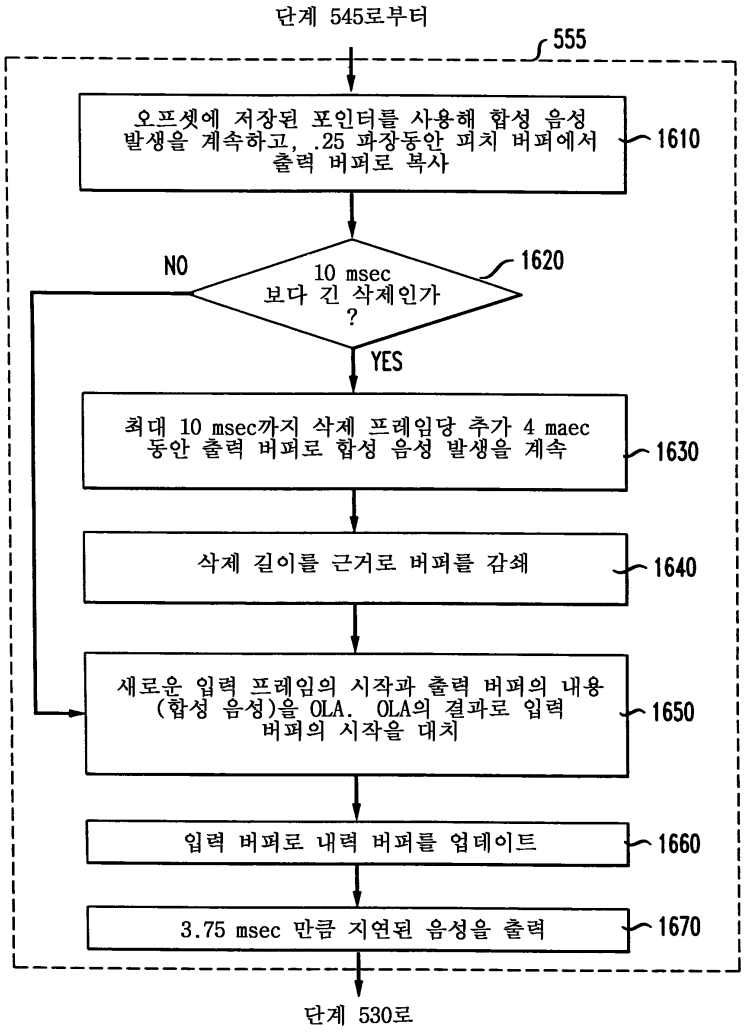
도면14


삭제 시작
피치 버퍼
2P만큼 우측으로 쉬프트
4P만큼 우측으로 쉬프트
OLA로 조합

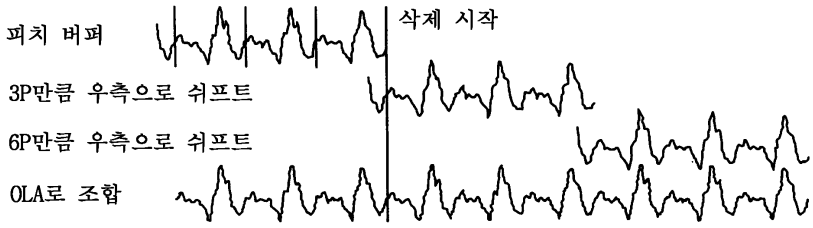
도면15


제1 10 msec 프레임
제2 10 msec 프레임
1P 피치 버퍼
2P 피치 버퍼
OLA 조합

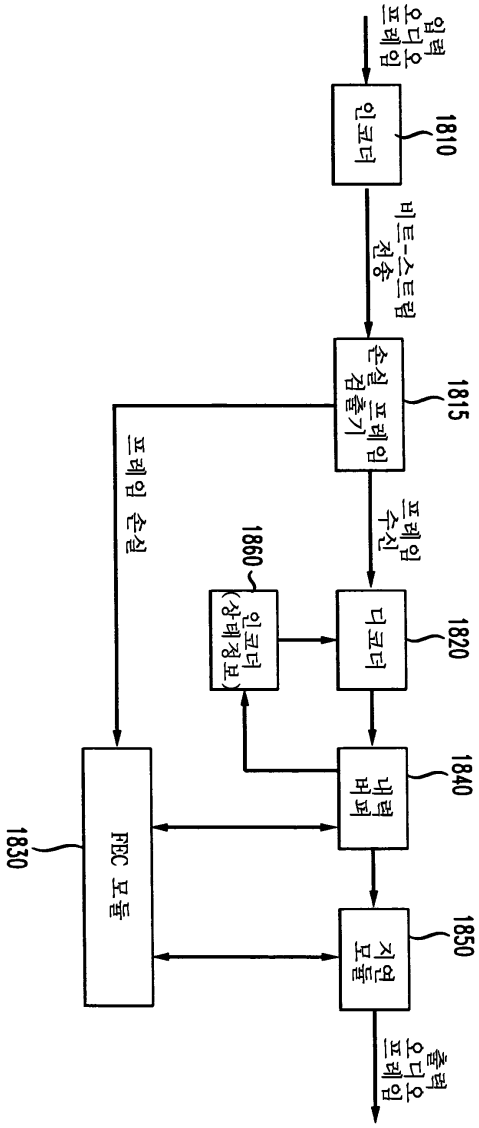
도면16


단계 545로부터
555
오프셋에 저장된 포인터를 사용해 합성 음성 발생을 계속하고, .25 파장동안 피치 버퍼에서 출력 버퍼로 복사
1610
NO
10 msec 보다 긴 삭제인가 ?
1620
YES
최대 10 msec까지 삭제 프레임당 추가 4 maec 동안 출력 버퍼로 합성 음성 발생을 계속
1630
삭제 길이를 근거로 버퍼를 감쇄
1640
새로운 입력 프레임의 시작과 출력 버퍼의 내용 (합성 음성)을 OLA. OLA의 결과로 입력 버퍼의 시작을 대치
1650
입력 버퍼로 내력 버퍼를 업데이트
1660
3.75 msec 만큼 지연된 음성을 출력
1670
단계 530로

도면17


피치 버퍼
삭제 시작
3P만큼 우측으로 쉬프트
6P만큼 우측으로 쉬프트
OLA로 조합

도면18


입력 오디오 프레임
인코더
1810
비트-스트림 전송
손실 프레임 검출기
1815
프레임 손실
프레임 수신
1860
인코더 (상태정보)
디코더
1820
내력 버퍼
1840
FEC 모듈
1830
지연 모듈
1850
출력 오디오 프레임