



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년08월01일
 (11) 등록번호 10-1425290
 (24) 등록일자 2014년07월24일

(51) 국제특허분류(Int. Cl.)
 G10L 19/02 (2006.01)
 (21) 출원번호 10-2012-7011268
 (22) 출원일자(국제) 2010년10월06일
 심사청구일자 2012년04월30일
 (85) 번역문제출일자 2012년04월30일
 (65) 공개번호 10-2012-0063543
 (43) 공개일자 2012년06월15일
 (86) 국제출원번호 PCT/EP2010/064917
 (87) 국제공개번호 WO 2011/042464
 국제공개일자 2011년04월14일
 (30) 우선권주장
 61/249,774 2009년10월08일 미국(US)
 (56) 선행기술조사문헌
 KR1020070107615 A
 전체 청구항 수 : 총 26 항

(73) 특허권자
 프라운호퍼 게젤샤프트 쭈르 프뢰데롱 데어 안겐
 반텐 포르슈 에. 베.
 독일 80686 뮌헨 한자슈트라세 27 체
 (72) 발명자
 누엔도르프, 막스
 독일 90408 뉘른베르크, 크레이링스트라세 15
 푸호스, 구일라우메
 독일 90409 뉘른베르크, 파크스트라세 12
 (뒷면에 계속)
 (74) 대리인
 특허법인이상

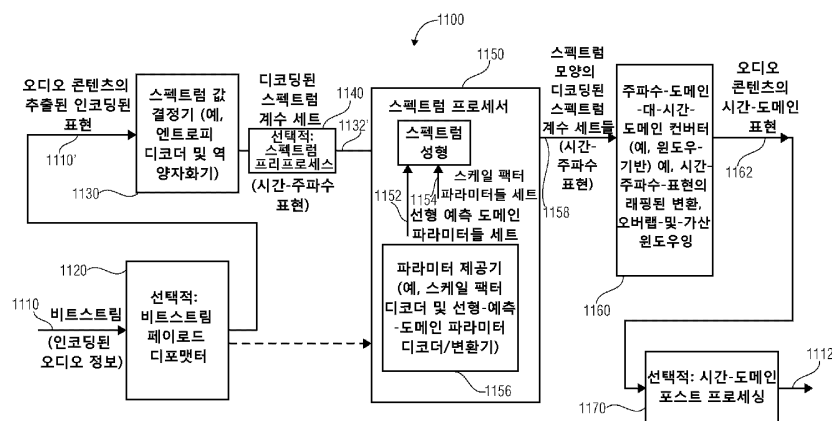
심사관 : 고재용

(54) 발명의 명칭 **멀티-모드 오디오 신호 디코더, 멀티-모드 오디오 신호 인코더 및 선형-예측-코딩 기반의 노이즈 성형을 사용하는 방법 및 컴퓨터 프로그램**

(57) 요약

오디오 콘텐츠의 인코딩된 표현에 기초하여 오디오 콘텐츠의 디코딩된 표현을 제공하는 멀티-모드 오디오 신호 디코더는 오디오 콘텐츠의 복수의 부분들에 대하여 디코딩된 스펙트럼 계수 세트들을 획득하도록 구성되는 스펙트럼 값 결정기를 포함한다. 오디오 신호 디코더는 또한 스펙트럼 프로세서를 포함하며, 스펙트럼 프로세서는 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라, 디코딩된 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하고, 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들 세트에 따라, 디코딩된 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하도록 구성된다. 오디오 신호 디코더는 또한 주파수-도메인-대-시간-도메인 컨버터를 포함하며, 주파수-도메인-대-시간-도메인 컨버터는 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 오디오 콘텐츠의 시간-도메인 표현을 획득하고, 상기 주파-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하도록 구성된다. 오디오 신호 인코더가 또한 기술된다.

대표도



(72) 발명자

레텔바흐, 니콜라우스

독일 90427 뉘른베르크, 스페스사르트스트라쎄 38

백스트로엠, 톰

독일 90443 뉘른베르크, 바우어가쎄 8-12

레콤테, 예레미

독일 90489 뉘른베르크 술츠바허 스트라쎄 39

헤레, 위르겐

독일 91054 부켤호프 할러슈트라쎄 24

특허청구의 범위

청구항 1

오디오 콘텐츠의 인코딩된 표현(1110;1208)에 기초하여 상기 오디오 콘텐츠의 디코딩된 표현(1112;1212)을 제공하는 멀티-모드 오디오 신호 디코더(1110;1200)에 있어서,

상기 오디오 콘텐츠의 복수의 부분들(1410, 1412, 1414, 1416)에 대하여 디코딩된 스펙트럼 계수(1132; 1230d; $r[i]$) 세트들(1132;1230d)을 획득하도록 구성되는 스펙트럼 값 결정기(1130; 1230a, 1230c);

스펙트럼 프로세서(1230e; 1378)로서, 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라, 디코딩된 스펙트럼 계수 세트(1132; 1230d; $r[i]$) 혹은, 그 프리-프로세싱된 버전(1132')에 스펙트럼 성형(shaping)을 적용하고, 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분(1410; 1416)에 대한 스케일 팩터 파라미터들(1152; 1260b) 세트에 따라, 디코딩된 스펙트럼 계수 세트(1132; 1230d; $r[i]$) 혹은, 그 프리-프로세싱된 버전(1132')에 스펙트럼 성형(shaping)을 적용하도록 구성되는, 상기 스펙트럼 프로세서(1230e; 1378); 및

주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)로서, 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트(1158; 1230f)에 기초하여 오디오 콘텐츠의 시간-도메인 표현(1162; 1232; $x_{j,m}$)을 획득하고, 상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 상기 오디오 콘텐츠의 시간-도메인 표현(1162;1232)을 획득하도록 구성되는, 상기 주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)를 포함하는 멀티-모드 오디오 신호 디코더.

청구항 2

청구항 1에 있어서,

상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분의 시간-도메인 표현을 상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분과 함께 오버랩-및-가산하도록 구성되는 오버래퍼(1233)를 더 포함하는 멀티-모드 오디오 신호 디코더.

청구항 3

청구항 2에 있어서,

상기 주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)는 래핑된 변환(lapped transform)을 이용하여 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분(1412; 1414)에 대한 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하고, 래핑된 변환을 사용하여 상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분(1410; 1416)에 대한 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하도록 구성되고,

상기 오버래퍼는 각기 다른 모드로 인코딩된 상기 오디오 콘텐츠의 연속하는 부분들의 시간-도메인 표현들을 오버래핑하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 4

청구항 3에 있어서,

상기 주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)는 동일한 변환 타입의 래핑된 변환들을 적용하여 각자 다른 모드로 인코딩된 상기 오디오 콘텐츠의 부분들에 대한 상기 오디오 콘텐츠의 시간-도메인 표현들을 획득하도록 구성되고; 그리고

상기 오버래퍼는 각자 다른 모드로 인코딩된 상기 오디오 콘텐츠의 연속하는 부분들의 시간-도메인 표현을 오버랩-및-가산하여 상기 래핑된 변환에 의해 야기된 시간-도메인 에일리어싱(aliasing)이 감소되거나 제거되도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 5

청구항 4에 있어서,

상기 오버래핑은, 연관된 래핑된 변환에 의해 제공되는 바와 같은, 제1 모드로 인코딩된 오디오 콘텐츠의 제1 부분(1414)의 윈도우잉된 시간-도메인 표현 또는 그것의 진폭-스케일되었지만 스펙트럼 왜곡되지 않은 버전을 오버랩-및-가산하고, 연관된 래핑된 변환에 의해 제공되는 바와 같은, 제2 모드로 인코딩된 오디오 콘텐츠의 제2 연속하는 부분(1416)의 윈도우잉된 시간-도메인 표현 또는 그것의 진폭-스케일되었지만 스펙트럼 왜곡되지 않은 버전을 오버랩-및-가산하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 6

청구항 1에 있어서,

상기 주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)는 각기 다른 모드로 인코딩된 상기 오디오 콘텐츠의 부분들(1410, 1412, 1414, 1416)의 시간-도메인 표현들을 제공하여 상기 제공된 시간-도메인 표현들이 동일한 도메인에 있도록 하되, 이들이, 상기 제공된 시간-도메인 표현 중의 하나 또는 둘 다에, 윈도우잉 전이 연산을 제외한 신호 성형 필터링 연산을 적용하지 않고, 상기 동일한 도메인 내에서 선형으로 결합 가능하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 7

청구항 1에 있어서,

상기 주파수-도메인-대-시간-도메인 컨버터(1160; 1230g)는 역 변경된 이산 코사인 변환을 수행하여, 상기 역 변경된 이산 코사인 변환의 결과로서, 오디오 신호 도메인 내의 상기 오디오 콘텐츠의 시간-도메인 표현을 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분 및 상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분 양쪽에 대하여 획득하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 8

청구항 1에 있어서,

디코딩된 선형-예측-코딩 필터 계수(α_1 내지 α_{16})를 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 선형-예측-코딩 필터 계수들의 인코딩된 표현에 기초하여 획득하도록 구성되는 선형-예측-코딩 필터 계수 결정기;

상기 디코딩된 선형-예측-코딩 필터 계수들(1260d; α_1 내지 α_{16})을 스펙트럼 표현(1260f; $X_0[k]$)으로 변환하여, 다른 주파수들과 연관된 선형-예측-모드 이득 값들($g[k]$)을 획득하도록 구성되는 필터 계수 변환기(1260e);

상기 디코딩된 스케일 팩터 값들(1260f)을 상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 상기 스케일 팩터 값들의 인코딩된 표현(1254)에 기초하여 획득하도록 구성되는 스케일 팩터 결정기(1260a);를 포함하고,

상기 스펙트럼 프로세서(1150; 1230e)는 스펙트럼 변경기를 포함하고, 상기 스펙트럼 변경기는 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 연관된 디코딩된 스펙트럼 계수들 세트(1132; 1230d; $r[i]$)나, 그 프리-프로세싱된 버전을, 상기 선형-예측-모드 이득 값과 결합하여, 상기 디코딩된 스펙트럼 계수들의 이득-프로세싱된 버전(1158; 1230f; $rr[i]$)을 획득하되, 상기 디코딩된 스펙트럼 계수들(1132; 1230d; $r[i]$)이나, 상기 프리-프로세싱된 버전의 기여분(contribution)에는 상기 선형-예측-모드 이득 값들($g[k]$)에 따라 가중치가 적용되도록 구성되고, 또한,

상기 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 연관된 디코딩된 스펙트럼 계수들 세트(1132; 1230d; $x_{ac_invquant}$)나, 그 프리-프로세싱된 버전을, 상기 스케일 팩터 값들(1260b)과 결합하여, 상기 디코딩된 스펙트럼 계수들($x_{ac_invquant}$)의 스케일-팩터-프로세싱된 버전(x_{rescal})을 획득하되, 상기 디코딩된 스펙트럼 계수들이나, 상기 프리-프로세싱된 버전의 기여분에는 상기 스케일 팩터 값들에 따라 가중치가 적용되도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 9

청구항 8에 있어서,

상기 필터 계수 변환기(1260e)는, 선형-예측-코딩 필터의 시간-도메인 임펄스 응답($\hat{w}[n]$)을 표현하는, 상기 디코딩된 선형-예측-코딩 필터 계수들(1260d)을, 오드(odd) 이산 푸리에 변환을 사용하여 스펙트럼 표현($X_0[k]$)으로 변환하도록 구성되고; 그리고

상기 필터 계수 변환기(1260e)는 상기 선형-예측-모드 이득 값들($g[k]$)을 상기 디코딩된 선형-예측-코딩 필터 계수들(1260d; a_1 내지 a_{16})의 상기 스펙트럼 표현($X_0[k]$)으로부터 도출하여, 상기 이득 값들이 상기 스펙트럼 표현($X_0[k]$)의 계수들($X_0[k]$)의 크기 함수가 되도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 10

청구항 8에 있어서,

상기 필터 계수 변환기(1260e)와 상기 스펙트럼 프로세서(1230e)는 주어진 디코딩된 스펙트럼 계수($r[i]$)나, 그 프리-프로세싱된 버전의, 상기 주어진 디코딩된 스펙트럼 계수의 이득-프로세싱된 버전($rr[i]$)으로의 기여분이 상기 주어진 디코딩된 스펙트럼 계수($r[i]$)와 연관된 선형-예측-모드 이득 값($g[k]$)의 크기에 의해 결정되도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 11

청구항 1에 있어서,

상기 스펙트럼 프로세서(1230e)는 주어진 디코딩된 스펙트럼 계수($r[i]$)나, 그 프리-프로세싱된 버전의, 상기 주어진 디코딩된 스펙트럼 계수의 이득-프로세싱된 버전($rr[i]$)으로의 기여분의 가중치가 상기 주어진 디코딩된 스펙트럼 계수($r[i]$)와 연관된 선형-예측-모드 이득 값($g[k]$)의 크기가 증가함에 따라 증가하거나, 또는 주어진 디코딩된 스펙트럼 계수($r[i]$)나, 그 프리-프로세싱된 버전의, 상기 주어진 디코딩된 스펙트럼 계수의 이득-프로세싱된 버전($rr[i]$)으로의 기여분의 가중치가 상기 디코딩된 선형-예측-코딩 필터 계수들의 스펙트럼 표현의 연관된 스펙트럼 계수($X_0[k]$)의 크기가 증가함에 따라 감소하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 12

청구항 1에 있어서,

상기 스펙트럼 값 결정기(1130; 1230a, 1230c)는 역 양자화를 디코딩된 양자화 스펙트럼 계수들에 적용하여, 디코딩되고 역 양자화된 스펙트럼 계수들(1132; 1230d)을 획득하도록 구성되고; 그리고

상기 스펙트럼 프로세서(1230e)는 주어진 디코딩된 스펙트럼 계수($r[i]$)에 대한 효과적인 양자화 단계를 상기 주어진 디코딩된 스펙트럼 계수($r[i]$)와 연관된 선형-예측-모드 이득 값($g[k]$)의 크기에 따라 조정함으로써 양자화 노이즈 성형을 수행하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 13

청구항 1에 있어서, 상기 오디오 신호 디코더는,

중간 선형-예측-모드 시작 프레임(1212)을 사용하여 주파수-도메인 모드 프레임(1410)으로부터 결합된 선형-예측 모드/대수-코드-여기된(excited) 선형-예측 모드 프레임으로 전이하도록 구성되고,

상기 오디오 신호 디코더는 상기 선형-예측 모드 시작 프레임에 대한 디코딩된 스펙트럼 계수들 세트를 획득하고,

스펙트럼 성형을 상기 선형-예측 모드 시작 프레임에 대한 상기 디코딩된 스펙트럼 계수들 세트나, 그 프리-프로세싱된 버전에, 그와 연관된 선형-예측-도메인 파라미터들 세트에 따라 적용하고,

상기 선형-예측 모드 시작 프레임의 시간-도메인 표현을 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 획득하고,

상대적으로 긴 좌측 전이 슬로프를 갖고 상대적으로 짧은 우측 전이 슬로프를 갖는 시작 윈도우를 상기 선형-예측 모드 시작 프레임의 시간-도메인 표현에 적용하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 14

청구항 13에 있어서,

상기 오디오 신호 디코더는 상기 선형 예측-모드 시작 프레임(1412)을 선행하는 주파수-도메인 모드 프레임(1410)의 시간-도메인 표현의 우측 부분을 상기 선형 예측-모드 시작 프레임의 시간-도메인 표현의 좌측 부분과 오버래핑하여, 시간-도메인 에일리어싱의 감소 또는 제거를 획득하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 15

청구항 13에 있어서,

상기 오디오 신호 디코더는 선형 예측-모드 시작 프레임(1412)과 연관된 선형 예측 도메인 파라미터들을 사용하여, 상기 선형 예측-모드 시작 프레임을 후속하는 상기 결합된 선형-예측 모드/대수-코드-여기된 선형 예측 모드 프레임의 적어도 한 부분을 인코딩하는 대수-코드-여기된 선형 예측 모드 디코더를 초기화하도록 구성되는, 멀티-모드 오디오 신호 디코더.

청구항 16

오디오 콘텐츠의 입력 표현(110; 310; 1010)에 기초하여 오디오 콘텐츠의 인코딩된 표현(112; 312; 1012)을 제공하는 멀티-모드 오디오 신호 인코더(100; 300; 900; 1000)에 있어서,

상기 오디오 콘텐츠의 입력 표현(110; 310; 1010)을 프로세싱하여, 상기 오디오 콘텐츠의 주파수-도메인 표현(122; 330b; 1030b)을 획득하도록 구성되는 시간-도메인-대-주파수-도메인 컨버터로서, 상기 주파수-도메인 표현(122)은 스펙트럼 계수들의 세트들의 시퀀스를 포함하는 상기 시간-도메인-대-주파수-도메인 컨버터(120; 330a; 350a; 1030a);

스펙트럼 프로세서(130; 330e; 350d; 1030e)로서, 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트(134; 340b)에 따라 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하여 스펙트럼 계수들의 스펙트럼-성형된 세트(132)를 획득하도록 하고, 상기 주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들(136) 세트에 따라 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하여 스펙트럼 계수들의 스펙트럼-성형된 세트(132)를 획득하도록, 구성되는 상기 스펙트럼 프로세서(130; 330e; 350d; 1030e); 및

양자화 인코더(140; 330g; 330i; 350f; 350h; 1030g, 1030i)로서, 상기 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 부분에 대한 스펙트럼 계수들의 스펙트럼-성형된 세트(132; 350e; 1030f)의 인코딩된 버전(142; 322, 342; 1032)을 제공하고, 상기 주파수-도메인 모드로 인코딩되는 상기 오디오 콘텐츠의 부분에 대한 스펙트럼 계수들의 스펙트럼-성형된 세트(132; 330f; 1030f)의 인코딩된 버전(142; 322, 342; 1032)을 제공하도록 구성되는, 상기 양자화 인코더(140; 330g; 330i; 350f; 350h; 1030g; 1030i)를 포함하는 멀티-모드 오디오 신호 인코더.

청구항 17

청구항 16에 있어서,

상기 시간-도메인-대-주파수-도메인 컨버터(120; 330a; 350a; 1030a)는 오디오 신호 도메인 내의 오디오 콘텐츠의 시간-도메인 표현(110; 310; 1010)을 상기 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분과 상기 주파수-도메인 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분 양쪽에 대한 상기 오디오 콘텐츠의 주파수-도메인 표현(122; 330b; 1030b)으로 전환하도록 구성되는, 멀티-모드 오디오 신호 인코더.

청구항 18

청구항 16에 있어서,

상기 시간-도메인-대-주파수-도메인 컨버터(120; 330a; 350a; 1030a)는 동일한 변환 타입의 래핑된 변환을 적용하여 각자 다른 모드로 인코딩되는 상기 오디오 콘텐츠의 부분들에 대한 주파수-도메인 표현들을 획득하도록 구성되는, 멀티-모드 오디오 신호 인코더.

청구항 19

청구항 16에 있어서,

상기 스펙트럼 프로세서(130; 330e; 350d; 1030e)는 상기 선형-예측 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 상관관계-기반 분석을 사용하여 획득된 선형-예측 도메인 파라미터들의 세트(134; 340b)에 따라, 또는 상기 주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 음향심리 모델 분석(330c; 1070a)를 사용하여 획득된 스케일 팩터 파라미터들의 세트(136; 330d; 1070b)에 따라, 스펙트럼 계수들의 상기 세트나, 또는 그 프리-프로세싱된 버전에 상기 스펙트럼 성형을 선택적으로 적용하도록 구성되는, 멀티-모드 오디오 신호 인코더.

청구항 20

청구항 19에 있어서,

상기 오디오 신호를 분석하여 상기 오디오 콘텐츠의 한 부분이 상기 선형-예측 모드 또는 상기 주파수-도메인 모드로 인코딩되는지 여부를 결정하도록 구성되는 모드 선택기를 포함하는 멀티-모드 오디오 신호 인코더.

청구항 21

청구항 16에 있어서,

상기 멀티-모드 오디오 신호 인코더는, 주파수-도메인 모드 프레임 및 결합된 변환-코딩된-여기(excitation) 선형-예측 모드/대수-코드-여기된 선형 예측 모드 프레임 사이에 선형-예측 모드 시작 프레임으로서 존재하는, 오디오 프레임을 인코딩하도록 구성되고,

상기 멀티-모드 오디오 신호 인코더는

상대적으로 긴 좌측 전이 슬로프와 상대적으로 짧은 우측 전이 슬로프를 갖는 시작 윈도우를 상기 선형-예측 모드 시작 프레임의 시간-도메인 표현에 적용하여, 윈도우잉된 시간-도메인 표현을 획득하고,

상기 선형 예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 주파수-도메인 표현을 획득하고,

상기 선형-예측 모드 시작 프레임에 대한 선형-예측 도메인 파라미터들의 세트를 획득하고,

상기 선형-예측 도메인 파라미터들 세트에 따라 상기 선형-예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전에 스펙트럼 성형을 적용하고,

상기 선형-예측 도메인 파라미터의 세트와 상기 선형-예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 상기 스펙트럼 성형된 주파수 도메인 표현을 인코딩하도록 구성되는, 멀티-모드 오디오 신호 인코더.

청구항 22

청구항 21에 있어서,

상기 멀티-모드 오디오 신호 인코더는 상기 선형-예측 모드 시작 프레임과 연관된 상기 선형-예측 도메인 파라미터들을 사용하여, 상기 선형-예측 모드 시작 프레임 후속하는 상기 결합된 변환-코딩된-여기 선형 예측 모드/대수-코드-여기된 선형 예측 모드 프레임의 적어도 한 부분을 인코딩하는 대수-코드 여기된 선형 예측 모드 인코더를 초기화하도록 구성되는, 멀티-모드 오디오 신호 인코더.

청구항 23

청구항 16에 있어서,

선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분이나, 그 프리-프로세싱된 버전을 분석하여, 상기 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 상기 부분과 연관되는 선형-예측-코딩 필터 계수들을 결정하도록 구성되는 선형-예측-코딩 필터 계수 결정기(340a; 1070c);

상기 선형-예측 코딩 필터 계수들을 스펙트럼 표현($X_0[k]$)으로 변환하여, 다른 주파수들과 연관된 선형-예측-모드 이득 값들($g[k]$, 350c)을 획득하도록 구성되는 필터 계수 변환기(350b; 1070d);

상기 주파수 도메인 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분이나, 그 프리-프로세싱된 버전을, 분석하여, 상기 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 상기 부분과 연관되는 스케일 팩터들을 결정하도록

구성되는 스케일 팩터 결정기(330c; 1070a);

결합기 배열(330e, 350d; 1030e)로서, 상기 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전을, 상기 선형-예측 모드 이득 값들($g[k]$)과 결합하여, 이득-프로세싱된 스펙트럼 컴포넌트들을 획득하되, 상기 오디오 콘텐츠의 상기 주파수-도메인 표현의 스펙트럼 컴포넌트들의 기여분에는 상기 선형-예측-모드 이득 값들에 따라 가중치가 적용되고, 또한

상기 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전을, 상기 스케일 팩터와 결합하여, 이득-프로세싱된 스펙트럼 컴포넌트들을 획득하되, 상기 오디오 콘텐츠의 상기 주파수-도메인 표현의 스펙트럼 컴포넌트들의 기여분에는 상기 스케일 팩터에 따라 가중치가 적용되도록, 구성되는 상기 결합기 배열(330e, 350d; 1030e)을 포함하고,

상기 이득-프로세싱된 스펙트럼 컴포넌트들은 스펙트럼 계수들의 스펙트럼 성형된 세트들을 형성하는, 멀티-모드 오디오 신호 인코더.

청구항 24

오디오 콘텐츠의 인코딩된 표현에 기초하여 상기 오디오 콘텐츠의 디코딩된 표현을 제공하는 방법에 있어서,

상기 오디오 콘텐츠의 복수의 부분들에 대한 디코딩된 스펙트럼 계수들의 세트들을 획득하는 단계;

선형-예측 모드로 인코딩된 상기 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라 디코딩된 스펙트럼 계수 세트에, 또는 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하고, 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들 세트에 따라 디코딩된 스펙트럼 계수 세트에, 또는 그 프리-프로세싱된 버전에 스펙트럼 성형을 적용하는 단계; 및

상기 선형-예측 모드로 인코딩된 상기 오디오 콘텐츠의 한 부분에 대해 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하고, 상기 주파수-도메인 모드로 인코딩된 상기 오디오 콘텐츠의 한 부분에 대해 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하는 단계를 포함하는 오디오 콘텐츠의 디코딩된 표현을 제공하는 방법.

청구항 25

오디오 콘텐츠의 입력 표현에 기초하여 오디오 콘텐츠의 인코딩된 표현을 제공하는 방법에 있어서,

상기 오디오 콘텐츠의 주파수-도메인 표현을 획득하도록, 상기 오디오 콘텐츠의 상기 입력 표현을 프로세싱하는 단계로서, 상기 주파수-도메인 표현(122)이 스펙트럼 계수들의 세트들의 시퀀스를 포함하는 상기 단계;

선형-예측 모드로 인코딩되는 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라 스펙트럼 계수 세트나 혹은 그 프리-프로세싱된 버전에 스펙트럼 성형을 적용하여 스펙트럼 계수들의 스펙트럼-성형된 세트(132)를 획득하는 단계;

주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들 세트에 따라 스펙트럼 계수 세트나 혹은 그 프리-프로세싱된 버전에 스펙트럼 성형을 적용하여 스펙트럼 계수들의 스펙트럼-성형된 세트(132)를 획득하는 단계;

양자화 인코딩을 이용하여 상기 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 부분에 대한 스펙트럼 계수들의 스펙트럼-성형된 세트의 인코딩된 버전을 제공하는 단계; 및

양자화 인코딩을 이용하여 상기 주파수-도메인 모드로 인코딩되는 상기 오디오 콘텐츠의 부분에 대한 스펙트럼 계수들의 스펙트럼-성형된 세트의 인코딩된 버전을 제공하는 단계를 포함하는 오디오 콘텐츠의 디코딩된 표현을 제공하는 방법.

청구항 26

컴퓨터에서 실행될 때 청구항 24 또는 청구항 25에 따른 방법을 수행하는 컴퓨터 프로그램을 기록한 컴퓨터로 판독 가능한 기록 매체.

명세서

기술 분야

- [0001] 본 발명에 따른 실시예들은 오디오 콘텐츠의 인코딩된 표현에 기초하여 오디오 콘텐츠의 디코딩된 표현을 제공하는 멀티-모드 오디오 신호 디코더에 관련된다.
- [0002] 본 발명에 따른 다른 실시예들은 오디오 콘텐츠의 입력 표현에 기초하여 오디오 콘텐츠의 인코딩된 표현을 제공하는 멀티-모드 오디오 신호 인코더에 관련된다.
- [0003] 본 발명에 따른 또 다른 실시예들은 오디오 콘텐츠의 인코딩된 표현에 기초하여 오디오 콘텐츠의 디코딩된 표현을 제공하는 방법에 관련된다.
- [0004] 본 발명에 따른 또 다른 실시예들은 오디오 콘텐츠의 입력 표현에 기초하여 오디오 콘텐츠의 인코딩된 표현을 제공하는 방법에 관련된다.
- [0005] 본 발명에 따른 또 다른 실시예들은 상기의 방법을 구현하는 컴퓨터 프로그램에 관련된다.

배경 기술

- [0006] 이하, 본 발명의 배경기술이 본 발명과 본 발명에 대한 이점에 대한 이해를 도울수 있도록 설명된다.
- [0007] 과거 수십년 동안, 오디오 콘텐츠를 디지털로 저장하고 배포하기 위한 가능성을 창출하기 위한 많은 노력이 있어 왔다. 이와 관련되어 성취된 가장 중요한 것은 국제 표준 ISO/IEC 14496-3이 정의된 것이다. 이러한 표준 3장은 오디오 콘텐츠의 인코딩과 디코딩에 관한 것이며, 3장 4절은 통상적인 오디오 코딩에 관한 것이다. ISO/IEC 14496-3의 3장 4절은 통상적인 오디오 콘텐츠의 인코딩 및 디코딩에 대한 개념을 정의한다. 또한, 품질의 개선 및/또는 필요 비트율을 낮추기 위한 추가적인 개선점이 제안되어 왔다.
- [0008] 무엇보다도, 주파수-도메인 기반의 오디오 코더들의 성능이 스피치를 포함하는 오디오 콘텐츠에 최적이지 아니라는 것이 발견되었다. 최근, 통합 스피치-및-오디오 코덱이 제안되어왔는데, 이는 두 세계, 즉 스피치 코딩과 오디오 코딩으로부터의 기술들을 효율적으로 결합한다(예를 들면, 인용문헌[1] 참조).
- [0009] 이러한 오디오 코더에서, 어떤 오디오 프레임들은 주파수 도메인으로 인코딩되고 어떤 오디오 프레임들은 선형-예측-도메인으로 인코딩된다.
- [0010] 그러나 서로 다른 도메인으로 인코딩된 프레임들 간의 전이는 상당한 정도의 비트율 손실 없이는 어렵다는 것이 발견되었다.

발명의 내용

해결하려는 과제

- [0011] 이러한 상황에 대한 관점에서, 스피치와 통상적인 오디오를 포함하는 오디오 콘텐츠를 인코딩하고 디코딩하는 개념을 창출하고자 하는 열망이 있는데, 이러한 개념은 서로 다른 모드를 사용하여 인코딩된 부분들간의 효율적인 전이의 실현을 허용한다.

과제의 해결 수단

- [0012] 본 발명에 따른 일 실시예는 오디오 콘텐츠의 인코딩된 표현에 기초하여 오디오 콘텐츠의 디코딩된 표현을 제공하는 멀티-모드 오디오 신호 디코더를 창출한다. 오디오 신호 디코더는 오디오 콘텐츠의 복수의 부분들에 대하여 디코딩된 스펙트럼 계수 세트들을 획득하도록 구성되는 스펙트럼 값 결정기를 포함한다. 멀티-모드 오디오 신호 디코더는 또한 스펙트럼 프로세서를 포함하며, 스펙트럼 프로세서는 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라, 디코딩된 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하고, 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들 세트에 따라, 디코딩된 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하도록 구성된다. 멀티-모드 오디오 신호 디코더는 또한 주파수-도메인-대-시간-도메인 컨버터를 포함하며, 주파수-도메인-대-시간-도메인 컨버터는 상기 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 오디오 콘텐츠의 시간-도메인 표현을 획득하고, 상기 주파-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대하여 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 상기 오디오 콘텐츠의 시간-도메인 표현을 획득하도록 구성

된다.

[0013] 이러한 멀티-모드 오디오 신호 디코더는 서로 다른 모드로 인코딩된 오디오 콘텐츠의 부분들 간의 효율적인 전이는 주파수 도메인 내의 스펙트럼 성형, 즉 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 부분들과 선형-예측 모드로 인코딩된 오디오 콘텐츠의 부분들에 대해, 디코딩된 스펙트럼 계수 세트들의 스펙트럼 성형을 수행함으로써 획득될 수 있다는 발견에 기초한다. 이와 같이 함으로써, 선형-예측 모드로 인코딩된 오디오 콘텐츠의 부분들에 대해 디코딩된 스펙트럼 계수의 스펙트럼 성형 세트에 기초하여 획득된 시간-도메인 표현이 "동일한 도메인 내에"(예를 들면, 동일한 변환 타입의 주파수-도메인-대-시간-도메인 변환의 출력값들이 존재하는) 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 부분들에 대해 디코딩된 스펙트럼 계수의 스펙트럼 성형 세트에 기초하여 획득된 시간 도메인 표현으로서 존재한다. 따라서, 선형 예측 모드로 인코딩된 오디오 콘텐츠의 한 부분과 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분의 시간-도메인 표현들이 효율적으로 허용할 수 없는 아티팩트(artifacts)들 없이 결합될 수 있다. 전형적인 주파수-도메인-대-시간-도메인 컨버터의 에일리어싱(aliasing) 제거 특징이, 동일한 도메인 내의, 주파수-도메인-대-시간-도메인 변환 신호들에 의해 이용될 수 있다(예를 들면, 둘 다 오디오 콘텐츠를 오디오 콘텐츠 도메인 내에 표현한다). 따라서, 양질의 전이가 서로 다른 모드로 인코딩된 오디오 콘텐츠의 부분들 간에 획득될 수 있는데, 이때, 이러한 전이를 허용하기 위하여 많은 비트율을 필요로 하지 않는다.

[0014] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 디코더는 오버래퍼를 포함하며, 오버래퍼는 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분의 시간-도메인 표현을 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분과 함께 오버랩-및-가산하도록 구성된다. 서로 다른 모드로 인코딩된 오디오 콘텐츠의 부분을 오버래핑함으로써, 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트들을 멀티-모드 오디오 신호 디코더의 두 모드 내의 주파수-도메인-대-시간-도메인 컨버터로 입력함으로써 획득되는 이점이 실현될 수 있다. 멀티-모드 오디오 신호 디코더의 두 모드 내의 주파수-도메인-대-시간-도메인 변환 전에 스펙트럼 성형을 수행함으로써, 서로 다른 모드로 인코딩된 오디오 콘텐츠의 부분들의 시간-도메인 표현은 매우 양호한 오버랩-및-가산-특징을 포함하는데, 이는 추가적인 부수 정보를 요구하지 않는 양질의 전이를 허용한다.

[0015] 바람직한 일 실시예에서, 주파수-도메인-대-시간-도메인 컨버터는 래핑된 변환을 이용하여 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 오디오 콘텐츠의 시간-도메인 표현을 획득하고 그리고 래핑된 변환을 사용하여 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 오디오 콘텐츠의 시간-도메인 표현을 획득하도록 구성된다. 이 경우, 오버래퍼는 각기 다른 모드로 인코딩된 오디오 콘텐츠의 연속하는 부분들의 시간-도메인 표현들을 오버래핑하도록 구성된다. 따라서, 순조로운 전이가 획득된다. 스펙트럼 성형이 주파수 도메인내에서 상술한 두 가지 모드에 대해 적용된다는 사실 때문에, 주파수-도메인-대-시간-도메인 컨버터에서 두 가지 모드로 제공하는 시간 도메인 표현이 호환되고 양질의 전이를 허용한다. 래핑된 변환의 사용은 품질과 비트율 효율성간의 개선된 균형(tradeoff)을 가져오는데, 이는 래핑된 변환이 양자화 에러가 존재할 때조차 심각한 비트율 오버헤드를 피하면서 순조로운 전이를 허용하기 때문이다.

[0016] 바람직한 일 실시예에서, 주파수-도메인-대-시간-도메인 컨버터는 동일한 변환 타입의 래핑된 변환들을 적용하여 각자 다른 모드로 인코딩된 오디오 콘텐츠의 부분들에 대한 오디오 콘텐츠의 시간-도메인 표현들을 획득하도록 구성된다. 이 경우, 오버래퍼는 각자 다른 모드로 인코딩된 오디오 콘텐츠의 연속하는 부분들의 시간-도메인 표현을 오버랩-및-가산하여 래핑된 변환에 의해 야기된 시간-도메인 에일리어싱(aliasing)이 감소되거나 제거되도록 구성된다. 이러한 개념은 주파수-도메인-대-시간-도메인 변환의 출력 신호가 상술한 두 모드들에 대해서 주파수-도메인 내의 스케일 팩터 파라미터들과 선형-예측-도메인 파라미터들을 적용함으로써 동일한 도메인(오디오 콘텐츠 도메인) 내에 존재한다는 사실에 기초한다. 따라서, 동일한 변환 타입의 래핑된 변환들을 오디오 신호 표현의 연속하고 부분적으로 래핑하는 부분들에 적용함으로써 획득되는, 에일리어싱(aliasing) 제거가 활용될 수 있다.

[0017] 바람직한 일 실시예에서, 오버래퍼는, 연관된 래핑된 변환에 의해 제공되는 바와 같은, 제1 모드로 인코딩된 오디오 콘텐츠의 제1 부분의 윈도우잉된 시간-도메인 표현 또는 그것의 진폭-스케일되었지만 스펙트럼 왜곡되지 않은 버전을 오버랩-및-가산하고, 연관된 래핑된 변환에 의해 제공되는 바와 같은, 제2 모드로 인코딩된 오디오 콘텐츠의 제2 연속하는 부분의 윈도우잉된 시간-도메인 표현 또는 그것의 진폭-스케일되었지만 스펙트럼 왜곡되지 않은 버전을 오버랩-및-가산하도록 구성된다. 합성 래핑된 변환의 출력 신호에서 회피하여, 오디오 콘텐츠의 연속하는 (부분적으로 오버래핑하는) 부분들에 사용되는 모든 서로 다른 코딩 모드들에 공통되지 않는 어느 신호 프로세싱(예를 들면, 필터링 또는 그 유사한)을 적용함으로써, 래핑된 변환의 에일리어싱-제거로부터

취할 수 있는 모든 장점이 획득된다.

- [0018] 바람직한 일 실시예에서, 주파수-도메인-대-시간-도메인 컨버터는 각기 다른 모드로 인코딩된 상기 오디오 콘텐츠의 부분들의 시간-도메인 표현들을 제공하여 상기 제공된 시간-도메인 표현들이 동일한 도메인에 있도록 하되, 이들이, 상기 제공된 시간-도메인 표현 중의 하나 또는 둘 다에, 윈도우잉 전이 연산을 제외한 신호 성형 필터링 연산을 적용하지 않고, 상기 동일한 도메인 내에서 선형으로 결합 가능하도록 구성된다. 즉 주파수-도메인-대-시간-도메인 변환의 출력 신호는 상술한 두 모드들에 대한 오디오 콘텐츠의 시간-도메인 표현들이다(그리고, 여기(excited)-도메인-대-시간-도메인 변환 필터링 연산에 대한 여기 신호들이 아니다).
- [0019] 바람직한 일 실시예에서, 주파수-도메인-대-시간-도메인 컨버터는 역 변경된 이산 코사인 변환을 수행하여, 역 변경된 이산 코사인 변환의 결과로서, 오디오 신호 도메인 내의 오디오 콘텐츠의 시간-도메인 표현을 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분 및 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분 양쪽에 대하여 획득하도록 구성된다.
- [0020] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 디코더는 LPC-필터 계수 결정기를 포함하며, LPC-필터 계수 결정기는 디코딩된 LPC-필터 계수를 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 LPC-필터 계수들의 인코딩된 표현에 기초하여 획득하도록 구성된다. 이 경우, 멀티-모드 오디오 신호 디코더는 또한 필터 계수 변환기를 포함하고, 필터 계수 변환기는 디코딩된 LPC-필터 계수들을 스펙트럼 표현으로 변환하여, 다른 주파수들과 연관된 선형-예측-모드 이득 값들을 획득하도록 구성된다. 따라서, LPC-필터 계수는 선형 예측 도메인 파라미터들로서 제공된다. 멀티-모드 오디오 신호 디코더는 또한 스케일 팩터 결정기를 포함하는데, 스케일 팩터 결정기는 디코딩된 스케일 팩터 값들(스케일 팩터 파라미터들로서 제공되는)을 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 값들의 인코딩된 표현에 기초하여 획득하도록 구성된다. 스펙트럼 프로세서는 스펙트럼 변경기를 포함하는데, 스펙트럼 변경기는 선형-예측 모드로 인코딩된 오디오 콘텐츠의 한 부분에 연관된 디코딩된 스펙트럼 계수들 세트나, 그 프리-프로세싱된 버전을, 선형-예측-모드 이득 값과 결합하여, (디코딩된) 스펙트럼 계수들의 이득-프로세싱된(그리고, 그 결과에 따른, 스펙트럼-성형된) 버전을 획득 하되, 디코딩된 스펙트럼 계수들이나, 프리-프로세싱된 버전의 기여분(contribution)에는 선형-예측-모드 이득 값들에 따라 가중치가 적용되도록 구성된다. 또한, 스펙트럼 변경기는 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 한 부분에 연관된 디코딩된 스펙트럼 계수들 세트나, 그 프리-프로세싱된 버전을, 스케일 팩터 값들과 결합하여, (디코딩된) 스펙트럼 계수들의 스케일-팩터-프로세싱된(스펙트럼 성형된) 버전을 획득하되, 디코딩된 스펙트럼 계수들이나, 그 프리-프로세싱된 버전의 기여분에는 스케일 팩터 값들에 따라 가중치가 적용되도록 구성된다.
- [0021] 이러한 접근방법을 사용함으로써, 고유한 노이즈-성형이 멀티-모드 오디오 신호 디코더의 두 모드들로 획득될 수 있는 한편, 여전히 주파수-도메인-대-시간-도메인 컨버터는 서로 다른 모드로 인코딩된 오디오 신호의 부분들 간의 전이로 양호한 특징을 갖는 출력 신호를 제공하도록 보장한다.
- [0022] 바람직한 일 실시예에서, 필터 계수 변환기는, 선형-예측-코딩 필터(LPC-필터)의 시간-도메인 임펄스 응답을 표현하는, 디코딩된 LPC-필터 계수들을, 오드(odd) 이산 푸리에 변환을 사용하여 스펙트럼 표현으로 변환하도록 구성된다. 또한, 필터 계수 변환기는 선형-예측-모드 이득 값들을 디코딩된 LPC-필터 계수들의 스펙트럼 표현으로부터 도출하여, 이득 값들이 스펙트럼 표현의 계수들의 크기 함수가 되도록 구성된다. 그리하여, 선형-예측 모드로 수행되는, 스펙트럼 성형은 선형-예측-코딩 필터의 노이즈-성형 기능을 취득한다. 따라서, 디코딩된 스펙트럼 표현의(또는 그 프리-프로세싱된 버전의) 양자화 노이즈가 변경되어, "중요한" 주파수에 비하여 양자화 노이즈가 상대적으로 작게 되며, 이때, "중요한" 주파수에 비하여 디코딩된 LPC-필터 계수의 스펙트럼 표현은 상대적으로 크다.
- [0023] 바람직한 일 실시예에서, 필터 계수 변환기와 결합기는 주어진 디코딩된 스펙트럼 계수나, 그 프리-프로세싱된 버전의, 주어진 디코딩된 스펙트럼 계수의 이득-프로세싱된 버전으로의 기여분이 주어진 디코딩된 스펙트럼 계수와 연관된 선형-예측-모드 이득 값의 크기에 의해 결정되도록 구성된다.
- [0024] 바람직한 일 실시예에서, 스펙트럼 값 결정기는 역 양자화를 디코딩된 양자화 스펙트럼 계수들에 적용하여, 디코딩되고 역 양자화된 스펙트럼 계수들을 획득하도록 구성된다. 이 경우, 스펙트럼 변경기는 주어진 디코딩된 스펙트럼 계수에 대한 효과적인 양자화 단계를 주어진 디코딩된 스펙트럼 계수와 연관된 선형-예측-모드 이득 값의 크기에 따라 조정함으로써 양자화 노이즈 성형을 수행하도록 구성된다. 따라서, 스펙트럼 도메인 내에서 수행되는 노이즈-성형은 LPC-필터 계수들로 기술되는 신호 특징들에 적용된다.

- [0025] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 디코더는 중간 선형-예측-모드 시작 프레임을 사용하여 주파수-도메인 모드 프레임으로부터 결합된 선형-예측 모드/대수-코드-여기된 선형-예측 모드 프레임으로 전이하도록 구성된다. 이 경우, 오디오 신호 디코더는 선형-예측 모드 시작 프레임에 대한 디코딩된 스펙트럼 계수들 세트를 획득하도록 구성된다. 또한, 오디오 디코더는 스펙트럼 성형을 선형-예측 모드 시작 프레임에 대한 상기 디코딩된 스펙트럼 계수들 세트나, 그 프리-프로세싱된 버전에, 그와 연관된 선형-예측-도메인 파라미터들 세트에 따라 적용하도록 구성된다. 또한, 오디오 신호 디코더는 선형-예측 모드 시작 프레임의 시간-도메인 표현을 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에 기초하여 획득하도록 구성된다. 오디오 디코더는 또한 상대적으로 긴 좌측 전이 슬로프를 갖고 상대적으로 짧은 우측 전이 슬로프를 갖는 시작 윈도우를 상기 선형-예측 모드 시작 프레임의 시간-도메인 표현에 적용하도록 구성된다. 이와 같이 함으로써, 주파수-도메인 모드 프레임 및 결합된 선형-예측 모드/대수-코드-여기된 선형-예측 모드 프레임이 창출되는데, 이는 선형 주파수-도메인 모드 프레임을 갖는 양질의 오버랩-및-가산 특징들을 포함하며, 동시에, 연속하는 결합된 선형-예측 모드/대수-코드-여기된 선형-예측 모드 프레임에서 사용가능한 선형-예측-도메인 계수들을 만든다.
- [0026] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 디코더는 선형 예측-모드 시작 프레임을 선행하는 주파수-도메인 모드 프레임의 시간-도메인 표현의 우측 부분을 선형 예측-모드 시작 프레임의 시간-도메인 표현의 좌측 부분과 오버래핑하여, 시간-도메인 에일리어싱의 감소 또는 제거를 획득하도록 구성된다. 본 실시예는 양호한 시간-도메인 에일리어싱 제거 특징은 주파수 도메인 내의 선형 예측-모드 시작 프레임의 스펙트럼 성형을 수행함으로써 획득될 수 있다는 발견에 기초하는데, 이는 또한 선행하는 주파수-도메인 모드 프레임의 스펙트럼 성형이 주파수-도메인 내에서 수행되기 때문이다.
- [0027] 바람직한 일 실시예에서, 오디오 신호 디코더는 선형 예측-모드 시작 프레임과 연관된 선형 예측 도메인 파라미터들을 사용하여, 결합된 선형-예측 모드/대수-코드-여기된 선형 예측 모드 프레임의 적어도 한 부분을 인코딩하는 대수-코드-여기된 선형 예측 모드 디코더를 초기화하도록 구성된다. 이러한 방식으로, 몇몇 종래의 접근방법에 따른 선형-예측-도메인 파라미터들의 추가적인 세트를 전송해야하는 필요성이 제거된다. 차라리, 선형 예측-모드 시작 프레임은, 상대적으로 긴 오버랩 기간에 대해서 조차, 선행하는 주파수-도메인 모드 프레임으로부터 양질의 전이를 생성하고, 대수-코드-여기된 선형 예측(ACELP) 모드 디코더를 초기화하도록 허용한다. 따라서, 양호한 오디오 품질을 갖는 전이가 상당히 효율적으로 획득된다.
- [0028] 본 발명에 따른 다른 실시예는 오디오 콘텐츠의 입력 표현에 기초하여 오디오 콘텐츠의 인코딩된 표현을 제공하는 멀티-모드 오디오 신호 인코더를 창출한다. 오디오 인코더는 오디오 콘텐츠의 입력 표현을 프로세싱하여, 오디오 콘텐츠의 주파수-도메인 표현을 획득하도록 구성되는 시간-도메인-대-주파수-도메인 컨버터를 포함한다. 오디오 신호 인코더는 스펙트럼 프로세서를 더 포함하는데, 스펙트럼 프로세서는 상기 선형-예측 모드로 인코딩되는 상기 오디오 콘텐츠의 한 부분에 대한 선형-예측-도메인 파라미터 세트에 따라 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하고, 상기 주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분에 대한 스케일 팩터 파라미터들 세트에 따라 스펙트럼 계수 세트 혹은, 그 프리-프로세싱된 버전에 스펙트럼 성형(shaping)을 적용하도록 구성된다.
- [0029] 상술한 멀티-모드 오디오 신호 인코더는, 낮은 왜곡을 갖는 단순한 오디오 디코딩을 허용하는, 효율적인 오디오 인코딩이 오디오 콘텐츠의 입력 표현은, 선형-예측 모드로 인코딩된 오디오 콘텐츠의 부분들과 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 부분들 모두에 대한 주파수-도메인(또한 시간-주파수 도메인으로서 고안된)으로 변환되면, 획득될 수 있다는 발견에 기초한다. 또한, 양자화 에러는 스펙트럼 성형을 선형-예측 모드로 인코딩된 오디오 콘텐츠의 부분들과 주파수-도메인 모드로 인코딩된 오디오 콘텐츠의 부분들 모두에 대한 스펙트럼 계수들(또는 그 프리-프로세싱된 버전) 세트에 적용함으로써 감소될 수 있다는 것이 발견되었다. 만일 서로 다른 타입의 파라미터들이 서로 다른 모드들로 스펙트럼 성형을 결정하기 위하여 사용된다면(즉, 선형-예측 모드의 선형-예측-도메인 파라미터들과 주파수-도메인 모드의 스케일 팩터 파라미터들), 노이즈 성형이 오디오 콘텐츠의 현재-프로세싱된 부분의 특징에 적응될 수 있는 한편, 여전히 시간-도메인-대-주파수-도메인 변환을 서로 다른 모드에 있는 동일한 오디오 신호(의 부분들)에 적용할 수 있다. 이에 따라, 멀티-모드 오디오 신호 인코더는 일반적인 오디오 부분들과 스피치 오디오 부분들을 모두 갖는 오디오 신호들에 대하여 적절한 타입의 스펙트럼 성형을 스펙트럼 계수들 세트들에 선택적으로 적용함으로써 양호한 코딩 성능을 제공할 수 있다. 즉, 선형-예측-도메인 파라미터들의 세트에 기초한 스펙트럼 성형이 스피치-같이 인지되는 오디오 프레임에 대한 스펙트럼 계수들의 세트에 적용될 수 있고, 그리고 스케일 팩터 파라미터들 세트에 기초한 스펙트럼 성형이 스피치-같이 인지되기보다는 일반적인 오디오 타입으로 인지되는 오디오 프레임에 대한 스펙트럼 계수들의 세트에 적용될 수 있다.

- [0030] 요약하면, 멀티-모드 오디오 신호 인코더는 시간적으로 변하는 특징들(몇몇 시간적 부분들에 대해서는 스피치 같고 다른 부분들에 대해서는 일반적인 오디오)을 갖는 오디오 콘텐츠의 인코딩을 허용하며, 오디오 콘텐츠의 시간-도메인 표현은 서로 다른 모드로 인코딩된 오디오 콘텐츠의 부분들에 대하여 동일한 방식으로 주파수 도메인으로 변환된다. 오디오 콘텐츠의 서로 다른 부분들의 서로 다른 특징들은 서로 다른 파라미터들(선형-예측-도메인 파라미터들 대 스케일 팩터 파라미터들)에 기초한 스펙트럼 성형을 적용함으로써 고려되어, 스펙트럼 성형된 스펙트럼 계수들이나 후속하는 양자화를 획득한다.
- [0031] 바람직한 일 실시예에서, 시간-도메인-대-주파수-도메인 컨버터는 오디오 신호 도메인 내의 오디오 콘텐츠의 시간-도메인 표현을 선형-예측 모드로 인코딩되는 오디오 콘텐츠의 한 부분과 주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분 양쪽에 대한 오디오 콘텐츠의 주파수-도메인 표현으로 전환하도록 구성된다. 주파수-도메인 모드와 선형-예측 모드 모두에 대하여 동일한 입력 신호에 기초한 시간-도메인-대-주파수-도메인 변환(변환 연산의 의미로, 예를 들면, MDCT 변환 연산이나 필터 뱅크-기반의 주파수 분리 연산)을 수행함으로써, 디코더-측면의 오버랩-및-가산 연산은 특히 매우 효율적으로 수행될 수 있으며, 이는 디코더 측면의 신호 재건을 용이하게 하고 서로 다른 모드들 간의 전이가 있을 때마다 추가적인 데이터를 전송할 필요성을 제거한다.
- [0032] 바람직한 일 실시예에서, 시간-도메인-대-주파수-도메인 컨버터는 동일한 변환 타입의 분석 래핑된 변환을 적용하여 각자 다른 모드로 인코딩되는 오디오 콘텐츠의 부분들에 대한 주파수-도메인 표현들을 획득하도록 구성된다. 다시, 동일한 변환 타입의 래핑된 변환들을 사용하면, 오디오 콘텐츠의 단순한 재건을 허용하는 한편, 폐쇄(blocking) 아트팩트를 피할 수 있다. 특히, 상당한 오버헤드 없이도 임계(critical) 샘플링을 사용할 수 있다.
- [0033] 바람직한 일 실시예에서, 스펙트럼 프로세서는 상기 선형-예측 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 상관관계-기반 분석을 사용하여 획득된 선형-예측 도메인 파라미터들의 세트에 따라, 또는 상기 주파수-도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 음향심리 모델 분석을 사용하여 획득된 스케일 팩터 파라미터들의 세트에 따라, 스펙트럼 계수들의 상기 세트나, 또는 그 프리-프로세싱된 버전에 상기 스펙트럼 성형을 선택적으로 적용하도록 구성된다. 이와 같이 함으로써, 적절한 노이즈 성형이, 상관-기반의 분석이 의미 있는 노이즈 성형 정보를 제공하는, 오디오 콘텐츠의 스피치-같은 부분들과, 심리음향 모델 분석이 의미 있는 노이즈 성형 정보를 제공하는, 오디오 콘텐츠의 일반적 오디오 부분들 모두에 대해서 성취될 수 있다.
- [0034] 바람직한 일 실시예에서, 오디오 신호 인코더는 오디오 신호를 분석하여 오디오 콘텐츠의 한 부분이 선형-예측 모드 또는 주파수-도메인 모드로 인코딩되는지 여부를 결정하도록 구성되는 모드 선택기를 포함한다. 따라서, 적절한 노이즈 성형 개념이 선택되어 상기 타입의 시간-도메인-대-주파수-도메인 변환이 어떤 경우에는 영향을 받지 않도록 한다.
- [0035] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 인코더는, 주파수-도메인 모드 프레임 및 결합된 변환-코딩된-여기 선형-예측 모드/대수-코드-여기된 선형 예측 모드 프레임 사이에 선형-예측 모드 시작 프레임으로서 존재하는, 오디오 프레임을 인코딩하도록 구성된다. 멀티-모드 오디오 신호 인코더는 상대적으로 긴 좌측 전이 슬로프와 상대적으로 짧은 우측 전이 슬로프를 갖는 시작 윈도우를 선형-예측 모드 시작 프레임의 시간-도메인 표현에 적용하여, 윈도우잉된 시간-도메인 표현을 획득하도록 구성된다. 멀티-모드 오디오 신호 인코더는 선형 예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 주파수-도메인 표현을 획득하도록 구성된다. 멀티-모드 오디오 신호 인코더는 선형-예측 모드 시작 프레임에 대한 선형-예측 도메인 파라미터들의 세트를 획득하고, 그리고 선형-예측 도메인 파라미터들 세트에 따라 상기 선형-예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전에 스펙트럼 성형을 적용하도록 구성된다. 멀티-모드 오디오 신호 인코더는 또한 선형-예측 도메인 파라미터의 세트와 상기 선형-예측 모드 시작 프레임의 상기 윈도우잉된 시간-도메인 표현의 상기 스펙트럼 성형된 주파수 도메인 표현을 인코딩하도록 구성된다. 이러한 방식으로, 전이 오디오 프레임의 인코딩된 정보가 획득되는데, 이때 전이 오디오 프레임의 인코딩된 정보는 오디오 콘텐츠의 재건에 사용될 수 있으며, 전이 오디오 프레임에 관한 인코딩된 정보는 순조로운 좌-측 전이를 허용함과 동시에 후속하는 오디오 프레임을 디코딩하는 ACELP 모드 디코더의 초기화를 허용한다. 멀티-모드 오디오 신호 인코더의 서로 다른 모드들간의 전이에 의해 야기되는 오버헤드가 최소화 된다.
- [0036] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 인코더는 선형-예측 모드 시작 프레임과 연관된 선형-예측 도메인 파라미터들을 사용하여, 선형-예측 모드 시작 프레임을 후속하는 결합된 변환-코딩된-여기 선형 예측 모드/대수-코드-여기된 선형 예측 모드 프레임의 적어도 한 부분을 인코딩하는 대수-코드 여기된 선형 예측 모드 인코더를 초기화하도록 구성된다. 따라서, 선형-예측 모드 시작 프레임에 대해 획득되고 또한 오디오 콘텐츠를 표현하는 비트 스트림으로 인코딩되는, 선형-예측-도메인 파라미터들이 후속하는 오디오 프레임의 인코딩을 위해

재-사용되는데, 여기서 ACELP-모드가 사용된다. 이것은 인코딩 효율을 증가시키고 추가적인 ACELP 초기화 사이드 정보 없는 효율적인 디코딩을 허용한다.

[0037] 바람직한 일 실시예에서, 멀티-모드 오디오 신호 인코더는 선형-예측 모드로 인코딩되는 오디오 콘텐츠의 한 부분이나, 그 프리-프로세싱된 버전을 분석하여, 선형-예측 모드로 인코딩되는 오디오 콘텐츠의 부분과 연관되는 LPC-필터 계수들을 결정하도록 구성되는 LPC-필터 계수 결정기를 포함한다. 멀티-모드 오디오 신호 인코더는 또한 선형-예측 코딩 필터 계수들을 스펙트럼 표현으로 변환하여, 다른 주파수들과 연관된 선형-예측-모드 이득값들을 획득하도록 구성되는 필터 계수 변환기를 포함한다. 멀티-모드 오디오 신호 인코더는 또한 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분이나, 그 프리-프로세싱된 버전을, 분석하여, 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 부분과 연관되는 스케일 팩터들을 결정하도록 구성되는 스케일 팩터 결정기를 포함한다. 멀티-모드 오디오 신호 인코더는 또한 결합기 배열을 포함하는데, 결합기 배열은 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전을, 상기 선형-예측 모드 이득값들과 결합하여, 이득-프로세싱된 스펙트럼 컴포넌트들(또한 계수들로 지시되는)을 획득하되, 상기 오디오 콘텐츠의 상기 주파수-도메인 표현의 스펙트럼 컴포넌트들(또는 스펙트럼 계수들)의 기여분에는 상기 선형-예측-모드 이득값들에 따라 가중치가 적용되도록 구성된다. 결합기는 또한 주파수 도메인 모드로 인코딩되는 오디오 콘텐츠의 한 부분의 주파수-도메인 표현이나, 그 프리-프로세싱된 버전을, 스케일 팩터와 결합하여, 이득-프로세싱된 스펙트럼 컴포넌트들(또는 스펙트럼 계수들)을 획득하되, 오디오 콘텐츠의 주파수-도메인 표현의 스펙트럼 컴포넌트들의 기여분에는 스케일 팩터에 따라 가중치가 적용되도록 구성된다.

[0038] 본 실시예에서, 이득-프로세싱된 스펙트럼 컴포넌트들은 스펙트럼 계수들(또는 스펙트럼 컴포넌트들)의 스펙트럼 성형된 세트들을 형성한다.

[0039] 본 발명에 따른 다른 또 다른 실시예는 오디오 콘텐츠의 디코딩된 표현을 오디오 콘텐츠의 인코딩된 표현에 기초하여 제공하는 방법을 창출한다.

[0040] 본 발명에 따른 다른 또 다른 실시예는 오디오 콘텐츠의 인코딩된 표현을 오디오 콘텐츠의 입력 표현에 기초하여 제공하는 방법을 창출한다.

[0041] 본 발명에 따른 다른 또 다른 실시예는 상기의 하나 이상의 방법들을 수행하는 컴퓨터 프로그램을 창출한다.

발명의 효과

[0042] 본 발명에 따른 실시예들은 동일 도메인 내의 주파수-도메인 코더와 LPC 코더 MDCT를 수행하는 한편 LPC를 MDCT 도메인 내의 양자화 에러를 성형하는데 이용하여, LPC가 여전히 ACELP 같은 스피치-코더로 스위칭하도록 사용될 수 있고 또한 시간-도메인 에일리어싱 제거(TDAC)는 TCX 로부터 주파수-도메인 코더로(또한 그 반대로)의 전이 동안 가능하며, 그때, 결정적 샘플링이 유지되는 효과가 있다. 또한, LPC는 여전히 ACELP의 주위에서 노이즈-성형으로서 사용되며, 이는 동일한 대상 함수를 사용하여 TCX와 ACELP 모두에 대하여, 예를 들면, 폐쇄-루프 결정 프로세스 내의 LPC-기반의 가중된 세그먼트 SNR을 최대화할 수 있도록 한다.

도면의 간단한 설명

[0043] 본 발명의 실시예들은 다음에 첨부되는 도면들을 참조하여 설명된다.

도 1은 본 발명의 실시예에 따른 오디오 신호 인코더를 나타내는 블록 구성도이다.

도 2는 참조 오디오 신호 인코더의 블록 구성도이다.

도 3은 본 발명의 실시예에 따른 오디오 신호 인코더의 블록 구성도이다.

도 4는 TCX 윈도우를 위한 LPC 계수들 보간의 예시도이다.

도 5는 디코딩된 LPC 필터 계수들에 기반한 선형-예측-도메인 이득값들을 산출하기 위한 함수의 컴퓨터 프로그램 코드를 나타낸다.

도 6은 선형-예측 모드 이득값들(또는 선형-예측-도메인 이득값들)을 가진 디코딩된 스펙트럼 계수들의 결합한 세트를 위한 컴퓨터 프로그램 코드를 나타낸다.

도 7은 오버헤드로서 이른바 "LPC"를 보내는 스위칭된 시간 도메인/주파수 도메인(TD/FD) 코덱을 위하여 다른 프레임들의 도식적 표현과 연관 정보를 나타낸다.

도 8은 전이를 위한 "LPC2MDCT"를 이용하여 주파수 도메인으로부터 선형-예측-도메인 코더로 전환을 위하여 프레임들의 도식적 표현과 연관 파라미터들을 나타낸다.

도 9는 TCX와 주파수 도메인 코더를 위한 LPC 기반 노이즈 성형을 포함하는 오디오 신호 인코더의 도식적 표현을 나타낸다.

도 10은 신호 도메인에서 수행되는 TCX MDCT와 함께 통합된 스피치-앤드-오디오-코딩(USAC)의 통합 뷰를 나타낸다.

도 11은 본 발명의 실시예에 따른 오디오 신호 디코더의 블록 구성도이다.

도 12는 신호 도메인에서 TCX-MDCT를 가진 USAC 디코더의 통합 뷰를 나타낸다.

도 13은 도 7과 12에 따른 오디오 신호 디코더들에서 수행될 수 있는 프로세싱 단계들을 도식적으로 나타낸다.

도 14는 도 11과 12에 따른 오디오 디코더들에서 후속의 오디오 프레임들의 프로세싱을 도식적으로 나타낸다.

도 15는 다양한 MOD[]의 함수로써 스펙트럼 계수들의 수를 나타내는 표이다.

도 16은 윈도우 시퀀스와 변환 윈도우들을 나타내는 표이다.

도 17a는 본 발명의 실시예에서 오디오 윈도우 전이를 도식적으로 나타낸다.

도 17b는 본 발명의 확장된 실시예에서 오디오 윈도우 전이를 나타내는 표이다.

도 18은 인코딩된 LPC 필터 계수에 따른 선형-예측-도메인 이득값들 $g[k]$ 를 산출하는 프로세싱 절차를 나타낸다.

발명을 실시하기 위한 구체적인 내용

[0044] 1. 도 1에 따른 오디오 신호 인코더

[0045] 다음에서 본 발명의 실시예에 따른 오디오 신호 인코더는 도 1을 참조하여 논의된다. 도 1은 멀티-모드 오디오 신호 인코더(100)의 블록 구성도이다. 멀티-모드 오디오 신호 인코더(100)는 때로는 역시 오디오 인코더로 간단하게 표시된다.

[0046] 오디오 인코더(100)는 오디오 콘텐츠의 입력 표현(110)을 수신하도록 구성된다. 여기서 입력 표현(110)은 일반적으로 시간-도메인 표현이다. 오디오 인코더(100)는, 그에 기반하여, 오디오 콘텐츠의 인코딩된 표현을 제공한다. 예를 들어, 오디오 인코더(100)는 인코딩된 오디오 표현인 비트스트림(112)을 제공한다.

[0047] 오디오 인코더(100)는 오디오 콘텐츠의 입력 표현(110) 또는, 그 프리-프로세싱된(pre-processed) 버전(110')을 수신하도록 구성되는 시간-도메인-대-주파수-도메인 컨버터(120)를 포함한다. 시간-도메인-대-주파수-도메인 컨버터(120)는, 입력 표현(110, 110')에 기반하여, 오디오 콘텐츠의 주파수-도메인 표현(122)을 제공한다. 주파수-도메인 표현(122)은 스펙트럼 계수들의 세트들의 시퀀스의 형태를 가지고 올 수 있다. 예를 들어, 시간-도메인-대-주파수-도메인 컨버터는 윈도우에 기반한 시간-도메인-대-주파수-도메인 컨버터일 수 있으며, 이는 입력 오디오 콘텐츠의 첫 번째 프레임의 시간-도메인 샘플들에 기초한 스펙트럼 계수들의 첫 번째 세트를 제공하고, 그리고 입력 오디오 콘텐츠의 두 번째 프레임의 시간-도메인 샘플들에 기초한 스펙트럼 계수들의 두 번째 세트를 제공한다. 입력 오디오 콘텐츠의 첫 번째 프레임은, 예를 들어, 대략 50%까지, 입력 오디오 콘텐츠의 두 번째 프레임을 가지고 오버랩할 수 있다. 시간-도메인 윈도우잉은 첫 번째 오디오 프레임으로부터 스펙트럼 계수들의 첫 번째 세트를 산출하도록 적용될 수 있으며, 윈도우잉은 또한 두 번째 오디오 프레임으로부터 스펙트럼 계수들의 두 번째 세트를 산출하도록 적용될 수 있다. 이리하여, 시간-도메인-대-주파수 도메인 컨버터는 입력 오디오 정보의 윈도우잉된 부분들(예를 들어, 오버랩핑된 프레임들)의 오버랩핑된 변환들을 수행하도록 구성될 수 있다.

[0048] 또한, 오디오 인코더(100)는, 오디오 콘텐츠(또는, 선택적으로, 그것의 스펙트럼의 포스트-프로세싱된(post-processed) 버전인 122')의 주파수-도메인 표현(122)을 수신하고, 그리고, 그것에 기초하여, 스펙트럼 계수들의 스펙트럼-성형된(spectrally-shaped) 세트들의 시퀀스를 제공하도록 구성된 스펙트럼 프로세서(130)를 포함한다. 스펙트럼 프로세서(130)는, 스펙트럼-성형된 세트(132)를 얻기 위하여, 선형-예측-도메인 파라미터들의 세트에 따라, 스펙트럼 계수들의 스펙트럼 성형을 스펙트럼 계수들의 세트(122) 또는 그것의 프리-프로세싱된 버전(122')에 적용하도록 구성될 수 있다. 또한, 스펙트럼 프로세서(130)는, 주파수 도메인 모드에서 인코딩

되기 위한 오디오 콘텐츠의 상기 부분에 대한 스펙트럼 계수들의 스펙트럼-성형된 세트(132)를 얻기 위하여, 주파수-도메인 모드에서 인코딩되기 위한 오디오 콘텐츠의 부분(예를 들어, 프레임)에 대한 스케일 팩터 파라미터들(136)의 세트에 따라, 스펙트럼 계수들의 세트(122) 또는 그것의 프리-프로세싱된 버전(122')에 스펙트럼 성형을 적용하도록 구성될 수 있다. 예를 들어, 스펙트럼 프로세서(130)는, 선형-예측-도메인 파라미터들(134)의 세트와 스케일 팩터 파라미터들(136)의 세트를 제공하도록 구성된 파라미터 제공기(138)를 포함한다. 예를 들어, 파라미터 제공기(138)는 선형-예측-도메인 분석기를 이용한 선형-예측-도메인 파라미터들(134)의 세트를 제공하고, 그리고, 음향-심리 모델 프로세서를 이용하는 스케일 팩터 파라미터(136)들의 세트를 제공할 수 있다. 그러나, 선형-예측-도메인 파라미터들(134) 또는 스케일 팩터 파라미터들(136)을 제공하는 다른 가능성들이 적용될 수 있다.

[0049] 또한, 오디오 인코더(100)는, 오디오 콘텐츠의 각각의 부분(예를 들어, 각각의 프레임)을 위한 스펙트럼 계수들(스펙트럼 프로세서(130))에 의해 제공된 것으로써의 스펙트럼-성형된 세트(132)를 수신하도록 구성된 양자화 인코더(140)를 포함한다. 그렇지 않으면, 양자화 인코더(140)는 스펙트럼 계수들의 스펙트럼-성형된 세트(132)의 포스트-프로세싱된 버전(132')을 수신할 수 있다. 양자화 인코더(140)는 스펙트럼 계수들(132)(또는, 선택적으로, 그것의 프리-프로세싱된 버전)의 스펙트럼-성형된 세트의 인코딩된 버전(142)을 제공하도록 구성된다. 예를 들어, 양자화 인코더(140)는, 선형-예측 모드에서 인코딩되도록 오디오 콘텐츠의 부분을 위한 스펙트럼 계수들의 스펙트럼 성형된 세트(132)의 인코딩된 버전(142)을 제공하고, 그리고 또한, 주파수-도메인 모드에서 인코딩되도록 오디오 콘텐츠의 부분을 위한 스펙트럼 계수들의 스펙트럼 성형된 세트(132)의 인코딩된 버전(142)을 제공하도록 구성된다. 다시 말해, 동일한 양자화 인코더(140)는, 오디오 콘텐츠의 부분이 선형-예측 모드에서 또는 주파수 예측 모드에서 인코딩되었는지에 상관없이, 스펙트럼 계수들의 스펙트럼-성형된 세트들을 인코딩하기 위하여 이용되어 질 수 있다.

[0050] 게다가, 오디오 인코더(100)는 스펙트럼 계수들의 스펙트럼-성형된 세트들의 인코딩된 버전들(142)에 기반한 비트스트림(112)을 제공하도록 구성된 비트스트림 페이로드 포맷터(bitstream payload formatter)(150)를 선택적으로 포함할 수 있다. 그러나, 비트스트림 페이로드 포맷터(150)는 구성 정보 컨트롤 정보 등 뿐만 아니라, 비트스트림(112)안에 추가적으로 인코딩된 정보를 당연히 포함할 수 있다. 예를 들어, 선택적인 인코더(160)는, 선형-예측-도메인 파라미터들의 인코딩된 세트(134) 그리고/또는 스케일 팩터 파라미터들의 세트(136)를 수신하고, 비트스트림 페이로드 포맷터(150)에 그것의 인코딩된 버전을 제공할 수 있다. 따라서, 선형-예측-도메인 파라미터들의 세트(134)의 인코딩된 버전은 선형-예측 모드에서 인코딩되는 오디오 콘텐츠의 부분을 위한 비트스트림(112)에 포함될 수 있고, 스케일 팩터 파라미터들의 세트(136)의 인코딩된 버전은 주파수-도메인에서 인코딩된 오디오 콘텐츠의 부분을 위한 비트스트림(112)에 포함될 수 있다.

[0051] 오디오 인코더(100)는 선택적으로, 오디오 콘텐츠(예를 들어, 오디오 콘텐츠의 프레임)의 부분이 선형-예측 모드 또는 주파수-도메인 모드에서 인코딩되는지 결정하도록 구성된 모드 컨트롤러(170)를 더 포함한다. 이 목적을 위하여, 모드 컨트롤러(170)는 오디오 콘텐츠의 입력 표현(110), 그것의 프리-프로세싱된 버전(110') 또는 그것의 주파수-도메인 표현(122)을 수신할 수 있다. 모드 컨트롤러(170)는, 예를 들어, 오디오 콘텐츠의 스피치-유사(speech-like) 부분들을 결정하는 스피치 탐색 알고리즘을 사용하고, 스피치-유사 부분에 대한 응답으로 선형-예측 모드에서 오디오 콘텐츠 부분을 인코딩하도록 표시하는 모드 컨트롤 신호(172)를 제공할 수 있다. 반대로, 만약 모드 컨트롤러가, 오디오 콘텐츠의 주어진 부분이 스피치-유사하지 않다면, 모드 컨트롤러(170)는 주파수-도메인 모드에서 오디오 콘텐츠의 상기 부분을 인코딩함을 표시하는 모드 컨트롤 신호(172)와 같은 모드 컨트롤 신호(172)를 제공한다.

[0052] 다음으로, 오디오 인코더(100)의 전체적인 기능성이 자세하게 논의될 것이다. 멀티-모드 오디오 신호 인코더(100)는 스피치-유사한 오디오 콘텐츠 부분들과 스피치-유사하지 않은 오디오 콘텐츠 부분들을 모두 효과적으로 인코딩하도록 구성된다. 이러한 목적을 위하여, 오디오 인코더(100)는 적어도 두 모드를 포함하는데, 즉, 선형-예측 모드와 주파수-도메인 모드이다. 그러나, 오디오 인코더(100)의 시간-도메인-대-주파수-도메인 컨버터(120)는 오디오 콘텐츠(예를 들어, 입력 표현(110), 또는 그것의 프리-프로세싱된 버전(110'))의 동일한 시간-도메인 표현을 선형-예측 모드와 주파수-도메인 모드 모두를 위한 주파수-도메인으로 변환하도록 구성된다. 그러나, 주파수-도메인 표현(122)의 주파수 해상도는 수행의 다른 연산 모드들과 다를 수 있다. 주파수-도메인 표현(122)은 즉시 양자화되거나 인코딩되지 않으며, 오히려 양자화와 인코딩전에 스펙트럼-성형된다. 스펙트럼-성형은 양자화 인코더(140)에 의해 도입되는 양자화 노이즈의 효과를 충분히 작게 유지하도록 하는 방식으로, 과도한 왜곡들을 피하도록 수행된다. 선형-예측 모드에서, 스펙트럼-성형은, 오디오 콘텐츠로부터 도출되는 선형-예측-도메인 파라미터들의 세트(134)에 상응하여 수행된다. 이러한 경우에, 스펙트럼 성형은, 예를 들어, 선형-

예측-도메인 파라미터들의 주파수-도메인 표현의 상응하는 스펙트럼 계수가 비교적 큰 값을 포함하고 있다면, 스펙트럼 계수들이 강조되도록(더 가중된) 수행될 수 있다. 다시 말해, 주파수-도메인 표현(122)의 스펙트럼 계수들은 선형-예측-도메인 파라미터들의 스펙트럼 도메인 표현의 상응하는 스펙트럼 계수들에 일치하도록 가중된다. 따라서, 선형-예측-도메인 파라미터들의 스펙트럼 도메인 표현의 상응하는 스펙트럼 계수가 비교적 더 큰 값을 갖도록, 주파수-도메인 표현(122)의 스펙트럼 계수들은, 스펙트럼 계수들의 스펙트럼-성형된 세트(132)에서의 더 높은 가중으로 인하여 비교적 높은 해상도를 가지도록 양자화된다. 다시 말해, 선형-예측-도메인 파라미터들(134)(예를 들어, 선형-예측-도메인 파라미터들(134)의 스펙트럼-도메인 표현과 일치하는)에 일치하는 스펙트럼 성형은 좋은 노이즈 성형을 가져올 수 있도록 하는 오디오 콘텐츠의 부분들이 있으며, 양자화 노이즈에 대해 더 민감한 주파수-도메인 표현(132)의 스펙트럼 계수들은 스펙트럼 성형에 더 높게 가중되기 때문에, 양자화 인코더(140)에 의해 도입되는 실질적인 양자화 노이즈는 실제로 감소된다.

[0053] 반대로, 주파수-도메인 모드에서 인코딩되는 오디오 콘텐츠 부분들은 다른 스펙트럼 성형을 경험한다. 이러한 경우에, 스케일 팩터 파라미터들(136)은, 예를 들어, 음향-심리 모델 프로세서를 사용하는 것을 결정한다. 음향-심리 모델 프로세서는 주파수-도메인 표현(122)의 스펙트럼 요소들의 스펙트럼 마스킹 그리고/또는 임시의 마스킹을 평가한다. 스펙트럼 마스킹과 시간적 마스킹의 이러한 평가는 주파수-도메인 표현(122)의 스펙트럼 요소들(예를 들어, 스펙트럼 계수들)이 높은 효과의 양자화 정확도를 가지고 인코딩되도록 하고, 주파수-도메인 표현(122)의 스펙트럼 요소들(예를 들어, 스펙트럼 계수들)은 비교적 낮은 효과의 양자화 정확도를 가지고 인코딩하도록 결정되곤 한다. 다시 말해, 음향-심리 모델 프로세서는, 예를 들어, 다양한 스펙트럼 요소들의 음향-심리 관련성을 결정하고, 음향-심리적으로 덜 중요한 스펙트럼 요소들은 낮은 또는 훨씬 낮은 양자화 정확도를 가지고 양자화되도록 표시할 수 있다. 따라서, 스펙트럼 성형(스펙트럼 프로세서(130)에 의해 수행되는)은, 음향-심리 모델 프로세서에 의해 제공되는 스케일 팩터 파라미터들(136)과 일치하도록, 주파수-도메인 표현(122)(또, 그것의 포스트-프로세싱된 버전(122'))의 스펙트럼 요소들(예를 들어, 스펙트럼 계수들)을 가중할 수 있다. 음향-심리적으로 중요한 스펙트럼 요소들은 스펙트럼 성형에서 더 높은 가중이 주어져서, 그들은 양자화 인코더(140)에 의해 높은 양자화 정확도를 가지고 효과적으로 양자화될 수 있다. 이리하여, 스케일 팩터들은 다양한 주파수들과 주파수 밴드들의 음향심리적 관련성을 표현할 수 있다.

[0054] 결론적으로, 오디오 인코더(100)는 적어도 두 개의 다른 모드 사이에 전환될 수 있는데, 이는 선형-예측 모드와 주파수-도메인 모드이다. 오디오 콘텐츠의 오버랩핑 부분들은 모드들의 차이에서 인코딩될 수 있다. 이러한 목적에서, 동일한 오디오 신호의 다른(그러나 바람직하게 오버랩핑) 부분들의 주파수-도메인 표현들은 다른 모드들에서 오디오 콘텐츠의 후속(예를 들어, 바로 다음)의 부분들을 인코딩할 때 사용된다. 주파수-도메인 표현(122)의 스펙트럼 도메인 요소들은 주파수-도메인 모드에서 인코딩되는 오디오 콘텐츠의 부분을 위한 선형-예측-도메인 파라미터들의 세트에 따라, 그리고 주파수-도메인 모드에서 인코딩되는 오디오 콘텐츠의 부분을 위한 스케일 팩터 파라미터들에 따라서 스펙트럼 성형된다. 적절한 스펙트럼 성형을 결정하도록 사용되고, 시간-도메인-대-주파수-도메인 전환과 양자화/인코딩 사이에서 수행되는 다양한 개념들은, 오디오 콘텐츠(스피치-유사 그리고 논-스피치-유사)의 다른 유형을 위한 성형을 하는데, 좋은 인코딩 효율과 낮은 왜곡 노이즈를 갖도록 한다.

[0055] 2.도 3에 따른 오디오 인코더

[0056] 다음에서, 본 발명의 다른 실시예에 따른 오디오 인코더(300)는 도 3을 참조하여 설명될 것이다. 도 3은 오디오 인코더(300)에 대한 블록 구성도를 나타낸다. 오디오 인코더(300)는 도 2에서 보여지는 블록 구성도의 참조 오디오 인코더(200)의 개선된 버전임을 알 수 있다.

[0057] 2.1 도 2에 따른, 참조 오디오 신호 인코더

[0058] 다시 말해, 도 3에 따른 오디오 인코더(300)에 대한 이해를 용이하게 하도록, 참조 통합-스피치-앤드-오디오-코딩 인코더(USAC 인코더)(200)가 도 2에 나타난 USAC 인코더의 블록 기능도에 참조하여 먼저 설명되어질 것이다. 참조 오디오 인코더(200)는 일반적으로 시간-도메인 표현인 오디오 콘텐츠의 입력 표현(210)을 수신하고, 그것에 기초하여, 오디오 콘텐츠의 인코딩된 표현(212)을 제공하도록 구성된다. 예를 들어, 오디오 인코더(200)는, 주파수-도메인 인코더(230) 그리고/또는 선형-예측-도메인 인코더(240)에 오디오 콘텐츠의 입력 표현(210)을 제공하도록 구성되는 스위치 또는 분배기(220)를 포함한다. 주파수-도메인 인코더(230)는 오디오 콘텐츠의 입력

표현(210')을 수신하고, 그것에 기초하여, 인코딩된 스펙트럼 표현(232)과 스케일 팩터 정보(234)를 제공하도록 구성된다. 선형-예측-도메인 인코더(240)는 입력 표현(210')을 수신하고, 그것에 기초하여, 인코딩된 여기(excitation)(242)와 인코딩된 LPC-필터 계수 정보(244)를 제공한다. 주파수-도메인 인코더(230)는, 예를 들어, 오디오 콘텐츠의 스펙트럼 표현(230b)을 제공하는 변형된-이산-코사인 변환 시간-도메인-대-주파수-도메인 컨버터(230a)를 포함한다. 주파수-도메인 인코더(230)는, 또한, 오디오 콘텐츠의 스펙트럼 마스킹과 시간적-마스킹을 분석하고, 스케일 팩터(230d)와 인코딩된 스케일 팩터 정보(234)를 제공하도록 하는 음향-심리 분석기(230c)를 포함한다. 주파수-도메인 인코더(230)는, 또한, 스케일 팩터들(230d)에 따라 시간-도메인-대-주파수-도메인 컨버터(230a)에 의해 제공되는 스펙트럼 값들을 스케일(scale)하도록 구성되는 스케일러(230e)를 포함한다. 그리하여 오디오 콘텐츠의 스케일링된 스펙트럼 표현(230f)을 얻을 수 있다. 주파수-도메인 인코더(230)는, 또한, 오디오 콘텐츠의 스케일링된 스펙트럼 표현(230f)을 양자화하도록 구성되는 양자화기(230g)와, 양자화기(230g)에 의해 제공되는 오디오 콘텐츠의 양자화된 스케일링된 스펙트럼 표현을 엔트로피-코딩하도록 구성된 엔트로피 코더(230h)를 포함한다. 엔트로피 코더(230h)는 결과적으로 인코딩된 스펙트럼 표현(232)을 제공한다.

[0059] 선형-예측-도메인 인코더(240)는, 인코딩된 여기(242)와 입력 오디오 표현(210')에 기초하는 인코딩된 LPC-필터 계수 정보(244)를 제공하도록 구성된다. LPD 코더(240)는 LPC-필터 계수들(240b)과 오디오 콘텐츠의 입력 표현(210')에 기초하는 인코딩된 LPC-필터 계수 정보(244)를 제공하도록 구성된 선형-예측 분석기(240a)를 포함한다. LPD 코더(240)는, 또한, 두 개의 병렬적인 브랜치(branch)들인 TCX 브랜치(250)와 ACELP 브랜치(260)를 포함하는 여기 인코딩을 포함한다. 브랜치들은 교환될 수 있는데(예를 들어, 스위치(270)를 이용하여), 변환-코딩된-여기(252) 또는 대수-코딩된-여기(262)를 제공한다. TCX 브랜치(250)는 오디오 콘텐츠의 입력 표현(210')과 LP 분석(240a)에 의해 제공된 LPC-필터 계수들(240b)을 모두 수신하도록 구성된 LPC-기반 필터(250a)를 포함한다. LPC-기반 필터(250a)는, 오디오 콘텐츠의 입력 표현(210')과 충분히 비슷한 출력 신호를 제공하기 위한 LPC-기반 필터에 의해 요구되는 여기를 설명하는 필터 출력 신호(250b)를 제공한다. TCX 브랜치는, 또한, 자극(stimulus) 신호(250d)를 수신하고, 그것에 기초하여, 자극 신호(250b)의 주파수-도메인 표현(250d)을 제공하도록 구성된 변형된-이산-코사인-변환(MDCT)를 포함한다. TCX 브랜치는, 또한, 주파수-도메인 표현(250b)을 수신하고, 그것의 양자화된 버전(250f)을 제공하도록 구성된 양자화기(250e)를 포함한다. TCX 브랜치는, 또한, 자극 신호(250b)의 주파수-도메인 표현(250d)의 양자화된 버전(250f)을 수신하고, 그것에 기초하여, 변환-코딩된 자극 신호(252)를 제공하도록 구성된 엔트로피-코더(250g)를 포함한다.

[0060] ACELP 브랜치(260)는 LP 분석(240a)에 의해 제공되는 LPC 필터 계수들(240b)을 수신하고, 또한, 오디오 콘텐츠의 입력 표현(210')을 수신하도록 구성되는 LPC-기반 필터(260a)를 포함한다. LPC-기반 필터(260a)는, 그것에 기반하여, 예를 들어, 오디오 콘텐츠의 입력 표현(210')과 충분히 비슷한 복원 신호를 제공하기 위해서 디코더-측의 LPC-기반 필터에 의해 요구되는 자극을 설명하는 자극 신호(260b)를 제공하도록 구성된다. ACELP 브랜치(260)는, 또한, 적절한 대수 코딩 알고리즘을 사용하는 자극 신호(260b)를 인코딩하도록 구성된 ACELP 인코더(260c)를 포함한다.

[0061] 상기를 요약하면, 스위칭 오디오 코덱에서, 비슷한, 예를 들어, MPEG-D에 따른 오디오 코덱은 스피치와 오디오 코딩 워킹 드래프트(USAC)를 통합하며, 이는 참조 [1]에 설명되는데, 입력 신호의 근접 부분들은 다른 코더들에 의해 프로세싱될 수 있다. 예를 들어, 스피치와 오디오 코딩 워킹 드래프트(USAC WD)의 통합에 따른 오디오 코덱은, 예를 들어, 참조 [2]에서 설명된, 소위 진보된 오디오 코딩(ACC)에 기반한 주파수-도메인 코더와, 예를 들어, 참조 [3]에서 설명되는, 소위 AMR-WB + 개념에 기반하는, TCX와 ACELP와 같은 선형-예측-도메인(LP) 코더들 사이에서 전환(switch)될 수 있다. USAC 인코더는 도 2에 도시된다.

[0062] 다른 코더들 사이의 전이의 디자인은 다른 코더들 사이에서 끊김 없이 전환되는 데 있어 중요하고 또는 필수적인 관심사라는 것이 발견되었다. 교환되는 구조에서 수집한 코딩 기술의 다른 성질 때문에 이와 같은 전이들을 달성하기가 어렵다는 것을 또한 발견하였다. 그러나 다양한 코더들에 의해 공유되는 일반적인 틀들은 전이를 쉽게 할 수 있다는 것을 발견하였다. 도 2에 따른 참조 오디오 인코더(200)에 지금 참조하면, 그것은 USAC에서 그것이 보여질 수 있다. 주파수-도메인 코더(230)는, 신호-도메인에서 변형된 이산 코사인 변환(MDCT)를 계산하고, 반면에, 변형된-코딩된 여기 브랜치(TCX)는, LPC 잔차 도메인(LPC 잔차(250b)를 이용하여)에서 변형된-이산-코사인-변환(MDCT 250c)를 계산한다. 또한, 두 개의 코더들(즉, 주파수-도메인 코더(230)와 TCX 브랜치(250))은 다른 도메인에 적용되면서, 필터 뱅크의 다른 종류를 공유한다. 이리하여, 참조 오디오 인코더(200)(USAC 오디오 인코더일 수 있는)는 MDCT에 커다란 특징들을 완전히 활용할 수 없고, 특히, 하나의 코더(예를 들어, 주파수-도메인 코더(230))로부터 다른 코더(예를 들어, TCX 코더(250))로 갈 때 시간-도메인-에일리어싱 제거(TDAC)를 활용하지 못한다.

- [0063] 도 2에 따른 참조 오디오 인코더(200)를 다시 참조하면, TCX 브랜치(250)와 ACELP 브랜치(260)는 선형 예측적인 코딩(LPC) 틀을 공유하고 있는 것으로 보여질 수 있다. 그것은 소스 모델 코더인 ACELP을 위한 중요한 특징이고, 여기서, LPC는 스피치의 발성의 관(vocal tract)을 모델링하기 위해 사용된다. TCX를 위하여, LPC는 MDCT 계수들(250d)에 도입된 양자화 노이즈를 성형하기 위해 사용된다. 그것은 MDCT(250c)를 수행하기 전에 시간-도메인에서 입력 신호(210')를 필터링함으로써(예를 들어, LPC-기반의 필터(250a))를 사용하여 이루어진다. 게다가, LPC는 ACELP의 적응적인 코드북에 반영되는 여기 신호를 얻음에 의하여, ACELP에서 전이 동안에 TCX안에서 사용되어진다. 추가적으로, 후속 ACELP 프레임을 위한 계수들의 보간된 LPC 세트들을 얻을 수 있게 허락한다.
- [0064] 2.2. 도 3에 따른 오디오 신호 인코더
- [0065] 다음으로, 도 3에 따른 오디오 신호 인코더(300)가 설명될 것이다. 이를 목적으로, 참조는 도 2에 따른 참조 오디오 신호 인코더(200)로 만들어질 것이고, 도 3에 따른 오디오 신호 인코더(300)는 도 2에 따른 오디오 신호 인코더(200)와 몇몇의 유사성을 가지고 있다.
- [0066] 오디오 신호 인코더(300)는 오디오 콘텐츠의 입력 표현(310)을 수신하고, 그리고, 그것에 기초하여, 오디오 콘텐츠의 인코딩된 표현(312)을 제공하도록 구성된다. 오디오 신호 인코더(300)는, 주파수 도메인 코더(230)에 의해 제공되는 오디오 콘텐츠의 부분의 인코딩된 표현인 주파수-도메인 모드와, 선형 예측-도메인 코더(340)에 의해 제공되는 오디오 콘텐츠의 부분의 인코딩된 표현인 선형-예측 모드 사이에서 전환될 수 있도록 구성된다. 다른 모드들에서 인코딩된 오디오 콘텐츠의 부분들은 몇몇의 실시예에서 오버랩핑될 수 있고, 다른 실시예에서 논-오버랩핑될 수 있다.
- [0067] 주파수-도메인 코더(330)는 주파수-도메인 모드에서 인코딩되는 오디오 콘텐츠의 부분을 위한 오디오 콘텐츠의 입력 표현(310')을 수신하고, 그것에 기초하여, 인코딩된 스펙트럼 표현(332)을 제공한다. 선형-예측 도메인 코더(340)는 선형-예측 모드에서 인코딩되는 오디오 콘텐츠의 부분을 위한 오디오 콘텐츠의 입력 표현(310'')을 수신하고, 그것에 기초하여, 인코딩된 여기(342)를 제공한다. 선택적으로, 스위치(320)는 주파수-도메인 코더(330) 그리고/또는 선형-예측-도메인 코더(340)에 입력 표현(310)을 제공하도록 사용될 수 있다.
- [0068] 주파수-도메인 코더는, 또한, 인코딩된 스케일 팩터 정보(334)를 제공한다. 선형-예측-도메인 코더(340)는 인코딩된 LPC-필터 계수 정보(344)를 제공한다.
- [0069] 출력-측의 멀티플렉서(380)는, 오디오 콘텐츠의 인코딩된 표현(312)으로써, 인코딩된 스펙트럼 표현(332)과 주파수-도메인에서 인코딩되는 오디오 콘텐츠의 부분을 위한 인코딩된 스케일 팩터 정보(334)를 제공하고, 오디오 콘텐츠의 인코딩된 표현(312)으로써, 인코딩된 여기(342)와 선형-예측 모드에서 인코딩된 오디오 콘텐츠의 부분을 위한 인코딩된 LPC 필터 계수 정보(344)를 제공하도록 구성된다.
- [0070] 주파수-도메인 인코더(330)는, 오디오 콘텐츠의 MDCT-변환된-주파수-도메인 표현(330b)을 얻기 위하여, 오디오 콘텐츠의 시간-도메인 표현(310')을 수신하고, 오디오 콘텐츠의 시간-도메인 표현(310')을 변환하는 변형된-이산-코사인-변환(330a)을 포함한다. 주파수-도메인 코더(330)는, 또한, 오디오 콘텐츠의 시간-도메인 표현(310')을 수신하고, 그것에 기초하여, 스케일 팩터(330d)와 인코딩된 스케일 팩터 정보(334)를 제공하도록 하는 음향-심리 분석(330c)을 포함한다. 주파수-도메인 코더(330)는, 또한, 다른 스케일 팩터값으로 오디오 콘텐츠의 MDCT-변환된 주파수-도메인 표현(330b)의 다른 스펙트럼 계수들을 스케일링하기 위하여, 오디오 콘텐츠의 MDCT-변환된 주파수-도메인 표현(330d)에 스케일 팩터들(330e)을 적용하도록 구성된 결합기(330e)를 포함한다. 따라서, 오디오 콘텐츠의 MDCT-변환된 주파수-도메인 표현(330d)의 스펙트럼-성형된 버전(330f)이 얻어지고, 여기서, 스펙트럼-성형은 스케일 팩터들(330d)에 따라 수행되고, 여기서, 비교적 큰 스케일 팩터(330e)와 연관되는 스펙트럼 영역(region)들은, 비교적 더 작은 스케일 팩터들(330e)이 연관된 스펙트럼 지역들보다 강조되어진다. 주파수-도메인 코더(330), 또한, 오디오 콘텐츠의 MDCT-변환된 주파수-도메인 표현(330b)의 스케일링된(스펙트럼-성형된) 버전(330f)을 수신하고, 그것의 양자화된 버전(330h)을 제공하도록 구성되는 양자화기를 포함한다. 주파수-도메인 코더(330)는, 또한, 양자화된 버전(330h)을 수신하고, 그것에 기초하여, 인코딩된 스펙트럼 표현(332)을 제공하도록 구성되는 엔트로피 코더(330i)를 포함한다. 양자화기(330g)와 엔트로피 코더(330i)는 양자화 인코더로서 간주되어질 수 있다.
- [0071] 선형-예측-도메인 코더(340)는 TCX 브랜치(350)와 ACELP 브랜치(360)를 포함한다. 추가적으로, LPD 코더(340)는, TCX 브랜치(350)와 ACELP 브랜치(360)에 의해 일반적으로 사용되는 LP 분석(340a)을 포함한다.

LP 분석(340a)은 LPC-필터 계수들(340b)과 인코딩된 LPC-필터 계수 정보(344)를 제공한다.

- [0072] TCX 브랜치(350)는, MDCT 변환 입력으로써, 시간-도메인 표현(310')을 수신하도록 구성된 MDCT 변환(350a)을 포함한다. 중요하게는, 주파수-도메인 코더의 MDCT(330a)와 TCX 브랜치(350)의 MDCT(350a)는 변환 입력 신호들로서 오디오 콘텐츠의 동일한 시간-도메인 표현의 (다른) 부분들을 수신한다.
- [0073] 따라서, 오디오 콘텐츠의 후속 그리고 오버랩핑된 부분들(예를 들어, 프레임들)은 다른 모드들로 인코딩되며, 주파수 도메인 코더(330)의 MDCT(330a)와 TCX 브랜치(350)의 MDCT(350a)는, 변환 입력 신호로서, 시간적 오버랩을 가진 시간 도메인 표현들을 수신할 수 있다. 다시 말해, 주파수 도메인 코더(330)의 MDCT(330a)와 TCX 브랜치(350)의 MDCT(350a)는, "동일한 도메인에서", 즉 오디오 콘텐츠를 표현하는 둘 다의 시간 도메인 신호들인 변환 입력 신호들을 수신한다. 이것은 오디오 인코더(200)와 대조되며, 여기서, 주파수 도메인 코더(230)의 MDCT(230a)는, TCX 브랜치(250)의 MDCT(250c)가 오디오 콘텐츠 자체의 시간 도메인 표현이 아니라, 신호 또는 여기 신호(250b)의 잔차 시간-도메인 표현을 수신하는데 반하여, 오디오 콘텐츠의 시간 도메인 표현을 수신한다.
- [0074] TCX 브랜치(350)는 이득 값들(350c)을 얻기 위하여, LPC 필터 계수들(340b)을 스펙트럼 도메인으로 변환하도록 구성된 필터 계수 변환기(350b)를 더 포함한다. 필터 계수 변환기(350b)는 때때로 또한, "선형-예측-대-MDCT-컨버터"로써 디자인된다. TCX 브랜치(350)는 또한 오디오 콘텐츠의 MDCT-변환된 표현과 이득값들(350c)을 수신하고, 그것에 기초하여, 오디오 콘텐츠의 MDCT-변환 표현의 스펙트럼 성형된 버전(350e)을 제공하도록 구성된 결합기(350d)를 포함한다. 이러한 목적을 위하여, 결합기(350d)는 스펙트럼 성형된 버전(350e)을 얻기 위하여 이득값(350c)에 따라 오디오 콘텐츠의 MDCT-변환된 표현의 스펙트럼 계수들을 가중한다. 또한, TCX 브랜치(350)는 오디오 콘텐츠의 MDCT-변환된 표현의 스펙트럼 성형된 버전(350e)을 수신하고, 그것의 양자화된 버전(350g)을 제공하도록 구성된 양자화기(350f)를 포함한다. 또한, TCX 브랜치(350)는 인코딩 여기(342)로써 양자화된 표현(350g)의 엔트로피-인코딩된(예를 들어, 산술적으로 인코딩된) 버전을 제공하도록 구성된 엔트로피 인코더(350h)를 포함한다.
- [0075] ACELP 브랜치는 LP 분석(340a)에 의해 제공된 LPC 필터 계수들(340b)과 오디오 콘텐츠의 시간 도메인 표현(310')을 수신하는 LPC 기반 필터(360a)를 포함한다. LPC 기반 필터(360a)는 LPC 기반 필터(260a)로써 동일한 기능을 맡으며, 여기 신호(260b)와 동등한 여기 신호(360b)를 제공한다. 또한, ACELP 브랜치(360)는, ACELP 인코더(260c)와 동등한 ACELP 인코더(360c)를 포함한다. ACELP 인코더(360c)는 ACELP 모드(선형 예측 모드의 서브-모드인)를 이용하여 인코딩되는 오디오 콘텐츠의 부분을 위한 인코딩된 여기(342)를 제공한다.
- [0076] 오디오 인코더(300)의 전체적인 기능성과 관련하여, 오디오 콘텐츠의 부분은 TCX 모드(선형 예측 모드의 첫 번째 서브-모드인) 또는 ACELP 모드(선형 예측 모드의 두 번째 서브-모드인) 안에서 주파수 도메인 모드로 인코딩된다. 만약, 오디오 콘텐츠의 부분이 주파수 도메인 모드 또는 TCX 모드에서 인코딩된다면, 오디오 콘텐츠의 부분은 주파수 도메인 코더의 MDCT(330a) 또는 TCX 브랜치의 MDCT(350a)를 이용하여 주파수 도메인으로 먼저 변환된다. MDCT(330a)와 MDCT(350a) 모두는 오디오 콘텐츠의 시간 도메인 표현으로 작동하고, 심지어는 주파수 도메인 모드와 TCX 모드사이에서 전이가 있을 때 오디오 콘텐츠의 시간 도메인 표현으로 작동한다. 주파수 도메인 모드에서, MDCT 변환기(330a)에 의해 제공되는 주파수 도메인 표현의 스펙트럼 성형은, 음향 심리 분석(330c)에 의해 제공된 스케일 팩터에 따라 수행되고, 그리고 TCX 모드에서, MDCT(350a)에 의해 제공되는 주파수 도메인 표현의 스펙트럼 성형은 LP 분석(340a)에 의해 제공되는 LPC 필터 계수들에 따라 수행된다. 양자화(330g)는 양자화(350f)와 비슷하거나 동일하고, 엔트로피 인코딩(330i)은 엔트로피 인코딩(350h)과 비슷하거나 동일하다. 또한, MDCT 변환(330a)은 MDCT 변환(350a)과 비슷하거나 동일하다. 그러나 MDCT 변환의 다른 차원들은 주파수 도메인 코더들(330)과 TCX 브랜치(350) 안에서 이용될 수 있다.
- [0077] 게다가, LPC 필터 계수들(340b)은 TCX 브랜치(350)와 ACELP 브랜치(360)에 의해 모두 사용될 수 있다. 이것은 TCX 모드에서 인코딩되는 오디오 콘텐츠의 부분들과 ACELP 모드에서 인코딩되는 오디오 콘텐츠의 부분들 사이에서 전이들(transitions)을 가능하게 한다.
- [0078] 상기를 요약하면, 본 발명의 실시예는, 통합 스피치와 오디오 코딩(USAC)의 문맥에서, 시간 도메인에서 TCX의 MDCT(350a)를 수행하고, 주파수 도메인(결합기(350d))에서 LPC-기반 필터링을 적용하는 것으로 구성된다. LPC 분석(예를 들어, LP 분석(340a)은 전처림(예를 들어, 오디오 신호 인코더(200)) 수행되고, 계수들(예를 들어, 계수들(340b))은 보통과(예를 들어, 인코딩된 LPC 필터 계수들(344)의 형태로) 마찬가지로 여전히 전송된다. 그러나 노이즈 성형은 더 이상 시간 도메인에서 필터링을 적용하지 않고, 주파수 도메인에서(예를 들어, 결합기(350d)에 의해 수행되는) 가중하는 것을 적용하여 행해진다. 주파수 도메인에서 노이즈 성형은 LPC 계수들(예를

들어, LPC 필터 계수들(340b))을 MDCT 도메인(필터 계수 변환기(350b)에 의해 수행되는)으로 전환함으로써 완료된다. 자세히, 도 3을 참조하면, 주파수 도메인에서 TCX의 LPC-기반 노이즈 성형의 적용에 대한 개념을 보여준다.

[0079] 2.3 LPC 계수들의 계산과 적용에 관한 세부 사항들

[0080] 다음으로, LPC 계수들의 계산과 적용이 설명된다. 첫 번째, LPC 계수들의 적절한 세트는 예를 들어, LPC 분석(340a)을 이용하여, 현재의 TCX 윈도우를 위하여 계산된다. TCX 윈도우는 TCX 모드에서 인코딩되는 오디오 콘텐츠의 시간 도메인 표현의 윈도우잉된 부분일 수 있다. LPC 분석 윈도우들은, 도 4에 나타난 바와 같이, LPC 코더 프레임들의 끝 경계에 위치된다.

[0081] 도 4를 참조하면, TCX 프레임, 즉, TCX 모드에서 인코딩되는 오디오 프레임이 나타난다. 가로 좌표(410)는 시간을 나타내고, 세로 좌표(420)는 윈도우 함수의 크기값들을 나타낸다.

[0082] 보간은 TCX 윈도우의 무게 중심에 상응하는 계수들(340b)의 LPC 세트를 계산하기 위하여 행해진다. 보간은 임피던스(impedance) 스펙트럼 주파수(ISF 도메인)에서 수행되고, 여기서, LPC 계수들은 보통으로 양자화되고 코딩된다. 보간된 계수들은 사이즈 sizeR+sizeM+sizeL의 TCX 윈도우의 중앙에서 집중된다.

[0083] 자세히, 도 4를 참조하면, TCX 윈도우를 위한 LPC 계수들 보간의 예를 보여준다.

[0084] 보간된 LPC 계수들은, 음향 심리적 고려를 가지고 적절한 노이즈 성형 인라인(inline)을 얻기 위하여, TCX(자세하게, 참조[3]을 보라)에서 행해진 것처럼 가중된다. 얻어진 보간되고 가중된 LPC 계수들(또한 간단하게 lpc_coeffs를 가지고 디자인된)은, 도 5 및 6에 그 의사코드(pseudo code)가 도신된, 방법을 이용하여 MDCT 스케일 팩터들(또한, 선형 예측 모드 이득값들로 디자인된)로 결국 전환된다.

[0085] 도 5는 입력 LPC 계수들("lpc_coeffs")에 기반하여 MDCT 스케일 팩터들("mdct_scaleFactor")을 제공하기 위한 함수 "LPC2MDCT"의 의사(pseudo) 프로그램 코드를 나타낸다. 보여질 수 있는 것처럼, 함수 "LPC2MDCT"는, 입력 변수로써, LPC 계수들 "lpc_coeffs", LPC 명령값 "lpc_order" 과 윈도우 사이즈값 "sizeR", "sizeM", "sizeL"을 수신한다. 첫 번째 단계로, 배열 "InRealData[i]"의 성분들은 참조 번호(510)에 나타난 것처럼, LPC 계수들의 변환된 버전으로 채워진다. 보여진 것처럼, 0과 lpc_order-1 사이의 인덱스를 가진 배열 "InRealData"의 성분들과 배열 "InImagData"의 성분들은, LPC 계수 "lpcCoeffs[i]"에 상응하여 결정되고 코사인텀(cosine term) 또는 사인텀(sine term)에 의해 변형된 값들로 설정된다. 인덱스 $i \geq \text{lpc_order}$ 를 가진 배열 "InRealData"와 "InImagData"는 0으로 설정된다.

[0086] 따라서, 배열들 "InRealData[i]"와 "InImagData[i]"는 복소 변형 텀 $(\cos(i \cdot \pi/\text{sizeN}) - j \cdot \sin(i \cdot \pi/\text{sizeN}))$ 을 가지고 변형된 LPC 계수들에 의해 표시되는 시간 도메인 응답의 실수부와 허수부를 표시한다.

[0087] 다음으로, 복소 고속 푸리에 변환(complex fast Fourier transform)이 적용되고, 여기서, 배열들 "InRealData[i]"와 "InImagData[i]"는 복소 고속 푸리에 변환의 입력 신호로 표현된다. 복소 고속 푸리에 변환의 결과는 배열 "OutRealData"와 "OutImagData"에 의해 제공된다. 이리하여, 배열들 "OutRealData"와 "OutImagData"는, 시간 도메인 필터 계수들에 의해 표현되는 LPC 필터 응답을 나타내는 스펙트럼 계수들(주파수 인덱스 i를 가진)을 표현한다.

[0088] 다음으로, 주파수 인덱스 i를 가지며, "mdct_scaleFactors[i]"로 표시되는 이른바 MDCT 스케일 팩터들이 계산된다. MDCT 스케일 팩터 "mdct_scaleFactors[i]"는, 스펙트럼 계수들(성분들 "OutRealData[i]"와 "OutImagData[i]"에 의해 표현되는)에 상응하는 절대값의 인버스(inverse)로써 계산된다.

[0089] 참조 번호(510)에 나타나는 복소-값인 변형 연산과, 참조 번호(520)에 나타나는 복소 고속 푸리에 변환의 실행은 오드 이산 푸리에 변환(odd Fourier transform)(ODFT)을 효과적으로 구성한다. 오드 이산 푸리에 변환은 다음의 공식을 가진다.

[0090]
$$X_0(k) = \sum_{n=0}^{n=N} x(n)e^{-j\frac{2\pi}{N}\left(k+\frac{1}{2}\right)n}$$

- [0091] 여기서, $N=sizeN$ 이고, MDCT의 사이즈의 두 배이다.
- [0092] 위의 공식에서, LPC 계수들 $lpc_coeffs[n]$ 는 변환 입력 함수 $x(n)$ 의 역할을 수행한다. 출력 함수 $X_0(k)$ 는 "OutRealData[k]"(실수부)와 "OutImagData[k]"(허수부) 값들에 의해 표현된다.
- [0093] 함수 "complex_fft()"는 종래의 복소 이산 푸리에 변환(DFT)의 고속 실행이다. 얻어진 MDCT 스케일 팩터들 ("mdct_scaleFactors")은 입력 신호의 MDCT 계수들(MDCT(350a)에 의해 제공되는)을 스케일링하는 양수값들이다. 스케일링은 도 6에 나타난 의사-코드에 따라 수행될 것이다.
- [0094] 2.4 윈도우잉과 오버랩핑에 관한 세부 사항들
- [0095] 후속의 프레임들 사이에서 윈도우잉과 오버랩핑은 도 7과 8에서 설명된다.
- [0096] 도 7은 오버헤드로서 LPC0을 보내는 교환된 시간-도메인/주파수-도메인 코덱에 의해 수행되는 윈도우잉을 나타낸다. 도 8은 주파수 도메인 코더로부터 전이로서 "lpc2mdct"를 사용하는 시간 도메인 코더로 전환할 때 수행되는 윈도우잉을 나타낸다.
- [0097] 도 7을 참조하면, 첫 번째 오디오 프레임(710)은 주파수-도메인 모드에서 인코딩되고, 윈도우(712)를 사용하여 윈도우잉된다.
- [0098] 대략 50%로 첫 번째 오디오 프레임(710)을 오버랩하고, 주파수-도메인 모드에서 인코딩되는 두 번째 오디오 프레임(716)은, "start window"로서 표시되는 윈도우(718)를 사용하여 윈도우잉된다. 시작 윈도우는 긴 좌측 전이 슬로프(718a)와 짧은 우측 전이 슬로프(718c)를 가진다.
- [0099] 선형 예측 모드에서 인코딩되는 세 번째 오디오 프레임(722)은, 우측 전이 슬로프(718c)에 매칭하는 짧은 좌측 전이 슬로프(724a)와 짧은 우측 전이 슬로프(724c)를 포함하는 선형 예측 모드 윈도우(724)를 이용하여 윈도우잉된다. 주파수 도메인 모드에서 인코딩되는 네 번째 오디오 프레임(728)은 비교적 짧은 좌측 전이 슬로프(730a)와 비교적 긴 우측 전이 슬로프(730c)를 가진 "stop window"(730)를 이용하여 윈도우잉된다.
- [0100] 주파수 도메인 모드에서 선형 예측 모드로 전이될 때, 즉, 두 번째 오디오 프레임(716)과 세 번째 오디오 프레임(722) 사이에서 전이로써, LPC 계수들(또는 "LPC0"으로서 표시되는)의 추가 세트는 통상 적절한 전이를 보장하기 위하여 선형 예측 도메인 코딩 모드로 보내진다.
- [0101] 그러나, 본 발명의 실시예는 주파수 도메인 모드와 선형 예측 모드사이에서 전이를 위한 시작 윈도우의 새로운 타입을 가진 오디오 인코더를 제공한다. 도 8을 참조하면, 첫 번째 오디오 프레임(810)은 이른바 "long window"(812)를 이용하여 윈도우잉되고, 주파수 도메인 모드에서 인코딩되는 것을 볼 수 있다. "long window"(812)는 비교적 긴 우측 전이 슬로프(812b)를 포함한다. 두 번째 오디오 프레임(816)은, 윈도우(812)의 우측 전이 슬로프(812b)와 매칭하는 비교적 긴 좌측 전이 슬로프(818a)를 포함하는 선형 예측 도메인 시작 윈도우(818)를 이용하여 윈도우잉된다. 또한, 선형 예측 도메인 시작 윈도우(818)는 비교적 짧은 우측 전이 슬로프(818b)를 포함한다. 두 번째 오디오 프레임(816)은 선형 예측 모드에서 인코딩된다. 따라서, LPC 필터 계수들은 두 번째 오디오 프레임(816)을 위하여 결정되고, 두 번째 오디오 프레임(816)의 시간 도메인 샘플들은 MDCT를 이용하여 스펙트럼 표현으로 변환된다. 두 번째 오디오 프레임(816)을 위하여 결정되는 LPC 필터 계수들은 주파수 도메인에 적용되고, 오디오 콘텐츠의 시간 도메인 표현에 기초하여 MDCT에 의해 제공되는 스펙트럼 계수들을 스펙트럼 성형하는 데 이용된다.
- [0102] 세 번째 오디오 프레임(822)은 전에 설명한 윈도우(724)와 동일한 윈도우(824)를 이용하여 윈도우잉된다. 세 번째 오디오 프레임(822)은 선형 예측 모드에서 인코딩된다. 네 번째 오디오 프레임(828)은 윈도우(730)와 실질적으로 동일한 윈도우(830)를 이용하여 윈도우잉된다.
- [0103] 도 8을 참조하여 설명되는 개념은, 이른바 "long window"를 이용하는 주파수 도메인 모드에서 인코딩되는 오디오 프레임(810)과, 윈도우(824)를 이용하는 선형 예측 모드에서 인코딩되는 세 번째 오디오 프레임(822) 사이의 전이가, 윈도우(818)를 이용하는 선형 예측 모드에서 인코딩되는 중간(부분적으로 오버랩핑) 두 번째 오디오 프레임(816)을 통하여 만들어진다는 장점을 가지고 있다. 두 번째 오디오 프레임은, 스펙트럼 성형이 주파수 도메인(즉, 필터 계수 변환기(350b)를 이용하여)에서 수행되도록 전형적으로 인코딩되는 것처럼, 비교적 긴 우측 전이 슬로프(812b)를 가진 윈도우를 이용하는 주파수 도메인에서 인코딩되는 오디오 프레임(810)과 두 번째 오디오 프레임(816) 사이에서 좋은 오버랩-및-가산이 얻어질 수 있다. 추가적으로, 인코딩된 LPC 필터 계수들은

스케일 팩터값들을 대신하여 두 번째 오디오 프레임(816)을 위하여 전송되어진다. 이것은 도 7의 전이로부터 도 8의 전이를 구별하고, 여기서, 추가 LPC 계수들(LPC0)은 스케일 팩터값에 추가하여 전송된다. 결과적으로, 두 번째 오디오 프레임(816)과 세 번째 오디오 프레임(822) 사이의 전이는, 예를 들어, LPC0 계수들이 도 7의 경우에서 전송되는 것과 같은 추가적인 추가 데이터를 전송하지 않고 좋은 품질로 수행될 수 있다. 이리하여, 세 번째 오디오 프레임(822)에서 이용되는 선형 예측 도메인 코덱을 초기화하는데 요구되는 정보는 추가 정보를 전송하지 않고 이용가능하다.

[0104] 요약하면, 도 8에 참조하여 설명된 실시예에서, 선형 예측 도메인 시작 윈도우(818)는 일반적인 스케일 팩터들(예를 들어 오디오 프레임(716)을 위하여 전송되는)을 대신하여 LPC-기반의 노이즈 성형을 이용할 수 있다. LPC 분석 윈도우(818)는 시작 윈도우(718)에 상응하고, 도 8에서 표현된 바와 같이, 추가적인 설정 LPC 계수들(예를 들어 LPC0 계수들과 같은)이 보내질 필요가 없다. 이러한 경우에서, ACELP(세 번째 오디오 프레임(822)의 적어도 하나의 부분에서 인코딩하기 위해 이용되는)의 적응적인 코드북은, 디코딩된 선형 예측 도메인 코더 시작 윈도우(818)의 계산된 LPC 잔차를 가지고 쉽게 채워질 수 있다.

[0105] 상기를 요약하면, 도 7은 오버헤드로써 LP0로 불리는 LPC 계수 세트의 추가 세트를 보낼 필요가 있는 전환된 시간 도메인/주파수 도메인 코덱의 함수를 보여준다. 도 8은 주파수 도메인 코더로부터 전이를 위하여 이른바 "LPC2MDCT"를 이용하는 선형 예측 도메인 코더로 전환을 나타낸다.

[0106] 3. 도 9에 따른 오디오 신호 인코더

[0107] 다음으로, 오디오 신호 인코더(900)는 도 8을 참조하여 설명되는 개념을 실행하도록 적용되는, 도 9에 참조하여 설명되어질 것이다. 도 9에 따른 오디오 신호 인코더(900)는 도 3에 따른 오디오 신호(300)와 매우 유사하며, 동일한 수단들과 신호들이 동일한 참조 숫자로 표시된다. 이와 같은 동일한 수단들과 신호들의 논의가 여기서는 생략되며, 그리고 참조는 오디오 신호 인코더(300)의 논의로 만들어진다.

[0108] 그러나, 오디오 신호 인코더(900)는, 주파수 도메인 코더(930)의 결합기(330e)가 스펙트럼 성형을 위하여 선택적으로 스케일 팩터들(340d) 또는 선형 예측 도메인 이득값들(350c)을 적용할 수 있다는 점에서 오디오 신호 인코더(300)와 비교하여 확장된다. 이러한 목적을 위하여, 스위치(930j)가 사용되며, 이는 스펙트럼 계수들(330b)의 스펙트럼 성형을 위한 결합기(330e)에 스케일 팩터들(330d) 또는 선형 예측 도메인 이득값들(350c)을 제공하도록 한다. 이리하여, 오디오 신호 인코더(900)는 심지어 실행의 세 가지 모드를 알고 있다. 즉,

[0109] 1. 주파수 도메인 모드: 오디오 콘텐츠의 시간 도메인 표현은 MDCT(330a)를 이용하여 주파수 도메인으로 변환되고, 스펙트럼 성형은 스케일 팩터들(330d)에 따라 오디오 콘텐츠의 주파수 도메인 표현(330b)에 적용된다. 스펙트럼 성형된 주파수 도메인 표현(330f)의 양자화되고 인코딩된 버전(332)과 인코딩된 스케일 팩터 정보(334)는, 주파수 도메인 모드를 이용하여 인코딩된 오디오 프레임을 위한 비트스트림 안에 포함된다.

[0110] 2. 선형 예측 모드: 선형 예측 모드에서, LPC 필터 계수들(340b)은 오디오 콘텐츠의 부분을 위하여 결정되고, 변환-코딩된-여기(첫 번째 서브-모드) 또는 ACELP-코딩된 여기는, 코딩된 여기는 비트율을 더욱 효율적으로 함에 따른, 상기 LPC 필터 계수들(340b)을 이용하여 결정된다. 인코딩된 여기(342)와 인코딩된 LPC 필터 계수 정보(344)는 선형 예측 모드에서 인코딩된 오디오 프레임을 위한 비트스트림 안에 포함된다.

[0111] 3. 스펙트럼 성형에 기반한 LPC 필터 계수를 가진 주파수 도메인 모드: 그렇지 않으면, 세 번째 가능한 모드로, 오디오 콘텐츠는 주파수 도메인 코더(930)에 의해 프로세싱될 수 있다. 그러나, 스케일 팩터들(330d)에 대신하여, 선형 예측 도메인 이득값들(350c)은 결합기(330e)에서 스펙트럼 성형을 위해 적용된다. 따라서, 오디오 콘텐츠의 스펙트럼 성형된 주파수 도메인 표현(330f)의 양자화되고 엔트로피 코딩된 버전(332)은 비트스트림 안에 포함되고, 여기서 스펙트럼 성형된 주파수 도메인 표현(330f)은, 선형 예측 도메인 코더(340)에 의해 제공되는 선형 예측 도메인 이득값들(350c)과 일치하도록 스펙트럼 성형된다. 추가적으로, 인코딩된 LPC 필터 계수 정보(344)는 이와 같은 오디오 프레임을 위한 비트스트림에 포함된다.

[0112] 상기한 세 가지 모드를 이용함에 의해, 두 번째 오디오 프레임(816)을 위한 도 8에 참조하여 표현된 전이를 완료하는 것이 가능하다. 만약에, 주파수 도메인 코더(930)에 의해 사용되는 MDCT의 차원이 TCX 브랜치(350)에 의해 이용되는 MDCT 차원에 상응한다면, 그리고, 주파수 도메인 코더(930)에 의해 이용되는 양자화(330g)가 TCX 브랜치(350)에서 이용되는 양자화(350f)에 상응한다면, 그리고, 주파수 도메인 코더에 의해 이용되는 엔트로피

코딩(330e)이 TCX 브랜치에서 이용되는 엔트로피 코딩(330h)에 상응한다면, 여기서, 선형 예측 도메인 이득값들에 따른 스펙트럼 성형을 가진 주파수 도메인 인코더(930)를 이용하는 오디오 프레임의 인코딩은, 선형 예측 도메인 코더를 이용하는 오디오 프레임(816)의 인코딩과 동등하다. 다시 말해, 오디오 프레임(816)의 인코딩은, MDCT(350g)이 MDCT(330a)의 특징을 맡고, 양자화(350f)가 양자화(330e)의 특징을 맡으며, 엔트로피 인코딩(350h)이 엔트로피 인코딩(330i)의 특징을 맡을 수 있도록, TCX 브랜치(350)를 적용함에 의해 행해지거나, 주파수 도메인 코더(930)에서 선형 예측 도메인 이득값(350c)을 적용함에 의해 행해질 수 있다. 두 결과들은 동등하고, 도 8을 참조하여 논의한 바와 같이 시작 윈도우의 프로세싱으로 이끈다.

[0113] 4. 도 10에 따른 오디오 신호 디코더

[0114] 다음으로, 신호 도메인에서 수행되는 TCX MDCT를 가진 USAC(통합된 스피치-및-오디오 코딩)의 통합된 관점이 도 10을 참조하여 설명되어진다.

[0115] 본 발명의 실시예에 따른 TCX 브랜치(350)와 주파수 도메인 코더(330, 930)는 거의 모든 동일한 코딩 틀(MDCT(330a, 350a); 결합기(330e, 350d); 양자화(330g, 350f); 엔트로피 코더(330i, 350h))를 공유하고, 그리고, 도 10에 도시된 바와 같이 단일 코더로서 간주될 수 있다. 이리하여, 본 발명에 따른 실시예들은, 전환된 코더 USAC의 더 통합된 구조에 허용되고, 여기서, 코덱의 단 두 가지 종류(주파수 도메인 코더와 시간 도메인 코더)가 범위가 정해질 수 있다.

[0116] 도 10을 참조하면, 오디오 신호 인코더(1000)는 오디오 콘텐츠의 입력 표현(1010)을 수신하고, 그것에 기초하여 오디오 콘텐츠의 인코딩된 표현(1012)을 제공하도록 구성됨을 볼 수 있다. 일반적으로 시간 도메인 표현인, 오디오 콘텐츠의 입력 표현(1010)은, 만약에 오디오 콘텐츠의 부분이 주파수 도메인 모드에서 또는 선형 예측 모드의 TCX 서브-모드에서 인코딩된다면, MDCT(1030a)에 입력된다. MDCT(1030a)는 시간 도메인 표현(1010)의 주파수 도메인 표현(1030b)을 제공한다. 주파수 도메인 표현(1030b)은, 스펙트럼 성형값(1040)을 가진 주파수 도메인 표현(1030b)을 결합하는 결합기(1030e)에 입력되어, 주파수 도메인 표현(1030b)의 스펙트럼 성형된 버전(1030f)을 얻도록 한다. 스펙트럼 성형된 표현(1030f)은 양자화기(1030g)를 이용하여 양자화되어, 그것의 양자화된 버전(1030h)을 얻도록 하고, 양자화 버전(1030h)은 엔트로피 코더(예를 들어, 산술 인코더)(1030i)에 보내진다. 엔트로피 코더(1030i)는, 스펙트럼 성형된 주파수 도메인 표현(1030f)의 양자화되고 엔트로피 코딩된 표현을 제공하고, 양자화된 인코딩 표현은 1032로 표시된다. MDCT(1030a), 결합기(1030e), 양자화기(1030g), 그리고 엔트로피 인코더(1030i)는 주파수 도메인 모드와 선형 예측 모드의 TCX 서브-모드를 위한 보통의 프로세싱 경로를 형성한다.

[0117] 오디오 신호 인코더(1000)는 ACELP 신호 프로세싱 경로(1060)를 포함하며, 이는 또한 오디오 콘텐츠의 시간 도메인 표현(1010)을 수신하고, 그것에 기초하여, LPC 필터 계수 정보(1040b)를 이용하는 인코딩된 여기(1062)를 제공한다. ACELP 신호 프로세싱 경로(1060)는, 선택적인 것으로 간주될 수 있으며, LPC 기반의 필터(1060a)를 포함하고, 오디오 콘텐츠의 시간 도메인 표현(1010)을 수신하고 잔차 신호와 여기 신호(1060b)를 ACELP 인코더(1060c)로 제공한다. ACELP 인코더는 여기 신호와 잔차 신호(1060b)에 기반한 인코딩된 여기(1062)를 제공한다.

[0118] 또한, 오디오 신호 인코더(1000)는, 오디오 콘텐츠의 시간 도메인 표현(1010)을 수신하고, 그것에 기초하여, 최근 오디오 프레임을 디코딩하기 위하여 요구되는 부가 정보의 인코딩된 버전뿐만 아니라, 스펙트럼 성형 정보(1040a)와 LPC 필터 계수 필터 정보(1040b)를 제공하도록 구성된 보통의 신호 분석기(1070)를 포함한다. 이리하여, 보통의 신호 분석기(1070)는, 만약에 최근 오디오 프레임이 주파수 도메인 모드에서 인코딩된다면, 음향 심리적 분석(1070a)을 이용하여 스펙트럼 성형 정보(1040a)를 제공하며, 만약에 최근 오디오 프레임이 주파수 도메인 모드에서 인코딩된다면, 인코딩된 스케일 팩터 정보를 제공한다. 스펙트럼 성형을 위하여 이용되는 스케일 팩터 정보는 음향 심리적 분석(1070a)에 의하여 제공되고, 스케일 팩터들(1070b)을 표현하는 인코딩된 스케일 팩터 정보는 주파수 도메인 모드에서 인코딩되는 오디오 프레임을 위한 비트스트림(1012)에 포함된다.

[0119] 선형예측 모드의 TCX 서브-모드에서 인코딩되는 오디오 프레임을 위하여, 보통의 신호 분석기(1070)는 선형 예측 분석(1070c)을 이용하여 스펙트럼 성형 정보(1040a)를 도출한다. 선형 예측 분석(1070c)은 선형 예측-대-MDCT 블록(1070d)에 의해 스펙트럼 표현으로 변환되는 LPC 필터 계수들의 세트 안에서 이루어진다. 따라서, 스펙트럼 성형 정보(1040a)는 상기에서 논의된 것처럼 LP 분석(1070c)에 의해 제공되는 LPC 필터 계수들로부터 도출된다. 결과적으로, 선형-예측 모드의 변환-코딩된 여기 서브-모드에서 인코딩되는 오디오 프레임을 위하여, 보통의 신호 분석기(1070)는 선형-예측 분석(1070c)에 기반하는(오히려 음향 심리적 분석(1070a)에 더

기반하여) 스펙트럼 성형 정보(1040a)를 제공하고, 또한, 비트스트림(1012) 안에 포함을 위하여, 인코딩된 스케일-팩터 정보보다 오히려 인코딩된 LPC 필터 계수 정보를 제공한다.

[0120] 게다가, 선형-예측 모드의 ACELP 서브-모드 안에서 인코딩된 오디오 프레임에 대하여, 보통의 신호 분석기(1070)의 선형-예측 분석(1070c)은 LPC 필터 계수 정보(1040b)를 ACELP 신호 프로세싱 브랜치(1060)의 LPC-기반의 필터(1060a)에 제공한다. 이러한 경우에, 보통의 신호 분석기(1070)는 비트스트림(1012) 안에 포함을 위하여 인코딩된 LPC 필터 계수 정보를 제공한다.

[0121] 상기를 요약하면, 동일한 신호 프로세싱 경로는 주파수-도메인 모드와 선형-예측 모드의 TCX 서브-모드를 위하여 이용된다. 그러나, 전에 또는 MDCT와 MDCT(1030a)의 차원의 결합으로 적용되는 윈도우는 인코딩 모드에 따라 다양할 수 있다. 그럼에도 불구하고, 주파수-도메인 모드와 선형-예측 모드의 TCX 서브-모드는, 인코딩된 LPC 필터 계수 정보가 선형-예측 모드에서 비트스트림에 포함되는데 반하여, 인코딩된 스케일-팩터 정보는 주파수-도메인 모드에서 비트스트림에 포함된다는 점에서 다르다.

[0122] 선형-예측 모드의 ACELP 서브-모드에서, ACELP-인코딩된 여기와 인코딩된 LPC 필터 계수 정보는 비트스트림에 포함된다.

[0123] 5. 도 11에 따른 오디오 신호 디코더

[0124] 5.1 디코더 개관

[0125] 다음으로, 오디오 신호 인코더가 설명될 것이며, 이는 상술한 오디오 신호 인코더에 의해 제공된 오디오 콘텐츠의 인코딩된 표현을 디코딩할 수 있다.

[0126] 도 11에 따른 오디오 신호 디코더(1100)는, 오디오 콘텐츠의 인코딩된 표현(1110)을 수신하고, 그것에 기초하여, 오디오 콘텐츠의 디코딩된 표현(1112)을 제공하도록 구성된다. 오디오 신호 인코더(1110)는, 오디오 콘텐츠의 인코딩된 표현(1110)을 포함하는 비트스트림을 수신하고, 상기 비트스트림으로부터 오디오 콘텐츠의 인코딩된 표현을 추출하여 오디오 콘텐츠의 추출된 인코딩된 표현(1110')을 얻도록 구성된 선택적 비트스트림 페이로드 디포맷터(1120)를 포함한다. 선택적 비트스트림 페이로드 디포맷터(1120)는 비트스트림으로부터 인코딩된 스케일-팩터 정보, 인코딩된 LPC 필터 계수 정보 및 추가적인 제어 정보 또는 신호 강화 부가 정보를 추출할 수 있다.

[0127] 또한, 오디오 신호 디코더(1100)는, 오디오 콘텐츠의 복수의 부분들(예를 들어, 오버랩핑 또는 논-오버랩핑 오디오 프레임들)을 위한 디코딩된 스펙트럼 계수들의 복수의 세트들(1132)을 얻도록 구성된 스펙트럼 값 결정기(1130)를 포함한다. 디코딩된 스펙트럼 계수들의 세트들은 프리프로세서(1140)를 이용하여 선택적으로 프리프로세싱될 수 있고, 그렇게 함으로써, 디코딩된 스펙트럼 계수들의 프리프로세싱된 세트(1132')들을 산출할 수 있다.

[0128] 또한, 오디오 신호 디코더(1100)는 스펙트럼 프로세서(1150)를 포함하는데, 스펙트럼 프로세서(1150)는, 선형-예측 모드에서 인코딩된 오디오 콘텐츠(예를 들어, 오디오 프레임)의 부분을 위한 선형-예측-도메인 파라미터들의 세트(1152)에 따라 디코딩된 스펙트럼 계수들의 세트(1132)에, 또는 그것의 프리프로세싱된 버전(1132')에 스펙트럼 성형을 적용하도록 구성되고, 주파수-도메인 모드에서 인코딩된 오디오 콘텐츠(예를 들어, 오디오 프레임)의 부분을 위한 스케일 팩터 파라미터들의 세트(1154)에 따라 디코딩된 스펙트럼 계수(1132)들, 또는 그것의 프리프로세싱된 버전(1132')에 스펙트럼 성형을 적용하도록 구성될 수 있다. 따라서, 스펙트럼 프로세서(1150)는 디코딩된 스펙트럼 계수들의 스펙트럼 성형된 세트들(1158)을 얻는다.

[0129] 또한, 오디오 신호 디코더(1100)는, 디코딩된 스펙트럼 계수들의 스펙트럼 성형된 세트(1158)를 수신하고, 선형-예측 모드에서 인코딩된 오디오 콘텐츠의 부분을 위한 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트(1158)에 기반한 오디오 콘텐츠의 시간-도메인 표현(1162)을 얻도록 구성되는 주파수-도메인-대-시간-도메인 컨버터(1160)를 포함한다. 또한, 주파수-도메인-대-시간-도메인 컨버터(1160)는, 주파수-도메인 모드에서 인코딩된 오디오 콘텐츠의 부분을 위한 디코딩된 스펙트럼 계수들의 각각의 스펙트럼 성형된 세트(1158)에 기초한 오디오 콘텐츠의 시간-도메인 표현(1162)을 얻도록 구성된다.

[0130] 또한, 오디오 신호 디코더(1100)는, 오디오 콘텐츠의 디코딩된 표현(1112)을 얻도록, 오디오 콘텐츠의 시간-도메인 표현(1162)의 시간-도메인 포스트 프로세싱을 선택적으로 수행하는 선택적 시간-도메인 프로세서(1170)를 포함한다. 그러나, 시간-도메인 포스트-프로세서(1170)의 부재에서, 오디오 콘텐츠의 디코딩된 표현(1112)은 주

주파수-도메인-대-시간-도메인 컨버터(1160)에 의해 제공되는 오디오 콘텐츠의 시간-도메인 표현(1162)과 동등할 수 있다.

[0131] 5.2 추가 세부 사항

[0132] 다음으로, 오디오 디코더(1100)의 더욱 세부 사항이 설명되며, 세부 사항들은 오디오 신호 디코더의 선택적 개선으로써 간주될 수 있다.

[0133] 오디오 신호 디코더(1100)는, 다른 모드를 이용하여 인코딩되는 오디오 콘텐츠의 후속 부분들(예를 들어, 오버랩핑 또는 논-오버랩핑 오디오 프레임들)에서 인코딩된 오디오 신호 표현을 다룰 수 있는 멀티-모드 오디오 신호 디코더임을 알 수 있다. 다음으로, 오디오 프레임들은 오디오 콘텐츠의 부분의 간단한 예로서 간주될 것이다. 오디오 콘텐츠가 오디오 프레임들로 세분되는 것처럼, 동일한 모드로 인코딩된 후속의(특히, 부분 오버랩핑 또는 논-오버랩핑) 오디오 프레임들의 디코딩된 표현들 사이에서, 그리고 또한, 다른 모드들에서 인코딩된 후속의(오버랩핑 또는 논-오버랩핑) 오디오 프레임들 사이에서 매끄러운 전이를 가지는 것이 특히 중요하다. 바람직하게, 비록 오버랩핑이 몇몇의 경우에서 그리고/또는 몇몇의 전이를 위하여 상당히 작을 수 있지만, 오디오 신호 디코더(1100)는 대략 50%까지 후속의 오디오 프레임들이 오버랩핑하는 오디오 신호 표현들은 다룬다.

[0134] 이러한 이유에 의하여, 오디오 신호 디코더(1100)는, 다른 모드에서 인코딩된 후속의 오디오 프레임들의 시간-도메인 표현들은 오버랩-및-가산하도록 구성된 오버랩핑을 포함한다. 예를 들어, 오버랩핑은 주파수-도메인-대-시간-도메인 컨버터(1160)의 일부이거나, 주파수-도메인-대-시간-도메인 컨버터(1160)의 출력으로 정렬될 수 있다. 후속 오디오 프레임들을 오버랩핑할 때 고효율과 좋은 품질을 얻기 위하여, 주파수-도메인-대-시간-도메인 컨버터는, 랩핑된(lapped) 변환을 이용하여 선형-예측 모드(예를 들어, 그것의 변환-코딩된-여기서-서브-모드에서)에서 인코딩된 오디오 프레임의 시간-도메인 표현을 얻도록, 그리고 또한, 랩핑된 변환을 이용하는 주파수-도메인 모드에서 인코딩된 오디오 프레임의 시간-도메인-표현을 얻도록 구성된다. 이러한 경우에, 오버랩핑은 다른 모드들에서 인코딩된 후속의 오디오 프레임들의 시간-도메인-표현들을 오버랩하도록 구성된다. 다른 모드들에서 인코딩된 오디오 프레임들을 위한 동일한 변환 타입일 수 있는, 주파수-도메인-대-시간-도메인 전환들을 위한 이와 같은 통합 랩핑된 변환을 이용함에 의해, 중요한 샘플링이 사용될 수 있고, 오버랩-및-가산 연산에 의해 발생한 오버헤드는 최소화된다. 동시에, 후속 오디오 프레임들의 시간-도메인-표현들의 오버랩핑 부분들 사이에서 시간 도메인 에일리어싱(aliasing)이 있다. 다른 모드들로 인코딩된 후속의 오디오 프레임들 사이의 전이에서 시간-도메인 에일리어싱 제거를 가질 가능성은, 주파수-도메인-대-시간-도메인 전환이 다른 모드들에서 동일한 도메인에 적용된다는 사실에 의해 야기되며, 첫 번째 모드로 인코딩되는 첫 번째 오디오 프레임의 디코딩된 스펙트럼 계수들의 스펙트럼 성형된 세트에서 수행되는 합성 랩핑된 변환의 출력은 두 번째 모드로 인코딩되는 후속의 오디오 프레임의 디코딩된 스펙트럼 계수들의 스펙트럼-성형된 세트에서 수행되는 랩핑된 변환의 출력과 직접적으로 결합된다(즉, 중간 필터링 과정없이 결합됨). 이리하여, 첫 번째 모드로 인코딩된 오디오 프레임을 위하여 수행되는 랩핑된 변환의 출력과 두 번째 모드로 인코딩된 오디오 프레임을 위하여 수행되는 랩핑된 변환의 출력의 선형 결합이 수행된다. 당연히, 적절한 오버랩 윈도우는 랩핑된 변환 프로세스의 부분으로써 또는 랩핑된 변환 프로세스의 다음으로 수행될 수 있다.

[0135] 따라서, 시간-도메인 에일리어싱 제거는, 다른 모드에서 인코딩되는 후속의 오디오 프레임들의 시간-도메인 표현들 사이에서 단지 오버랩-및-가산 연산에 의해 얻어진다.

[0136] 다시 말해, 주파수-도메인-대-시간-도메인 컨버터(1160)는, 두 개 모두의 모드를 위한 동일한 도메인에 있는 시간-도메인 출력 신호를 제공하는 것이 중요하다. 주파수-도메인-대-시간-도메인 전환(예를 들어, 연관된 전이 윈도우와 결합한 랩핑된 변환)의 출력 신호들이 다른 모드들을 위해 동일한 도메인에 있다는 사실은, 주파수-도메인-대-시간-도메인 전환의 출력 신호들이 심지어 다른 모드들 사이의 전이에서 선형적으로 결합할 수 있다는 것을 의미한다. 예를 들어, 주파수-도메인-대-시간-도메인 전환의 출력 신호들은, 스피커 신호의 임시적 진화를 표현하는 오디오 콘텐츠의 시간-도메인 표현들이다. 다시 말해, 후속의 오디오 프레임들의 오디오 콘텐츠의 시간-도메인 표현들(1162)은 스피커 신호들을 산출하기 위하여 일반적으로 프로세싱될 수 있다.

[0137] 게다가, 스펙트럼 프로세서(1150)는 파라미터 제공기(1156)를 포함하며, 이는 선형-예측 도메인 파라미터들의 세트(1152)와 비트스트림(1110)으로부터 추출된 정보에 기반한, 예를 들어, 인코딩된 스케일 팩터 정보와 인코딩된 LPC 필터 파라미터 정보에 기반한 스케일 팩터 파라미터들의 세트(1154)를 제공하도록 구성된다. 예를 들어, 파라미터 제공기(1156)는, 선형-예측 모드에서 인코딩된 오디오 콘텐츠의 부분을 위한 LPC 필터 계수들의

인코딩된 표현에 기초한 디코딩된 LPC 필터 계수들을 얻도록 구성되는 LPC 필터 계수 결정기를 포함한다. 또한, 파라미터 제공기(1156)는, 다른 주파수들과 연관된 선형-예측 모드 이득값들을 얻기 위하여, 디코딩된 LPC 필터 계수들을 스펙트럼 표현으로 변환하도록 구성되는 필터 계수 변환기를 포함할 수 있다. 선형-예측 모드 이득값들(때때로, 또한 $g[k]$ 로 표시되는)은 선형-예측 도메인 파라미터들의 세트(1152)를 구성할 수 있다.

[0138] 파라미터 제공기(1156)는, 주파수-도메인 모드에서 인코딩된 오디오 프레임에 대한 스케일 팩터값들의 인코딩된 표현에 기반한 디코딩된 스케일 팩터값들을 얻도록 구성된 스케일 팩터 결정기를 더 포함할 수 있다. 디코딩된 스케일 팩터값들은 스케일 팩터 파라미터들의 세트(1154)로서 제공할 수 있다.

[0139] 따라서, 스펙트럼 변경으로써 간주되는 스펙트럼-성형은, 선형-예측 모드에서 인코딩된 오디오 프레임과 연관된 디코딩된 스펙트럼 계수들의 세트(1132) 또는 그것의 프리프로세싱된 버전(1332')과, 선형-예측 모드 이득값들(선형-예측 도메인 파라미터들의 세트(1152)를 구성하는)을 결합하도록 구성되는데, 이는 디코딩된 스펙트럼 계수들(1132) 또는 그것의 프리-프로세싱된 버전(1132')의 기여가 선형-예측 모드 이득값들에 따라 가중되는 디코딩된 스펙트럼 계수들(1132)의 이득 프로세싱된(즉, 스펙트럼 성형된) 버전(1158)을 얻기 위함이다. 추가적으로, 스펙트럼 변경기는, 주파수-도메인 모드에서 인코딩된 오디오 프레임에 연관된 디코딩된 스펙트럼 계수들의 세트(1132) 또는, 그것의 프리-프로세싱된 버전(1132')과 스케일 팩터 값들(스케일 팩터 파라미터들의 세트(1154)를 구성하는)을 결합하도록 구성될 수 있으며, 이는 디코딩된 스펙트럼 계수들(1132), 또는 그것의 프리-프로세싱된 버전(1132')의 기여가 스케일 팩터값들(스케일 팩터 파라미터들의 세트(1154)의)에 따라 가중되는 디코딩된 스펙트럼 계수들(1132)의 스케일-팩터-프로세싱된 버전(1158)을 얻기 위함이다. 따라서, 스펙트럼 성형의 첫 번째 타입, 즉 선형-예측 도메인 파라미터들의 세트(1152)에 따른 스펙트럼 성형은, 선형-예측 모드에서 수행되고, 그리고 스펙트럼 성형의 두 번째 타입, 즉, 스케일 팩터 파라미터의 세트(1154)에 따른 스펙트럼 성형은 주파수-도메인 모드에서 수행된다. 결과적으로, 시간-도메인-표현(1162)상에 양자화 노이즈의 해로운 영향은 스피치-유사 오디오 프레임들(스펙트럼 성형이 바람직하게는 선형-예측-도메인 파라미터들의 세트(1152)에 따라 수행되는)과, 일반적인 오디오, 예를 들어, 스펙트럼 성형이 바람직하게는 스케일 팩터 파라미터들의 세트(1154)에 따라 수행되도록 논-스피치-유사 오디오 프레임들, 모두를 위해서 작게 유지된다. 그러나, 스피치-유사 그리고 논-스피치-유사 오디오 프레임들 모두를 위하여, 즉, 선형-예측 모드에서 인코딩되는 오디오 프레임들과 주파수 도메인 모드에서 인코딩되는 오디오 프레임을 위한 스펙트럼 성형을 이용하는 노이즈-성형을 수행함에 의하여, 멀티-모드 오디오 디코더(1100)는 낮은-복잡도 구조를 포함하고, 동시에 다른 모드에서 인코딩되는 오디오 프레임의 시간-도메인 표현들(1162)의 에일리어싱-제거하는 오버랩-및-가산을 허용한다.

[0140] 다른 세부 사항들은 아래에서 논의될 것이다.

[0141] 6. 도 12에 따른 오디오 신호 디코더

[0142] 도 12는 본 발명의 추가의 실시예에 따른 오디오 신호 디코더(1200)의 블록 구성도를 나타낸다. 도 12는 신호 도메인에서 변환-코딩된 여기-변경된-이산-코사인-변환(TCX-MDCT)을 가진 통합-스피치-앤드-오디오-코딩(USAC)의 통합적인 관점을 보여준다.

[0143] 도 12에 따른 오디오 신호 디코더(1200)는 비트스트림 페이로드 디포맷터(1120)의 기능을 수행하는 비트스트림 디멀티플렉서(1210)를 포함한다. 비트스트림 디멀티플렉서(1210)는 오디오 콘텐츠를 표현하는 비트스트림으로부터 인코딩된 스펙트럼 값들과 추가적인 정보(예를 들어, 인코딩된 스케일-팩터 정보와 인코딩된 LPC 필터 파라미터 정보)를 포함하는 오디오 콘텐츠의 인코딩된 표현을 추출한다.

[0144] 또한, 오디오 신호 디코더(1200)는, 비트스트림 디멀티플렉서에 의해 제공되는 오디오 콘텐츠의 인코딩된 표현의 요소들을 오디오 신호 디코더(1200)의 다른 요소 프로세싱 블록들에 분배하도록 구성되는 스위치들(1216, 1218)을 포함한다. 예를 들어, 오디오 신호 디코더(1200)는, 스위치(1216)로부터 인코딩된 주파수-도메인 표현(1228)을 수신하고, 그것에 기초하여, 오디오 콘텐츠의 시간-도메인 표현(1232)을 제공하는 결합된 주파수-도메인-모드/TCX 서브-모드 브랜치(1230)를 포함한다. 또한, 오디오 신호 디코더(1200)는 스위치(1216)로부터 ACELP-인코딩된 여기 정보(1238)를 수신하고, 그것에 기초하여, 오디오 콘텐츠의 시간-도메인 표현(1242)을 제공하도록 구성된 ACELP 디코더(1240)를 포함한다.

[0145] 또한, 오디오 신호 디코더(1200)는, 주파수-도메인 모드에서 인코딩된 오디오 프레임을 위한 인코딩된 스케일-팩터 정보(1254)와, TCX 서브-모드와 ACELP 서브-모드를 포함하는 선형-예측 모드에서 인코딩된 오디오 프레임을 위한 인코딩된 LPC 필터 계수 정보(1256)를 스위치(1218)로부터 수신하도록 구성된 파라미터 제공기(1260)를

포함한다. 파라미터 제공기(1260)는 스위치(1218)로부터 컨트롤 정보(1258)를 수신하도록 더 구성된다. 파라미터 제공기(1260)는 결합된 주파수-도메인 모드/TCX 서브-모드 브랜치(1230)를 위하여 스펙트럼-성형 정보(1262)를 제공하도록 구성된다. 추가적으로, 파라미터 제공기(1260)는 LPC 필터 계수 정보(1264)를 ACELP 디코더(1240)에 제공하도록 구성된다.

[0146] 결합된 주파수 도메인 모드/TCX 모드 브랜치(1230)는 인코딩된 주파수 도메인 정보(1228)를 수신하고, 그것에 기초하여, 역 양자화기(1230c)로부터 제공받은 디코딩된 주파수 도메인 정보(1230b)를 제공하는 엔트로피 디코더(1230a)를 포함한다. 역 양자화기(1230c)는, 디코딩된 주파수 도메인 정보(1230b)에 기반하여, 예를 들어, 디코딩된 스펙트럼 계수들의 세트의 형태로, 디코딩되고 역 양자화된 주파수 도메인 정보(1230d)를 제공한다. 결합기(1230e)는, 스펙트럼-성형된 주파수 도메인 정보(1230f)를 얻도록, 디코딩되고 역 양자화된 주파수 도메인 정보(1230d)를 스펙트럼 성형 정보(1262)와 결합하도록 구성된다. 역 변경된-이산-코사인-변환(1230g)은, 스펙트럼 성형된 주파수 도메인 정보(1230f)를 수신하고, 그것에 기초하여, 오디오 콘텐츠의 시간 도메인 표현(1232)을 제공한다.

[0147] 엔트로피 디코더(1230a), 역 양자화기(1230c) 및 역 변경된 이산 코사인 변환(1230g)은, 비트스트림에 포함되고, 파라미터 제공기(1260)에 의하여 비트스트림으로부터 산출되는 몇몇의 컨트롤 정보를 선택적으로 수신할 수 있다.

[0148] 파라미터 제공기(1260)는 인코딩된 스케일 팩터 정보(1254)를 수신하고, 디코딩된 스케일 팩터 정보(1260b)를 제공하는 스케일 팩터 디코더(1260a)를 포함한다. 또한, 파라미터 제공기(1260)는 인코딩된 LPC 필터 계수 정보(1256)를 수신하고, 그것에 기초하여, 디코딩된 LPC 필터 계수 정보(1260d)를 필터 계수 변환기(1260e)에 제공하도록 구성되는 LPC 계수 디코더(1260c)를 포함한다. 또한, LPC 계수 디코더(1260c)는 LPC 필터 계수 정보(1264)를 ACELP 디코더(1240)에 제공한다. 필터 계수 변환기(1260e)는 LPC 필터 계수들(1260d)을 주파수 도메인(또한, 스펙트럼 도메인으로 표시된)으로 변환하고, 다음으로 LPC 필터 계수들(1260d)로부터 선형 예측 모드 이득값들(1260f)을 산출하도록 구성된다. 또한, 파라미터 제공기(1260)는, 예를 들어, 스위치(1260g)를 이용하여, 스펙트럼 성형 정보(1262)로써 디코딩된 스케일 팩터들(1260b) 또는 선형 예측 모드 이득값(1260f)을 선택적으로 제공하도록 구성된다.

[0149] 도 12에 따른 오디오 신호 인코더(1200)는, 스테이지들 사이에서 순환되는 추가적인 프리프로세싱 단계들과 포스트-프로세싱 단계들의 수에 의해 보충될 수 있다. 프리프로세싱 단계들과 포스트-프로세싱 단계들은 모드의 차이로 다를 수 있다.

[0150] 몇몇의 세부 사항들이 다음에 설명되어질 것이다.

[0151] 7. 도 13에 따른 신호 흐름

[0152] 다음으로, 가능한 신호 흐름이 도 13을 참조하여 설명될 것이다. 도 13에 따른 신호 흐름(1300)은 도 12에 따른 오디오 신호 디코더(1200)에 의해 발생할 수 있다.

[0153] 도 13의 신호 흐름(1300)은, 단순히 주파수 도메인 모드와 단순함을 위한 선형 예측 모듈의 TCX 서브-모드에서의 연산을 단지 설명하는 것을 알 수 있다. 그러나, 선형 예측 모듈의 ACELP 서브-모드에서 디코딩은 도 12를 참조하여 논의된 것과 같이 행해진다.

[0154] 보통의 주파수 도메인 모드/TCX 서브-모드 브랜치(1230)는 인코딩된 주파수 도메인 정보(1228)를 수신한다. 인코딩된 주파수 도메인 정보(1228)는, 주파수 도메인 모드에서 주파수 도메인 채널 스트림("fd_channel_stream")으로부터 추출되는, 이른바 산술적인 코딩된 스펙트럼 데이터 "ac_spectral_data"를 포함할 수 있다. 인코딩된 주파수 도메인 정보(1228)는 TCX 서브-모드에서 선형 예측 도메인 채널 스트림("lpd_channel_stream")으로부터 추출되는, 이른바 TCX 코딩("tcx_coding")을 포함할 수 있다. 엔트로피 디코딩(1330a)은 산술 디코더를 이용하여 수행될 수 있다. 예를 들어, 엔트로피 코딩(1330a)은 산술적 디코더를 이용하여 수행될 수 있다. 따라서, 양자화된 스펙트럼 계수들 "x_ac_quant"은 주파수-도메인 인코딩된 오디오 프레임들을 위해 얻어지고, 양자화된 TCX 모드 스펙트럼 계수들 "x_tcx_quant"은 TCX 모드에서 인코딩된 오디오 프레임들을 위하여 얻어진다. 양자화된 주파수 도메인 모드 스펙트럼 계수들과 양자화된 TCX 모드 스펙트럼 계수들은 몇몇의 실시예에서 정수일 수 있다. 예를 들어, 엔트로피 디코딩은, 문맥-인식 방법으로 인코딩된 스펙트럼 계수들의 그룹들을 공동으로 디코딩할 수 있다. 게다가, 어떤 스펙트럼 계수를 인코딩하도록 요구되는 비트들의 개수는 스펙트럼 계수들의 크기에 따라 달라짐으로써, 이는 더 많은 코드단어 비트들이 비교적 더 큰 크기를 가진 스펙트럼 계수를 인코딩하기 위하여 요구되도록 한다.

[0155] 다음으로, 양자화 주파수 도메인 모드 스펙트럼 계수들과 양자화 TCX 모드 스펙트럼 계수들의 역 양자화(1330c)는, 예를 들어 역 양자화기(1230c)를 이용하여 수행될 것이다. 역 양자화는 다음의 공식으로 설명되어질 수 있다.

[0156]
$$x_invquant = Sign(x_quant) \cdot |x_quant|^{\frac{4}{3}}$$

[0157] 따라서, 역 양자화 주파수 도메인 모드 스펙트럼 계수들("x_ac_invquant")은 주파수 도메인 모드에서 인코딩된 오디오 프레임에 위하여 얻어지고, 역 양자화 TCX 모드 스펙트럼 계수들("x_tcx_invquant")은 TCX 서브-모드에서 인코딩된 오디오 프레임을 위하여 얻어진다.

[0158] 7.1 주파수 도메인에서 인코딩된 오디오 프레임을 위한 프로세싱

[0159] 다음으로, 주파수 도메인 모드에서 프로세싱이 요약된다. 주파수 도메인 모드에서, 노이즈 필링(1340)은, 역 양자화 주파수 도메인 모드 스펙트럼 계수들(1330d)("x_ac_invquant")의 노이즈-필링된(noise-filled) 버전(1342)을 얻도록, 역 양자화 주파수 도메인 모드 스펙트럼 계수들에 선택적으로 적용된다. 다음으로, 역 양자화 주파수 도메인 모드 스펙트럼 계수들의 노이즈 필링된 버전(1342)의 스케일이 수행될 수 있고, 여기서, 스케일링은 1344로 표시된다. 스케일링에서, 스케일 팩터 파라미터들(또한, 간단하게 스케일 팩터들 또는 sf[g][sfb]로 표시되는)은 역 양자화 주파수 도메인 모드 스펙트럼 계수들(1342)("x_ac_invquant")을 스케일링하도록 적용된다. 예를 들어, 다른 스케일 팩터들은 다른 주파수 밴드들(주파수 범위 또는 스케일 팩터 밴드들)의 스펙트럼 계수들과 연관될 수 있다. 따라서, 역 양자화 스펙트럼 계수들(1342)은 스케일링된 스펙트럼 계수들(1346)을 얻도록 연관된 스케일 팩터들과 곱해질 수 있다. 스케일링(1344)은 바람직하게는 국제 표준 ISO/IEC 14496-3, 서브파트 4, 서브-절 4.6.2 및 4.6.3에서 설명된 것처럼 수행될 수 있다. 예를 들어, 스케일링(1344)은 결합기(1230e)를 이용하여 수행될 수 있다. 따라서, 스케일링된 (그리고 결과적으로, 스펙트럼 성형된) 버전(1346), 주파수 도메인 모드 스펙트럼 계수들의 "x_rescal"이 얻어지며, 이는 주파수 도메인 표현(1230f)과 동등할 수 있다. 다음으로, 중앙/측면 프로세싱(1348)의 결합과 임시적 노이즈 성형 프로세싱(1350)의 결합은 주파수 도메인 모드 스펙트럼 계수들의 스케일링된 버전(1346)에 기반하여 선택적으로 수행될 수 있고, 이는 스케일링된 주파수 도메인 모드 스펙트럼 계수들(1346)의 프리-프로세싱된 버전(1352)을 얻기 위함이다. 예를 들어, 선택적인 중앙/측면 프로세싱(1348)은 ISO/IEC 14496-3: 2005, 오디오-비주얼 객체들의 정보 테크놀로지-코딩 - 파트 3: 오디오, 서브파트 4, 서브-절 4.6.8.1에서 설명된 것과 같이 수행될 수 있다. 선택적 시간적 노이즈 성형은 ISO/IEC 14496-3: 2005, 오디오-비주얼 객체들의 정보 테크놀로지-코딩 - 파트 3: 오디오, 서브파트 4, 서브-절 4.6.9에서 설명된 것과 같이 수행될 수 있다.

[0160] 다음으로, 역 변경된 이산 코사인 변환(1354)은 주파수-도메인 모드 스펙트럼 계수들의 스케일링된 버전(1346) 또는 그것의 포스트-프로세싱된 버전(1352)에 적용될 수 있다. 결과적으로, 최근에 프로세싱된 오디오 프레임의 오디오 콘텐츠의 시간 도메인 표현(1356)이 얻어진다. 또한, 시간 도메인 표현(1356)은 $x_{i,n}$ 으로 표시될 수 있다. 추정을 간단화하는 것으로써, 오디오 프레임마다 하나의 시간 도메인 표현 $x_{i,n}$ 이 존재한다고 가정할 수 있다. 그러나, 몇몇의 경우에, 단일 오디오 프레임에 연관된 멀티 윈도우들(예를 들어, 이른바 "short windows")에서, 오디오 프레임마다 복수 개의 시간 도메인 표현 $x_{i,n}$ 이 존재할 수 있다.

[0161] 다음으로, 윈도우잉(1358)은, $z_{i,n}$ 으로 또한 표시되는 윈도우잉된 시간 도메인 표현(1360)을 얻도록, 시간 도메인 표현(1356)에 적용된다. 따라서, 단순화된 경우에서, 오디오 프레임마다 하나의 윈도우가 있는 경우, 윈도우잉된 시간 도메인 표현(1360)은 주파수 도메인 모드에서 인코딩된 오디오 프레임마다 얻어진다.

[0162] 7.2 TCX 모드에서 인코딩된 오디오 프레임을 위한 프로세스

[0163] 다음으로, 프로세싱은 TCX 모드에서 전체적으로 또는 부분적으로 인코딩된 프레임을 위하여 설명될 것이다. 이러한 논의와 관련하여, 오디오 프레임은, 예를 들어, 선형 예측 모드의 다른 서브-모드로 인코딩될 수 있는 4개의 서브-프레임으로 분리될 수 있는 것을 알 수 있다. 예를 들어, 오디오 프레임의 서브-프레임들은 선형 예측 모드의 TCX 서브-모드로 또는 선형 예측 모드의 ACELP 서브-모드로 선택적으로 인코딩될 수 있다. 따라서, 서브-프레임의 각각은, 오디오 품질과 비트율 사이에서 최적의 코딩 효율 또는 최적의 트레이드오프가 얻어지도록

록 인코딩될 수 있다. 예를 들어, "mod[]"로 명명된 배열을 이용하는 시그널링은, TCX 서브-모드와 ACELP 서브-모드에서 인코딩되는 상기 오디오 프레임의 서브-프레임들을 표시하는 선형 예측 모드에서 인코딩된 오디오 프레임을 위한 비트스트림에 포함될 수 있다. 그러나, 본 개념은, 만약에 전체 프레임이 TCX 모드에서 인코딩되었다고 가정한다면, 가장 쉽게 이해될 수 있음을 알 수 있다. 오디오 프레임이 두 개의 TCX 서브-프레임들을 포함하는 다른 경우가, 상기 개념의 선택적인 확장으로 간주될 수 있다.

[0164] TCX 모드에서 전체 프레임이 인코딩된다고 가정하면, 노이즈 필링(1370)은, "quant[]"로 표시되는 역 양자화 TCX 모드 스펙트럼 계수들(1330d)에 적용되는 것을 볼 수 있다. 따라서, "r[i]"로 표시되는 TCX 모드 스펙트럼 계수들(1372)의 노이즈 필링된 세트가 얻어진다. 추가적으로, 이른바 스펙트럼 디-성형(de-shaping)(1374)은, "r[i]"로 표시되는 TCX 모드 스펙트럼 계수들의 스펙트럼-디-성형된 세트(1376)를 얻도록, TCX 모드 스펙트럼 계수들(1372)의 노이즈 필링된 세트에 적용된다. 다음으로, 스펙트럼 성형(1378)이 적용되며, 여기서 스펙트럼 성형은, 선형-예측-코딩(LPC) 필터의 필터 응답을 설명하는 인코딩된 LPC 계수들로부터 산출되는 선형-예측-도메인 이득값에 따라 수행된다. 스펙트럼 성형(1378)은 예를 들어 결합기(1230a)를 이용하여 수행될 수 있다. 따라서, 또한 "rr[i]"로 표시되는 TCX 모드 스펙트럼 계수들의 재구성된 세트(1380)가 얻어진다. 다음으로, TCX 모드에서 인코딩된 프레임(또는, 그렇지 않으면 서브-프레임)의 시간 도메인 표현(1384)을 얻도록, TCX 모드 스펙트럼의 재구성된 세트(1380)에 기반하여, 역 MDCT(1382)가 수행된다. 다음으로, TCX 모드에서 인코딩된 프레임(또는 서브-프레임)의 리스케일링된 시간 도메인 표현(1388)을 얻기 위하여, 여기서, 리스케일링된 시간 도메인 표현은 또한 "x_w[i]"으로 표시되며, 리스케일링(1386)은 TCX 모드에서 인코딩된 프레임(또는 서브-프레임)의 시간 도메인 표현(1384)에 적용된다. 리스케일링(1386)은 TCX 모드에서 인코딩된 프레임의 모든 시간 도메인 값들 또는 TCX 모드에서 인코딩된 서브-프레임의 스케일링과 일반적으로 동일한 것을 알 수 있다. 따라서, 리스케일링(1386)은 일반적으로 주파수 왜곡을 가져오지 않으며, 그것은 선택적인 주파수가 아니기 때문이다.

[0165] 리스케일링(1386)의 후속으로, 윈도우잉(1390)은 TCX 모드에서 인코딩되는 프레임(또는 서브-프레임)의 리스케일링된 시간 도메인 표현(1388)에 적용된다. 따라서, 윈도우잉된 시간 도메인 샘플들(1392)(또한, "z_{i,n}"으로 표시되는)이 얻어지고, 이는 TCX 모드에서 인코딩된 프레임(또는 서브-프레임)의 오디오 콘텐츠를 표현한다.

[0166] 7.3 오버랩-및-가산 프로세싱

[0167] 프레임들의 시퀀스의 시간 도메인 표현들(1360, 1392)은 오버랩-및-가산(overlap-and-add) 프로세싱(1394)을 이용하여 결합된다. 오버랩-및-가산 프로세싱에서, 첫 번째 오디오 프레임의 우측(시간적으로 후에) 부분의 시간 도메인 샘플들은, 후속의 두 번째 오디오 프레임의 좌측(시간적으로 전에) 부분의 시간 도메인 샘플들과 오버랩되고 가산된다. 이러한 오버랩-및-가산 프로세싱(1394)은, 동일한 모드에서 인코딩되는 후속의 오디오 프레임들과 다른 모드에서 인코딩되는 후속의 오디오 프레임들을 모두를 위해 수행된다. 비록 역 MDCT(1954)와 오버랩-및-가산 프로세싱(1394)의 출력 사이, 그리고 역 MDCT(1382)와 오버랩-및-가산 프로세싱(1394)의 출력 사이에서 어떤 왜곡 프로세싱을 피하기 위한 오디오 디코더의 특징적인 구조 때문에, 후속의 오디오 프레임들이 다른 모드(예를 들어, 주파수 도메인 모드와 TCX 모드에서)에서 인코딩된다고 하더라도, 시간 도메인 에일리어싱 제거는 오버랩-및-가산 프로세싱(1394)에 의해 수행된다. 다시 말해서, 윈도우잉(1358, 1390)과 리스케일링(1386)(그리고, 선택적으로, 프리-엠포시스(pre-emphasis) 필터링과 디-엠포사이징(de-emphasizing) 동작의 스펙트럼 비-왜곡(non-distorting) 결합)을 제외하고, 역 MDCT 프로세싱(1354, 1382)과 오버랩-및-가산 프로세싱(1394) 사이에 추가적인 프로세싱이 없다.

[0168] 8. MDCT 기반 TCX 관한 세부 사항들

[0169] 8.1 MDCT 기반 TCX-틀 설명

[0170] 코어 모드가 선형 예측 모드(비트스트림 변수 "core_mode"가 일(one)과 같다는 사실에 의해 표시되는)일 때, 그리고 세 개의 TCX 모드들(예를 들어, 오버랩의 256 샘플들을 포함하는, 512 샘플의 TCX 부분을 제공하기 위한 첫 번째 TCX 모드로부터, 256 오버랩 샘플들을 포함하는, 768 시간 도메인 샘플들을 제공하기 위한 두 번째 TCX 모드, 256 오버랩 샘플들을 포함하는, 1280 TCX 샘플들을 제공하는 세 번째 TCX 모드)의 하나 또는 그 이상이 "선형 예측 도메인" 코딩으로써 선택된다. 즉, 만약 "mod[x]"의 네 개의 배열들 중에 하나가 0보다 크면(여기서, 네 개의 배열들 mod[0], mod[1], mod[2], mod[3]은 비트스트림 변수로부터 도출되고, 최근의 오디오

오 프레임의 네 개의 서브-프레임들을 위한 LPC 서브-모드를 나타내며, 즉, 서브-프레임이 선형 예측 모드의 ACELP 서브-모드에서 또는 선형 예측 모드의 TCX 서브-모드에서 인코딩되는지 여부, 그리고, 비교적 긴 TCX 인코딩, 중간 길이 TCX 인코딩 또는 짧은 길이 TCX 인코딩이 이용되는지 여부를 표시한다), TCX 틀에 기반한 MDCT가 이용된다. 다시 말해, 만약에 최근의 오디오 프레임의 서브-프레임들의 하나가 선형 예측 모드의 TCX 서브-모드에서 인코딩되면, TCX 틀이 이용된다. TCX에 기반한 MDCT는, 산술적 디코더(엔트로피 디코더(1230a) 또는 엔트로피 디코딩(1330a)을 실행시키곤 하는)로부터 양자화 스펙트럼 계수들은 수신한다. 양자화 계수들(또는 그것의 역 양자화 버전(1230b))은 컴퍼트(comfort) 노이즈(노이즈 필링 동작(1370)에 의해 수행될 수 있는)에 의해 먼저 완료된다. 주파수-도메인 노이즈 성형에 기반한 LPC는 결과, 스펙트럼 계수들(예를 들어, 결합기(1230e) 또는 스펙트럼 성형 동작(1378)을 이용하여)(또는 그것의 스펙트럼-디-성형된 버전)에 적용되고, 역 MDCT 변환(MDCT(1230g) 또는 역 MDCT 동작((1382)에 의해 수행되는)은 시간 도메인 합성 신호를 얻도록 수행된다.

- [0171] 8.2 MDCT-기반의 TCX-정의들
- [0172] 다음으로, 몇몇의 정의들이 주어질 것이다.
- [0173] "lg"는 산술적인 디코더(예를 들어, 선형 예측 모드에서 인코딩된 오디오 프레임을 위한)에 의한 양자화 스펙트럼 계수들 출력의 개수를 나타낸다.
- [0174] 비트스트림 변수 "noise_factor"는 노이즈 레벨 양자화 인덱스를 나타낸다.
- [0175] 변수 "noise level"은 재구성된 스펙트럼에서 주입된 노이즈의 레벨을 나타낸다.
- [0176] 변수"noise[]"는 발생된 노이즈의 벡터를 나타낸다.
- [0177] 비트스트림 변수 "global_gain"은 리스케일링 이득 양자화 인덱스를 나타낸다.
- [0178] 변수 "g"는 리스케일링 이득을 나타낸다.
- [0179] 변수 "rms"는 합성된 시간-도메인 신호"x[]"의 루트 평균 제곱을 나타낸다.
- [0180] 변수 "x[]"는 합성된 시간-도메인 신호를 나타낸다.

- [0181] 8.3 디코딩 프로세스
- [0182] MDCT-기반의 TCX는, 산술적인 디코더(1230a)로부터 mod[] 값(즉, 변수 mod[]의 값에 의해)에 의해 결정되는 양자화 스펙트럼 계수들, lg를 요구한다. 이 값(즉, 변수 mod[]의 값)은 또한, 역 MDCT(1230g)(또는 역 MDCT 프로세싱(1382)과 상응하는 윈도우잉(1390))에 적용되는 윈도우 길이와 형상을 정의한다. 윈도우는 세 개의 부분들로 구성되며, 이는 L 샘플들의 좌측 오버랩(또는 좌-측 전이 슬로프로 나타나는), M 샘플들 중 하나의 중앙부분, R 샘플들의 우측 오버랩 부분(또는 우-측 전이 슬로프로 나타나는)이다. 길이 2*lg의 MDCT 윈도우를 얻기 위하여, ZL 제로들이 좌측에 더해지고, ZR 제로들이 우측에 더해진다.
- [0183] "short_window"로부터 또는 "short_window"으로 전이의 경우에 있어서, 상응하는 오버랩 영역 L 또는 R 은, "short_window"의 가능한 더 짧은 윈도우 슬로프에 적용하기 위하여 128(샘플들)로 감소시키는 것이 필요하다. 결과적으로, 영역 M과 상응하는 제로 영역 ZL 또는 ZR은 각각 64 샘플들까지 확장될 필요가 있다.
- [0184] 다시 말해, 보통 256 샘플=L=R의 오버랩이 있다. 그것은 FD 모드에서 LPD 모드의 경우에서 128로 줄어든다.
- [0185] 도 15는, 좌 제로 영역 ZL의 시간 도메인 샘플들, 좌 오버랩 영역 L, 중앙 부분 M, 우 오버랩 영역 R과 우 제로 영역 ZR의 개수뿐만 아니라 mod[]로써 스펙트럼 계수들의 개수를 보여준다.

[0186] MDCT 윈도우는 다음의 수학적식으로 주어진다.

$$W(n) = \begin{cases} 0 & \text{for } 0 \leq n < ZL \\ W_{SIN_LEFT,L}(n-ZL) & \text{for } ZL \leq n < ZL+L \\ 1 & \text{for } ZL+L \leq n < ZL+L+M \\ W_{SIN_RIGHT,R}(n-ZL-L-M) & \text{for } ZL+L+M \leq n < ZL+L+M+R \\ 0 & \text{for } ZL+L+M+R \leq n < 2lg \end{cases}$$

[0187]

$W_{SIN_LEFT,L}$ 와 $W_{SIN_RIGHT,R}$ 의 정의들은 아래에 주어질 것이다.

[0188]

[0189] MDCT 윈도우 $W(n)$ 은 윈도우잉 단계(1390)에 적용되며, 이는 윈도우잉 역 MDCT(예를 들어, 역 MDCT(1230g)의) 부분으로써 간주될 수 있다.

[0190] 산술적인 디코더(1230a)(또는, 그렇지 않으면, 역 양자화(1230c)에 의해)에 의해 산출되고, 또한 "quant[]"로 표시되는, 양자화 스펙트럼 계수들은 콤팩트 노이즈에 의해 완료된다. 주입된 노이즈의 레벨은 다음과 같이 디코딩된 비트스트림 변수 "noise_factor"에 의해 결정된다.

$$\text{noise_level} = 0.0625 * (8 - \text{noise_factor})$$

[0191]

[0192] "noise[]"로 표시되는 노이즈 벡터는, "random_sign()"로 표시되고, 값 -1 또는 +1로 랜덤하게 전달하는 랜덤 함수를 이용하여 계산된다. 다음의 관계가 유지된다.

$$\text{noise}[i] = \text{random_sign()} * \text{noise_level}$$

[0193]

[0194] "quant[]"와 "noise[]" 벡터는, "r[]"로 표시되는 재구성된 스펙트럼 계수들 벡터를 형성하도록 결합되고, "quant[]"에서, 8 연이은 제로들의 연속은 "noise[]"의 요소들로 대체된다. 8 개의 비-제로(non-zero)들의 연속(run)은 다음의 공식에 따라 탐지된다.

$$\begin{cases} rl[i] = 1 \text{ for } i \in [0, lg/6[\\ rl[lg/6 + i] = \sum_{k=0}^{\min(7, lg-8 \lfloor i/8 \rfloor - 1)} |quant[lg/6 + 8 \lfloor i/8 \rfloor + k]|^2 \text{ for } i \in [0, 5 \cdot lg/6[\end{cases}$$

[0195]

[0196] 다음과 같이 재구성된 스펙트럼들 얻는다.

$$r[i] = \begin{cases} \text{noise}[i] & \text{if } rl[i] = 0 \\ \text{quant}[i] & \text{otherwise} \end{cases}$$

[0197]

[0198] 상술된 노이즈 필터링은 엔트로피 디코더(1230a)에 의해 수행된 엔트로피 디코딩과 결합기(1230e)에 의해 수행된 결합 사이에서 포스트-프로세싱으로써 수행될 수 있다.

[0199] 스펙트럼 디-성형(de-shaping)이, 다음의 단계에 따라 재구성된 스펙트럼(예를 들어, 재구성된 스펙트럼(1376), r[])에 적용된다.

[0200] 1. 스펙트럼의 첫 번째 쿼터(quarter)의 8-차원의 블록을 위한 인덱스 m에서 8-차원의 블록의 에너지 E_m 을 계산한다.

[0201] 2. 비율 $R_m = \sqrt{E_m/E_1}$ 을 계산한다. 여기서, 1은 모든 E_m 의 최대값을 가진 블록 인덱스이다.

[0202] 3. 만약 $R_m < 0.1$ 이면, $R_m = 0.1$ 로 설정

[0203] 4. 만약 $R_m < R_m - 1$ 이면, $R_m = R_m - 1$ 로 설정

[0204] 스펙트럼의 첫 번째 쿼터에 속하는 각각의 8-차원의 블록은 팩터 R_m 에 의해 곱해진다.

[0205] 스펙트럼 디-성형은, 엔트로피 디코더(1230a)와 결합기(1230e) 사이에 신호 경로에 배열된 포스트-프로세싱으로 썬 수행될 것이다. 예를 들어, 스펙트럼 디-성형은 스펙트럼 디-성형(1374)에 의해 수행된다.

[0206] 역 MDCT를 적용하기에 앞서, MDCT 블록(즉, 좌 그리고 우 폴딩(folding) 포인트)의 양 끝에 상응하는 두 개의 양자화 LPC 필터들이 되찾아지고, 그들의 가중된 버전이 계산되며, 상응하는 훼손된(64 포인트들, 변환길이가 무엇이든) 스펙트럼들이 계산된다.

[0207] 다시 말해, LPC 필터 계수들의 첫 번째 세트는 시간의 첫 번째 구간을 위해 얻어지고, LPC 필터 계수들의 두 번째 세트는 시간의 두 번째 구간을 위하여 결정된다. LPC 필터 계수들의 세트들은 바람직하게는, 비트스트림에 포함된 상기 LPC 필터 계수들의 인코딩된 표현으로부터 산출된다. 시간의 첫 번째 구간은 바람직하게는 최근 TCX-인코딩된 프레임(또는 서브-프레임)의 시작 또는 그 이전이며, 시간의 두 번째 구간은 바람직하게는 TCX 인코딩된 프레임 또는 서브-프레임의 종료 또는 그 이후이다. 따라서, LPC 필터 계수들의 효과적인 세트는, 첫 번째 세트의 LPC 필터 계수들과 두 번째 세트의 LPC 필터 계수들의 가중된 평균을 형성함에 의해 결정된다.

[0208] 가중된 LPC 스펙트럼은 LPC 필터 계수들에 오드(odd) 이산 푸리에 변환(ODFT)을 적용함에 의해 계산된다. 복소 변형은, 오드 이산 푸리에 변환(ODFT)을 계산하기 전에 LPC (필터) 계수들에 적용되며, 이는 ODFT 주파수 빈스(bins)가 MDCT 주파수 빈스와 동조하도록(aligned)(바람직하게는 완벽하게) 하기 위함이다. 예를 들어, 주어진

LPC 필터 $\hat{A}(z)$ 의 가중된 LPC 합성 스펙트럼이 다음과 같이 계산된다.

$$X_o[k] = \sum_{n=0}^{M-1} x_l[n] e^{-j \frac{2\pi k}{M} n}$$

with

$$x_l[n] = \begin{cases} \hat{w}[n] e^{-j \frac{\pi}{M} n} & \text{if } 0 \leq n < lpc_order + 1 \\ 0 & \text{if } lpc_order + 1 \leq n < M \end{cases}$$

[0209]

[0210] 여기서, $\hat{w}[n]$, $n=0 \dots lpc_order+1$, 는 가중된 LPC 필터의 계수들이며 다음에 의해 주어진다.

$$\hat{W}(z) = \hat{A}(z / \gamma_1) \quad \text{with } \gamma_1 = 0.92$$

[0211]

[0212] 다시 말해, 0과 $lpc_order-1$ 사이에 n 가지는, $\hat{w}[n]$ 값들에 의해 표현되는, LPC 필터의 시간 도메인 응답은, 스펙트럼 도메인으로 변환되며, 이는 스펙트럼 계수들 $X_o[k]$ 을 얻기 위함이다. LPC 필터의 시간 도메인 응답 $\hat{w}[n]$ 은, 선형 예측 코딩 필터를 표현하는 시간 도메인 계수들 a_1 내지 a_{16} 으로부터 산출될 수 있다.

[0213] 이득 $g[k]$ 는 다음의 공식에 따라 LPC 계수들(예를 들어, a_1 내지 a_{16})의 스펙트럼 표현 $X_o[k]$ 으로부터 계산될 수 있다.

$$g[k] = \sqrt{\frac{1}{X_o[k] X_o^*[k]}} \quad \forall k \in \{0, \dots, M-1\}$$

[0214]

[0215] 여기서, $M=64$ 는 계산된 이득들에 적용되는 밴드의 개수이다.

[0216] 다음으로, 재구성된 스펙트럼(1230f, 1380), $rr[i]$ 은 계산된 이득들 $g[k]$ (또한, 선형 예측 모드 이득값들로 표시되는)에 따라 얻어진다. 예를 들어, 이득값 $g[k]$ 는 스펙트럼 계수(1230d, 1376), $r[i]$ 와 연관될 수 있다. 그렇지 않으면, 복수 개의 이득값들은 스펙트럼 계수(1230d, 1376), $r[i]$ 와 연관될 수 있다. 가중한 계수 $a[i]$ 는 하나 또는 그 이상의 이득값들 $g[k]$ 로부터 산출할 수 있고, 또는 가중한 계수 $a[i]$ 는 몇몇의 실시예에서 이득값 $g[k]$ 와 심지어 동일할 수 있다. 결과적으로, 가중한 계수 $a[i]$ 는, 스펙트럼 생성된 스펙트럼 계수 $rr[i]$ 에 스펙트럼 계수 $r[i]$ 의 기여를 결정하도록, 연관된 스펙트럼 값 $r[i]$ 과 곱해질 수 있다.

[0217] 예를 들어, 다음의 방정식이 유지된다.

[0218]
$$rr[i] = g[k] \cdot r[i]$$

[0219] 그러나, 다른 관계들이 또한 이용될 수 있다.

[0220] 상기에서, LPC 스펙트럼들이 훼손된다는 사실을 고려하면, 변수 k 는 $i(lg/64)$ 와 동일하다. 재구성된 스펙트럼 $rr[]$ 은 역 MDCT(1230g, 1382)로 제공된다. 역 MDCT를 수행할 때, 아래에서 자세하게 묘사될 것으로, 재구성된 스펙트럼 값들 $rr[i]$ 은 시간-주파수 값들 $X_{i,k}$ 로써 또는 시간-도메인 값들 $spec[i][k]$ 로써 제공된다. 다음의 관계가 유지될 수 있다.

[0221] $X_{i,k} = rr[k]$; 또는

[0222] $spec[i][k] = rr[k]$

[0223] TCX 브랜치에서 스펙트럼 프로세싱의 상기 논의에서, 변수 i 는 주파수 인덱스라는 것이 지적된다. 반대로, MDCT 필터 बैं크와 블록 스위칭의 논의에서, 변수 i 는 윈도우 인덱스이다. 그 기술에 숙련된 사람은, 문맥으로부터, 변수 i 가 주파수 인덱스인지 윈도우 인덱스인지 쉽게 인식할 수 있다.

[0224] 또한, 만약에 오디오 프레임이 단지 하나의 윈도우를 포함하고 있다면, 윈도우 인덱스가 프레임 인덱스와 동등하다는 것을 알 수 있다. 만약에 프레임이 멀티 윈도우를 포함하고 있다면, 때때로 이러한 경우에, 프레임마다 멀티 윈도우 인덱스 값들이 존재할 수 있다.

[0225] 비-윈도우잉된 출력 신호 $x[]$ 는 이득 g 에 의해 리스케일링되고, 디코딩된 글로벌 이득 인덱스("global_gain")의 역 양자화에 의해 얻어진다.

[0226]
$$g = \frac{10^{global_gain / 28}}{2 \cdot rms}$$

[0227] 여기서, rms 는 다음으로 계산된다.

[0228]
$$rms = \sqrt{\frac{\sum_{k=lg/2}^{3*lg/2-1} rr^2[k]}{L + M + R}}$$

[0229] 리스케일링된 합성된 시간-도메인 신호는 다음과 같다.

[0230]
$$x_w[n] = x[n] \cdot g$$

[0231] 리스케일링 후에, 윈도우잉과 오버랩-가산이 적용된다. 윈도우잉은, 위에서 설명된 것처럼, 윈도우 $W(n)$ 을 이용하고 도 15에서 보여진 윈도우잉 파라미터들을 고려하여 수행될 수 있다. 따라서, 윈도우잉된 시간 도메인 신호 표현 $z_{i,n}$ 이 다음으로 얻어진다.

[0232]
$$z_{i,n} = x_w[n] \cdot W(n)$$

[0233] 다음으로, 만약에 TCX 인코딩된 프레임들(또는 오디오 서브프레임들)과 ACELP 인코딩된 프레임들(또는 오디오 서브프레임들)이 존재한다면, 개념이 손쉽게 설명될 것이다. 또한, TCX-인코딩된 프레임들 또는 서브프레임들로

전송되는 LPC 필터 계수들은, 몇몇의 실시예에서 ACELP 디코딩을 초기화하기 위하여 적용될 것이다.

[0234] 또한, TCX 합성의 길이는, TCX 프레임 길이(오버랩 없이): 1, 2, 또는 3 mod[]에 대하여 각각(256,512 또는 1024 샘플들)에 의해 주어지는 것을 알 수 있다.

[0235] 나중에, 다음의 표기법이 적용되는데, x[]는 역 변형된 이산 코사인 변환의 출력을 표시하고, z[]는 시간 도메인에서 디코딩된 윈도우잉된 신호를 표시하며, out[]는 합성된 시간 도메인 신호를 표시한다.

[0236] 역 변형된 이산 코사인 변환의 출력은 다음과 같이 리스케일링되고 윈도우잉된다.

$$[0237] \quad z[n] = x[n] \cdot w[n] \cdot g; \forall 0 \leq n < N$$

[0238] N은 MDCT 윈도우 사이즈 $N=2lg$ 에 상응한다.

[0239] 전에 코딩 모드가 FD 모드 또는 TCX에 기반한 MDCT 일 때, 통상 오버랩과 가산은, 최근에 디코딩된 윈도우잉된 신호 $z_{i,n}$ 과 전의 디코딩된 윈도우잉된 신호 $z_{i-1,n}$ 사이에 적용되며, 인덱스 i는 이미 디코딩된 MDCT 윈도우들의 개수에 카운팅한다. 마지막 시간 도메인 합성 out는 다음의 공식에 의해 얻어진다.

[0240] $z_{i-1,n}$ 이 FD 모드로부터 오는 경우에

$$[0241] \quad out[i_{out} + n] = \begin{cases} z_{i-1, \frac{N-l}{2}+n}; \forall 0 \leq n < \frac{N-l}{4} - \frac{L}{2} \\ z_{i, \frac{N-N-l}{4}+n} + z_{i-1, \frac{N-l}{2}+n}; \forall \frac{N-l}{4} - \frac{L}{2} \leq n < \frac{N-l}{4} + \frac{L}{2} \\ z_{i, \frac{N-N-l}{4}+n}; \forall \frac{N-l}{4} + \frac{L}{2} \leq n < \frac{N-l}{4} + \frac{N-R}{2} \end{cases}$$

[0242] N_l 은 FD 모드로부터 오는 윈도우 시퀀스의 사이즈이다. i_{out} 는 출력 버퍼 아웃을 인덱싱하고, 기록된 샘플들

의 개수 $\frac{N-l}{4} + \frac{N-R}{2}$ 에 의해 증가된다.

[0243] $z_{i,n}$ 이 MDCT에 기반한 TCX로부터 오는 경우에,

$$[0244] \quad out[i_{out} + n] = \begin{cases} z_{i, \frac{N-L}{4}+n} + z_{i-1, \frac{3*N_{i-1}-L}{4}+n}; \forall 0 \leq n < L \\ z_{i, \frac{N-L}{4}+n}; \forall L \leq n < \frac{N+L-R}{2} \end{cases}$$

[0245] N_{i-1} 은 전의 MDCT 윈도우의 사이즈이다. i_{out} 는 출력 버퍼 아웃을 색인하고, 기록된 샘플들의 개수 $(N+L-R)/2$ 만큼씩 증가된다.

[0246] 다음으로, 몇몇의 가능성이 ACELP 모드에서 인코딩된 프레임 또는 서브-프레임으로부터 MDCT-기반의 TCX 모드에서 인코딩된 프레임 또는 서브-프레임으로 전이에서 아티팩트(artifacts)를 줄이는 것으로 설명될 것이다. 그러나, 다른 접근 방법 또한 이용될 수 있다.

[0247] 다음으로, 첫 번째 접근 방법이 간단히 설명된다. ACELP로부터 올 때, 특별한 윈도우 케인(cane)이 R로부터 0으로 줄이는 수단에 의해 후속 TCX를 위해 사용될 수 있고, 두 개의 후속 프레임들 사이에서 오버랩핑 영역이 제거될 수 있다.

[0248] 다음으로, 두 번째 접근 방법이 간단하게 설명된다(USAC에서 그리고 전에 설명된 것처럼). ACELP로부터 올 때,

후속 TCX 윈도우가 128 샘플들까지 증가하는 M의 수단들에 의해 확장된다. 디코더에서, 윈도우의 오른쪽 부분, 즉 첫 번째 R 논-제로 디코딩된 샘플들은 디코딩된 ACELP 샘플들에 의해 간단하게 제거되거나 대체된다.

[0249] 재구성된 합성 $out[i_{out}+n]$ 은 프리-엠포시스 필터($1-0.68z^{-1}$)를 통하여 필터링된다. 발생한 프리-엠포시스된 합성은, 여기 신호를 얻도록, 분석 필터에 의해 필터링된다. 계산된 여기는 ACELP 적응적 코드북을 업데이트하고, 후속 프레임에서 TCX로부터 ACELP의 스위칭을 허락한다. 분석 필터 계수들은 서브프레임 기반으로 보간된다.

[0250] 9. 필터뱅크와 블록 스위칭에 관한 세부 사항들

[0251] 다음으로, 역 변경된 이산 코사인 변환과 블록 스위칭에 관한 세부 사항들, 즉, 후속의 프레임들 또는 서브프레임들 사이에서 수행되는 오버랩-및-가산은 더 상세하게 설명될 것이다. 다음에서 설명되는 역 변경된 이산 코사인 변환은, 주파수 도메인에서 인코딩된 오디오 프레임들과 TCX 모드에서 인코딩된 오디오 프레임 또는 서브프레임 모두에 적용될 수 있다. TCX 모드에서 사용을 위한 윈도우들($W(n)$)이 상기에서 설명되어진 반면에, 주파수-도메인-모드를 위해 사용되는 윈도우들은 다음에서 설명될 것이다: 특히 주파수-모드에서 인코딩된 프레임으로부터 TCX 모드에서 인코딩된 후속 프레임으로 전이에서, 또는 그 반대로, 적절한 윈도우들의 선택은 시간-도메인 에일리어싱 제거를 갖도록 허락하는데, 이는 낮거나 에일리어싱하지 않는 전이들이 비트율 오버헤드 없이 얻어질 수 있도록 하기 위함이다.

[0252] 9.1 필터뱅크와 블록 스위칭-설명

[0253] 신호(예를 들어, 시간-주파수 표현(1158, 1230f, 1352, 1380)의 시간/주파수 표현은, 필터 뱅크 모듈(예를 들어, 모듈(1160, 1230g, 1354-1358-1394, 1382-1386-1390-1394)안으로 그것을 제공함으로써 시간 도메인에 매핑된다. 이 모듈은 역 변형된 이산 코사인 변환(IMDCT), 그리고 윈도우 및 오버랩-추가 함수로 구성된다. 입력 신호의 특성들에 시간/주파수 해상도를 적용하기 위하여, 블록 스위칭 틀이 또한 적용된다. N은 윈도우 길이를 표현하는데, N은 비트스트림 변수 "window_sequence"의 함수이다. 각각의 채널을 위하여, N/2 시간-주파수값들 $X_{i,k}$ 는, IMDCT를 통하여 N 시간 도메인 값들 $x_{i,n}$ 으로 변환된다. 각각의 채널을 위하여, 윈도우 함수를 적용한 후에, 각각의 채널 $out_{i,n}$ 을 위한 출력 샘플들을 재구성하도록, $z_{i,n}$ 시퀀스의 첫 절반은 이전의 블록 윈도우잉된 시퀀스 $z_{(i-1),n}$ 의 두 번째 절반에 더해진다.

[0254] 9.2. 필터뱅크와 블록 스위칭-정의들

[0255] 다음으로, 비트스트림 변수들의 몇몇의 정의가 주어질 것이다.

[0256] 비트스트림 변수 "window_sequence"는 어떤 윈도우 시퀀스(즉, 블록 사이즈)가 사용되었는지 표시하는 두 개의 비트를 포함한다. 비트스트림 변수 "window_sequence"는 일반적으로 주파수-도메인에서 인코딩된 오디오 프레임들을 위해 사용된다.

[0257] 비트스트림 변수 "window_shape"는 어떤 윈도우 함수가 선택되었는지 표시하는 하나의 비트를 포함한다.

[0258] 도 16의 표는, 일곱 개의 변환 윈도우들에 기반한 11개의 윈도우 시퀀스(또한, **window_sequences**로써 표시되는)를 보여준다. (ONLY_LONG_SEQUENCE, LONG_START_SEQUENCE, EIGHT_SHORT_SEQUENCE, LONG_STOP_SEQUENCE, STOP_START_SEQUENCE).

[0259] 다음으로, LPD_SEQUENCE는 소위 선형 예측 도메인 코덱 내의 모든 허용되는 윈도우/코딩 모드 조합을 참조한다. 주파수 도메인 코딩된 프레임을 디코딩하는 맥락에서, 후속하는 프레임이, LPD_SEQUENCE로 표현되는, LP도메인 코딩 모드로 인코딩되는지 여부를 아는 것은 중요하다. 그러나, LPD_SEQUENCE 내의 정확한 구조는 LP 도메인 코딩된 프레임을 디코딩할 때 고려된다.

[0260] 즉, 선형-예측 모드로 인코딩된 오디오 프레임이 싱글 TCX-인코딩된 프레임, 복수의 TCX-인코딩된 서브프레임

또는 TCX-인코딩된 서브프레임들과 ACELP-인코딩된 서브프레임들의 조합을 포함할 수 있다.

[0261] 9.3. 필터뱅크 및 블록 스위칭-디코딩 프로세스

[0262] 9.3.1 필터뱅크 및 블록 스위칭-IMDCT

[0263] IMDCT의 분석적인 표현은 다음과 같다:

$$x_{i,n} = \frac{2}{N} \sum_{k=0}^{\frac{N}{2}-1} spec[i][k] \cos\left(\frac{2\pi}{N} \left(n+n_0\right) \left(k + \frac{1}{2}\right)\right) \text{ for } 0 \leq n < N$$

[0264]

[0265] 여기서:

[0266] n = 샘플 인덱스

[0267] i = 윈도우 인덱스

[0268] k = 스펙트럼 계수 인덱스

[0269] N = 윈도우_시퀀스 값에 기초한 윈도우 길이

[0270] n₀ = (N/2 + 1)/2

[0271] 역변환에 대한 합성 윈도우 길이 N은 구문요소 "window_sequence"와 그 알고리즘 컨텍스트의 함수이다. 이는 다음과 같이 정의된다:

[0272] 윈도우 길이 2048:

$$N = \begin{cases} 2048, & \text{if ONLY_LONG_SEQUENCE} \\ 2048, & \text{if LONG_START_SEQUENCE} \\ 256, & \text{if EIGHT_SHORT_SEQUENCE} \\ 2048, & \text{if LONG_STOP_SEQUENCE} \\ 2048, & \text{if STOP_START_SEQUENCE} \end{cases}$$

[0273]

[0274] 도 17a 또는 17b의 테이블의 주어진 테이블 셀 내의 틱 표시(☑)는 특정 행(row)에 나열된 윈도우 시퀀스가 특정 열(column)에 나열된 윈도우 시퀀스에 의해 후속될 수 있다는 것을 지시한다.

[0275] 최초 실시예의 의미 있는 블록 전이가 도 17a에 기재된다. 추가적인 실시예의 의미있는 블록 전이는 도 17d의 테이블에 기재된다. 도 17b에 따른 실시예의 추가적인 블록 전이가 별도로 아래 설명될 것이다.

[0276] 9.3.2 필터뱅크 및 블록 스위칭 - 윈도우잉 및 블록 스위칭

[0277] 비트스트림 변수들 (또는 요소들) "window_sequence" 및 "window_shape" 요소에 따라, 서로 다른 변환 윈도우들이 사용된다. 다음에 기술되는 윈도우 절반들의 조합은 모든 가능한 윈도우 시퀀스를 제안한다.

[0278] "window_shape" == 1 에 대하여, 윈도우 계수들이 다음과 같이 카이저-베셀 (Kaiser - Bessel) 도출된(KBD) 윈도우에 의해 주어진다.

$$W_{KBD_LEFT,N}(n) = \sqrt{\frac{\sum_{p=0}^n [W'(p,\alpha)]}{\sum_{p=0}^{N/2} [W'(p,\alpha)]}} \quad \text{for } 0 \leq n < \frac{N}{2}$$

$$W_{KBD_RIGHT,N}(n) = \sqrt{\frac{\sum_{p=0}^{N-n-1} [W'(p,\alpha)]}{\sum_{p=0}^{N/2} [W'(p,\alpha)]}} \quad \text{for } \frac{N}{2} \leq n < N$$

[0279]

[0280] 여기서:

[0281] W' , 카이저-베셀 커널 윈도우 함수, 또한 [5] 참조, 가 아래 정의된다.

$$W'(n,\alpha) = \frac{I_0\left[\pi\alpha\sqrt{1.0 - \left(\frac{n - N/4}{N/4}\right)^2}\right]}{I_0[\pi\alpha]} \quad \text{for } 0 \leq n \leq \frac{N}{2}$$

$$I_0[x] = \sum_{k=0}^{\infty} \left[\frac{\left(\frac{x}{2}\right)^k}{k!} \right]^2$$

[0282]

$$\alpha = \begin{cases} 4 & \text{for } N = 2048 \text{ (1920)} \\ 6 & \text{for } N = 256 \text{ (240)} \end{cases}$$

[0283] α = 커널 윈도우 알파 팩터,

[0284] 그렇지 않으면, "window_shape" == 0 에 대하여, 싱글 윈도우가 다음과 같이 채워진다.

$$W_{SIN_LEFT,N}(n) = \sin\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)\right) \quad \text{for } 0 \leq n < \frac{N}{2}$$

$$W_{SIN_RIGHT,N}(n) = \sin\left(\frac{\pi}{N}\left(n + \frac{1}{2}\right)\right) \quad \text{for } \frac{N}{2} \leq n < N$$

[0285]

[0286] 윈도우 길이 N은 KBD와 사인(sine) 윈도우에 대하여 2048(1920)이나 256(240)일 수 있다.

[0287] 가능한 윈도우 시퀀스를 획득하는 방법은 이 부속절의 파트 a)-e)에서 설명된다.

[0288] 모든 종류의 윈도우 시퀀스에 대하여, 최초의 변환 윈도우의 왼쪽 절반의 변수 "window_shape"이 선행 블록의 윈도우 모양에 의해 결정되며, 이는 변수 "window_shape_previous_block"으로 기술된다. 다음 식은 이를 표현한

다.

$$W_{LEFT,N}(n) = \begin{cases} W_{KBD_LEFT,N}(n), & \text{if } "window_shape_previous_block" == 1 \\ W_{SIN_LEFT,N}(n), & \text{if } "window_shape_previous_block" == 0 \end{cases}$$

[0289]

[0290]

여기서:

[0291]

"window_shape_previous_block"은 변수이며, 이는 선행 블록(i-1)의 비트스트림 변수 "window_shape"과 동일하다.

[0292]

디코딩되는 첫 번째 행의 데이터 블록 "raw_data_block()"에 대하여, 윈도우의 좌측 절반 및 우측 절반의 변수 "window_shape"이 동일하다.

[0293]

선행 블록이 LPD 모드를 사용하여 코딩되는 경우에, "window_shape_previous_block"은 0으로 설정된다.

[0294]

a) ONLY_LONG_SEQUENCE:

[0295]

"window_sequence" == ONLY_LONG_SEQUENCE 로 지시되는 윈도우 시퀀스는 윈도우 타입, 2048(1920)의 총 윈도우 길이 N_I 를 갖는 „LONG_WINDOW“ 와 동일하다.

[0296]

"window_shape" == 1 에 대하여, 변수값, „ONLY_LONG_SEQUENCE“ 에 대한 윈도우가 다음과 같이 주어진다:

$$W(n) = \begin{cases} W_{LEFT,N_I}(n), & \text{for } 0 \leq n < N_I/2 \\ W_{KBD_RIGHT,N_I}(n), & \text{for } N_I/2 \leq n < N_I \end{cases}$$

[0297]

[0298]

"window_shape" == 0 에 대하여, 변수값, „ONLY_LONG_SEQUENCE“ 에 대한 윈도우가 다음과 같이 기술될 수 있다:

$$W(n) = \begin{cases} W_{LEFT,N_I}(n), & \text{for } 0 \leq n < N_I/2 \\ W_{SIN_RIGHT,N_I}(n), & \text{for } N_I/2 \leq n < N_I \end{cases}$$

[0299]

[0300]

윈도우잉 이후에, 시간 도메인 값들($z_{i,n}$)은 다음과 같이 표현될 수 있다:

$$z_{i,n} = w(n) \cdot x_{i,n};$$

[0301]

[0302]

b) LONG_START_SEQUENCE :

[0303]

윈도우 타입 "LONG_START_SEQUENCE" 가 올바른 오버랩과 가산을 윈도우 타입 "ONLY_LONG_SEQUENCE"로부터 좌측에 낮은-오버랩(짧은 윈도우 슬로프) 윈도우 절반을 갖는 블록으로의 블록 전이에 대하여 획득하도록 사용될 수 있다 (EIGHT_SHORT_SEQUENCE, LONG_STOP_SEQUENCE, STOP_START_SEQUENCE, 또는 LPD_SEQUENCE).

[0304]

후속하는 윈도우 시퀀스가 윈도우 타입 "LPD_SEQUENCE"가 아닌 경우에:

[0305]

윈도우 길이 N_I 및 N_s 가 각각 2048(1920) 및 256(240)으로 설정된다.

[0306]

후속하는 윈도우 시퀀스가 윈도우 타입 "LPD_SEQUENCE"인 경우에:

[0307]

윈도우 길이 N_I 및 N_s 가 각각 2048(1920) 및 512(480)으로 설정된다.

[0308] "window_shape" == 1 에 대하여, 윈도우 타입 "LONG_START_SEQUENCE"에 대한 윈도우가 다음과 같이 주어진다:

$$W(n) = \begin{cases} W_{LEFT,N_l}(n), & \text{for } 0 \leq n < N_l/2 \\ 1.0, & \text{for } N_l/2 \leq n < \frac{3N_l - N_s}{4} \\ W_{KBD_RIGHT,N_s}(n + \frac{N_s}{2} - \frac{3N_l - N_s}{4}), & \text{for } \frac{3N_l - N_s}{4} \leq n < \frac{3N_l + N_s}{4} \\ 0.0, & \text{for } \frac{3N_l + N_s}{4} \leq n < N_l \end{cases}$$

[0309]

[0310] "window_shape" == 0 에 대하여, 윈도우 타입 "LONG_START_SEQUENCE"에 대한 윈도우가 다음과 같이 표현된다.

$$W(n) = \begin{cases} W_{LEFT,N_l}(n), & \text{for } 0 \leq n < N_l/2 \\ 1.0, & \text{for } N_l/2 \leq n < \frac{3N_l - N_s}{4} \\ W_{SIN_RIGHT,N_s}(n + \frac{N_s}{2} - \frac{3N_l - N_s}{4}), & \text{for } \frac{3N_l - N_s}{4} \leq n < \frac{3N_l + N_s}{4} \\ 0.0, & \text{for } \frac{3N_l + N_s}{4} \leq n < N_l \end{cases}$$

[0311]

[0312] 윈도우잉된 시간-도메인 값들이 a)에 설명된 식으로 계산될 수 있다.

[0313] c) EIGHT_SHORT

[0314] "window_sequence" == EIGHT_SHORT에 대한 윈도우 시퀀스는 8개의 오버래핑되고 가산된 SHORT_WINDOW들을 포함하며, 각각 256(240)의 길이 N_s 를 갖는다. window_sequence의 총 길이는 선두 및 후속 0 들과 더불어 2048(1920)이다. 8개의 짧은 블록 각각은 처음에 별도로 윈도우잉된다. 짧은 블록 수는 변수 $j=0, \dots, M-1$ ($M = N_l/N_s$)로 인덱싱된다.

[0315] 선행 블록의 window_shape은 8개 짧은 블록들($W_0(n)$) 중 첫 번째 블록에만 영향을 미친다. 만일 window_shape == 1이라면, 윈도우 함수들이 다음과 같이 주어진다:

$$W_0(n) = \begin{cases} W_{LEFT,N_s}(n), & \text{for } 0 \leq n < N_s/2 \\ W_{KBD_RIGHT,N_s}(n), & \text{for } N_s/2 \leq n < N_s \end{cases}$$

$$W_j(n) = \begin{cases} W_{KBD_LEFT,N_s}(n), & \text{for } 0 \leq n < N_s/2 \\ W_{KBD_RIGHT,N_s}(n), & \text{for } N_s/2 \leq n < N_s \end{cases}, 0 < j \leq M-1$$

[0316]

[0317] 한편, 만일 window_shape == 0이라면, 윈도우 함수들이 다음과 같이 주어진다:

$$W_0(n) = \begin{cases} W_{LEFT,N_s}(n), & \text{for } 0 \leq n < N_s/2 \\ W_{SIN_RIGHT,N_s}(n), & \text{for } N_s/2 \leq n < N_s \end{cases}$$

$$W_j(n) = \begin{cases} W_{SIN_LEFT,N_s}(n), & \text{for } 0 \leq n < N_s/2 \\ W_{SIN_RIGHT,N_s}(n), & \text{for } N_s/2 \leq n < N_s \end{cases}, 0 < j \leq M-1$$

[0318]

[0319] 윈도우잉된 시간 도메인 값들 $z_{i,n}$ 내에 결과하는 EIGHT_SHORT window_sequence 간의 오버랩과 가산이 다음과 같

이 기술된다:

$$z_{l,n} = \begin{cases} 0, & \text{for } 0 \leq n < \frac{N-l-N_s}{4} \\ x_{0,n-\frac{N-l-N_s}{4}} \cdot W_0(n-\frac{N-l-N_s}{4}), & \text{for } \frac{N-l-N_s}{4} \leq n < \frac{N-l+N_s}{4} \\ x_{j-1,n-\frac{N-l+(2j-3)N_s}{4}} \cdot W_{j-1}(n-\frac{N-l+(2j-3)N_s}{4}) + x_{j,n-\frac{N-l+(2j-1)N_s}{4}} \cdot W_j(n-\frac{N-l+(2j-1)N_s}{4}), & \text{for } 1 \leq j < M, \frac{N-l+(2j-1)N_s}{4} \leq n < \frac{N-l+(2j+1)N_s}{4} \\ x_{M-1,n-\frac{N-l+(2M-3)N_s}{4}} \cdot W_{M-1}(n-\frac{N-l+(2M-3)N_s}{4}), & \text{for } \frac{N-l+(2M-1)N_s}{4} \leq n < \frac{N-l+(2M+1)N_s}{4} \\ 0, & \text{for } \frac{N-l+(2M+1)N_s}{4} \leq n < N-l \end{cases}$$

[0320]

[0321] d) LONG_STOP_SEQUENCE

[0322] window_squence는 윈도우 시퀀스 "EIGHT_SHORT_SEQUENCE" 또는 윈도우 타입 "LPD_SEQUENCE"로부터 윈도우 타입 "ONLY_LONG_SEQUENCE"로 다시 스위칭할 때 필요하다.

[0323] 선행하는 윈도우 시퀀스가 LPD_SEQUENCE 가 아닌 경우에:

[0324] 윈도우 길이 N_l 및 N_s 가 2048(1920) 및 256(240)으로 각각 설정된다.

[0325] 선행하는 윈도우 시퀀스가 LPD_SEQUENCE 인 경우에:

[0326] 윈도우 길이 N_l 및 N_s 가 2048(1920) 및 512(480)로 각각 설정된다.

[0327] 만일 **window_shape == 1**이라면, 윈도우 타입 "LONG_STOP_SEQUENCE"에 대한 윈도우가 다음과 같이 주어진다:

$$W(n) = \begin{cases} 0.0, & \text{for } 0 \leq n < \frac{N-l-N_s}{4} \\ W_{LEFT,N_s}(n-\frac{N-l-N_s}{4}), & \text{for } \frac{N-l-N_s}{4} \leq n < \frac{N-l+N_s}{4} \\ 1.0, & \text{for } \frac{N-l+N_s}{4} \leq n < N_l/2 \\ W_{KBD_RIGHT,N_l}(n), & \text{for } N_l/2 \leq n < N-l \end{cases}$$

[0328]

[0329] 만일 **window_shape == 0**이라면, "LONG_START_SEQUENCE"에 대한 윈도우가 다음과 같이 결정된다:

$$W(n) = \begin{cases} 0.0, & \text{for } 0 \leq n < \frac{N-l-N_s}{4} \\ W_{LEFT,N_s}(n-\frac{N-l-N_s}{4}), & \text{for } \frac{N-l-N_s}{4} \leq n < \frac{N-l+N_s}{4} \\ 1.0, & \text{for } \frac{N-l+N_s}{4} \leq n < N_l/2 \\ W_{SIN_RIGHT,N_l}(n), & \text{for } N_l/2 \leq n < N-l \end{cases}$$

[0330]

[0331] 윈도우잉된 시간 도메인 값들은 a)에 설명된 수식으로 계산될 수 있다.

[0332] e) STOP_START_SEQUENCE:

[0333] 윈도우 타입 "STOP_START_SEQUENCE"가 올바른 오버랩과 가산을 윈도우 타입 "ONLY_LONG_SEQUENCE"로부터 우측

에 낮은-오버랩(짧은 윈도우 슬로프) 윈도우 절반을 갖는 블록으로부터 좌측에 낮은-오버랩(짧은 윈도우 슬로프) 윈도우 절반을 갖는 블록으로의 블록 전이에 대하여 획득하도록 사용될 수 있으며, 싱글 룭 전이가 현행 프레임에 대하여 요구된다면 사용될 수 있다.

[0334] 후속하는 윈도우 시퀀스가 LPD_SEQUENCE 가 아닌 경우에:

[0335] 윈도우 길이 N_l 및 N_{sr} 이 2048(1920) 및 256(240)으로 각각 설정된다.

[0336] 후속하는 윈도우 시퀀스가 LPD_SEQUENCE 인 경우에:

[0337] 윈도우 길이 N_l 및 N_{sr} 이 2048(1920) 및 512(480)로 각각 설정된다.

[0338] 선행하는 윈도우 시퀀스가 LPD_SEQUENCE 가 아닌 경우에:

[0339] 윈도우 길이 N_l 및 N_{sl} 이 2048(1920) 및 256(240)으로 각각 설정된다.

[0340] 선행하는 윈도우 시퀀스가 LPD_SEQUENCE 인 경우에:

[0341] 윈도우 길이 N_l 및 N_{sl} 이 2048(1920) 및 512(480)으로 각각 설정된다.

[0342] 만일 $window_shape == 1$ 이라면, 윈도우 타입 "STOP_START_SEQUENCE"에 대한 윈도우가 다음과 같이 주어진다:

$$W(n) = \begin{cases} 0.0, & \text{for } 0 \leq n < \frac{N_l - N_{sl}}{4} \\ W_{LEFT_N_{sl}}(n - \frac{N_l - N_{sl}}{4}), & \text{for } \frac{N_l - N_{sl}}{4} \leq n < \frac{N_l + N_{sl}}{4} \\ 1.0, & \text{for } \frac{N_l + N_{sl}}{4} \leq n < \frac{3N_l - N_{sr}}{4} \\ W_{KBD_RIGHT_N_{sr}}(n + \frac{N_{sr}}{2} - \frac{3N_l - N_{sr}}{4}), & \text{for } \frac{3N_l - N_{sr}}{4} \leq n < \frac{3N_l + N_{sr}}{4} \\ 0.0, & \text{for } \frac{3N_l + N_{sr}}{4} \leq n < N_l \end{cases}$$

[0343]

[0344] 만일 $window_shape == 0$ 이라면, 윈도우 타입 "STOP_START_SEQUENCE"에 대한 윈도우는 다음과 같이 보인다:

$$W(n) = \begin{cases} 0.0, & \text{for } 0 \leq n < \frac{N_l - N_{sl}}{4} \\ W_{LEFT_N_{sl}}(n - \frac{N_l - N_{sl}}{4}), & \text{for } \frac{N_l - N_{sl}}{4} \leq n < \frac{N_l + N_{sl}}{4} \\ 1.0, & \text{for } \frac{N_l + N_{sl}}{4} \leq n < \frac{3N_l - N_{sr}}{4} \\ W_{SIN_RIGHT_N_{sr}}(n + \frac{N_{sr}}{2} - \frac{3N_l - N_{sr}}{4}), & \text{for } \frac{3N_l - N_{sr}}{4} \leq n < \frac{3N_l + N_{sr}}{4} \\ 0.0, & \text{for } \frac{3N_l + N_{sr}}{4} \leq n < N_l \end{cases}$$

[0345]

[0346] 윈도우잉된 시간-도메인 값들은 a)에 설명되는 식으로 계산될 수 있다.

[0347] 9.3.3 필터뱅크 및 블록 스위칭 - 선행 윈도우 시퀀스로 오버래핑 및 가산

[0348] EIGHT_SHORT 윈도우 시퀀스 내의 오버래핑과 가산 이외에, 모든 윈도우 시퀀스 중의 (또는 모든 프레임이나 서브프레임 중의) 첫 번째(좌측) 부분이 선행 윈도우 시퀀스(또는 선행 프레임이나 서브프레임)의 두 번째(우측)

부분으로 오버래핑되고 가산되어 마지막 시간 도메인 값들 $out_{i,n}$ 을 산출한다. 이러한 연산에 대한 수학적 표현은 다음과 같이 기술될 수 있다.

[0349] ONLY_LONG_SEQUENCE, LONG_START_SEQUENCE, EIGHT_SHORT_SEQUENCE, LONG_STOP_SEQUENCE, STOP_START_SEQUENCE의 경우에:

$$out_{i,n} = z_{i,n} + z_{i-1, n+\frac{N}{2}}; \quad \text{for } 0 \leq n < \frac{N}{2}, \quad N = 2048 (1920)$$

[0350]

주파수-도메인 모드로 인코딩된 오디오 프레임들 간의 오버랩-및-가산에 대한 상기 수식이 서로 다른 모드로 인코딩된 오디오 프레임들의 시간-도메인 표현의 오버랩-및-가산에 대하여 또한 사용될 수 있다.

[0352] 대안으로, 오버랩-및-가산이 다음과 같이 정의될 수 있다:

[0353] ONLY_LONG_SEQUENCE, LONG_START_SEQUENCE, EIGHT_SHORT_SEQUENCE, LONG_STOP_SEQUENCE, STOP_START_SEQUENCE의 경우에:

$$out[i_{out} + n] = Z_{i,n} + Z_{i-1, n+\frac{N-l}{2}}; \quad \forall 0 \leq n < \frac{N-l}{2}$$

[0354]

[0355] $N-l$ 은 윈도우 시퀀스의 크기이다. i_{out} 은 출력 버퍼 out 를 인덱싱하고 쓰여진(written) 샘플의 수 $\frac{N-l}{2}$ 만큼 씩 증가된다.

[0356] LPD_SEQUENCE의 경우에:

[0357] 다음으로, 에일리어싱 아티팩트를 감소시키도록 사용될 첫 번째 접근방법이 기술될 것이다. ACELP로부터 나올 때, 특정 윈도우 케인(cane)이 다음 TCX에 대하여 R에서 0으로 감소시키는 방법으로 사용되어, 두 후속하는 프레임들간의 오버래핑 구간을 제거할 수 있다.

[0358] 다음으로, 에일리어싱 아티팩트를 감소시키도록 사용될 두 번째 접근방법이 기술될 것이다(USAC WD5 및 그 이전에 기술된 바와 같이). ACELP로부터 나올 때, 다음 TCX 윈도우는 M(중간 길이)을 128 샘플씩 증가시키고 또한 TCX 윈도우와 연관되는 MDCT 계수들의 수를 증가시키는 방법으로 확대된다. 디코더에서, 윈도우의 우측 부분, 즉 최초의 R 비-제로 디코딩된 샘플들이 단순히 버려지고 디코딩된 ACELP 샘플들로 대체된다. 즉, 추가적인 MDCT 계수들(예를 들면, 1024 대신 1152)을 제공함으로써, 에일리어싱 아티팩트들이 감소된다. 다르게 말하면, 별도의 MDCT 계수들을 제공함으로써(MDCT 계수들의 수가 오디오 프레임당 시간 도메인 샘플들의 수의 반보다 크도록), 시간-도메인 표현의 에일리어싱-없는 부분이 획득될 수 있고, 이는 예정된 에일리어싱 제거에 대한 필요성을 비-중요 스펙트럼 샘플링의 비용으로 제거한다.

[0359] 한편, 선행하는 디코딩된 윈도우인 신호 $z_{i-1,n}$ 은 TCX 기반의 MDCT로부터 올때, 종래의 오버랩 및 가산이 마지막 시간 신호 out 에 대하여 수행된다. 오버랩 및 가산은 FD 모드 윈도우 시퀀스가 LONG_START_SEQUENCE 또는 EIGHT_SHORT_SEQUENCE 일때, 다음 식으로 표현될 수 있다.

$$out[i_{out} + n] = \begin{cases} z_{i, \frac{N-l-N-s}{4}+n} + z_{i-1, \frac{3N-l-N-s}{4}+n}; & \forall 0 \leq n < \frac{N-s}{2} \\ z_{i, \frac{N-l-N-s}{4}+n}; & \forall \frac{N-s}{2} \leq n < \frac{N-l+N-s}{4} \end{cases}$$

[0360]

[0361] N_{i-1} 은 TCX 기반의 MDCT에 적용된 선행 윈도우의 크기 $2lg$ 에 대응한다. i_{out} 은 출력 버퍼 out 를 인덱싱하고 쓰

여진(written) 샘플의 $(N_l + N_{sl})/4$ 의 수 만큼씩 증가 된다. $N_{sl}/2$ 는 도 15의 테이블에 정의된 TCX 기반의 선행 MDCT의 값 L 과 동일하다.

[0362] STOP_START_SEQUENCE에 대하여, FD 모드와 TCX 기반의 MDCT간의 오버랩 및 가산은 다음 식으로 표현된다.

$$out[i_{out} + n] = \begin{cases} z_{i, \frac{N_l - N_{sl}}{4} + n} + z_{i-1, \frac{3 \cdot N_l - 2 \cdot N_{sl}}{4} + n} ; \forall 0 \leq n < \frac{N_{sl}}{2} \\ z_{i, \frac{N_l - N_{sl}}{4} + n} ; \forall \frac{N_{sl}}{2} \leq n < \frac{N_l + N_{sl}}{4} \end{cases}$$

[0363]

[0364] N_{sl} 은 TCX 기반의 MDCT에 적용된 선행 윈도우의 크기 $2lg$ 에 대응한다. i_{out} 은 출력 버퍼 out 를 인덱싱하고 쓰여진(written) 샘플의 수 $(N_l + N_{sl})/4$ 만큼씩 증가 된다. $N_{sl}/2$ 는 도 15의 테이블에 정의된 TCX 기반의 선행 MDCT의 값 L 과 동일하다.

[0365] 10. $\hat{w}[n]$ 의 계산과 관련된 세부사항.

[0366] 다음으로, 선형-예측-도메인 이득 값들 $g[k]$ 의 계산과 관련되어 자세한 내용이 이해를 돕기 위하여 기술될 것이다. 전형적으로, 인코딩된 오디오 콘텐츠(선형-예측 모드로 인코딩되는)를 표현하는 비트스트림은 인코딩된 LPC 필터 계수들을 포함한다. 인코딩된 LPC필터 계수들은, 예를 들면, 대응하는 코드 단어들로 기술될 수 있고 오디오 콘텐츠를 복구하는 선형 예측 필터를 기술할 수 있다. LPC-인코딩된 오디오 프레임당 LPC 필터 계수들의 선형 예측 필터 세트의 수는 다양하다는 것을 주목해야한다. 실제로, 선형-예측 모드로 인코딩된 오디오 프레임에 대하여 비트스트림 내에 인코딩된, LPC 필터 계수 세트의 실제 숫자는 오디오 프레임의 ACELP-TCX 모드 조합(이는 때로는 "수퍼프레임"으로 지시된다)에 따라 다르다. 이러한 ACELP-TCX 모드 조합은 비트스트림 변수에 의해 결정될 수 있다. 그러나, 당연히 오직 TCX 만 가용한 경우들이 있고, ACELP 모드는 또한 가용하지 않은 경우들이 있다.

[0367] 비트스트림은 전형적으로 ACELP TCX 모드 조합에서 요구하는 LPC 필터 계수들의 세트 각각에 대응하는 양자화 인덱스를 추출하도록 파싱된다.

[0368] 첫 번째 프로세싱 스텝(1810)에서, LPC 필터의 역양자화가 수행된다. LPC필터들(즉, LPC 필터 계수들의 세트, 예를 들면, a_1 부터 a_{16})이 선 스펙트럼 주파수(LSF) 표현(LPC 필터 계수들의 인코딩된 표현들인)을 사용하여 양자화된다는 것이 주목되어야 한다. 첫 번째 프로세싱 스텝(1810)에서, 역양자화된 선 스펙트럼 주파수들(LSF)은 인코딩된 인덱스로부터 도출된다.

[0369] 이러한 목적으로, 첫번째 스테이지 근사값이 계산될 수 있고 선택적 대수 벡터 양자화(AVQ) 정련(refinement)이 계산될 수 있다. 역-양자화 선 스펙트럼 주파수들은 첫번째 스테이지 근사값과 역-가중된 AVQ 기여분을 더함으로써 재건될 수 있다. AVQ 정련의 존재는 LPC 필터의 실제 양자화 모드에 의존적일 수 있다.

[0370] 역-양자화된 선 스펙트럼 주파수 벡터는, LPC 필터 계수들의 인코딩된 표현으로부터 도출될 수 있으며, 후에 선-스펙트럼 쌍 파라미터들로 변환되어, 보간되고, 다시 LPC 파라미터들로 변환된다. 프로세싱 스텝(1810)에서 수행되는, 역양자화 절차는 선-스펙트럼-주파수-도메인 내의 LPC 파라미터들의 세트를 초래한다. 선-스펙트럼-주파수들이, 이때, 프로세싱 스텝(1820)에서, 코사인 도메인으로 변환되는데, 이는 선-스펙트럼 쌍으로 기술된다. 따라서, 선-스펙트럼 쌍 q_i (또는 그 보간된 버전)가 선형-예측 필터 계수들 a_k 로 변환되는데, 이는 프레임 또는 서브프레임 내의 재건된 신호를 합성하는데 사용된다. 선형-예측-도메인으로의 변환은 다음과 같이 수행된다. 계수들 $f_{1(i)}$ 및 $f_{2(i)}$ 가, 예를 들면, 다음 재귀 릴레이션을 사용하여 도출될 수 있다.

```

for i=1 to 8
    f1(i) = -2q2i-1*f1(i-1)+2f1(i-2)
    for j=i-1 down to 1
        f1(j) = f1(j)-2q2i-1*f1(j-1)+f1(j-2)
    end
end
end

```

[0371]

[0372] 초기값은 $f_1(0) = 1$ 및 $f_1(-1) = 0$ 이다. 계수들 $f_{2(i)}$ 는 q_{2i-1} 를 q_{2i} 로 대체함으로써 유사하게 계산된다.

[0373] 일단 $f_{1(i)}$ 및 $f_{2(i)}$ 의 계수가 발견되면, 계수들 $f_1'(i)$ 및 $f_2'(i)$ 가 다음과 같이 계산된다.

$$f_1'(i) = f_1(i) + f_1(i-1), \quad i = 1, \dots, 8$$

$$f_2'(i) = f_2(i) - f_2(i-1), \quad i = 1, \dots, 8$$

[0374]

[0375] 마지막으로, LP 계수들 a_i 가 $f_1'(i)$ 와 $f_2'(i)$ 로부터 다음과 같이 계산된다.

$$a_i = \begin{cases} 0.5f_1'(i) + 0.5f_2'(i), & i = 1, \dots, 8 \\ 0.5f_1'(17-i) - 0.5f_2'(17-i), & i = 9, \dots, 16 \end{cases}$$

[0376]

[0377] 요약하면, LPC 계수들 a_i 의 선-스펙트럼 쌍 계수들 q_i 로부터의 도출은 상술한 것처럼 프로세싱 스텝들(1830, 1840, 1850)을 사용하여 수행된다.

[0378] 가중된 LPC 필터의 계수들 $\hat{w}[n]$, $n=0 \dots \text{lpc_order}-1$, 이 프로세싱 스텝(1860)에서 획득된다. 계수 a_i 로부터 계수 $\hat{w}[n]$ 를 도출할 때, 계수 a_i 는 필터 특징 $\hat{A}[z]$ 를 갖는 a 필터의 시간-도메인 계수이고, 계수 $\hat{w}[n]$ 은 주파수-도메인 응답 $\hat{W}[z]$ 를 갖는 필터의 시간-도메인 계수라는 것이 고려되어야 한다. 또한, 다음의 관계가 있음이 고려되어야 한다:

$$\hat{W}(z) = \hat{A}(z / \gamma_1) \quad \text{with } \gamma_1 = 0.92$$

[0379]

[0380] 상기의 관점에서, 계수들 $\hat{w}[n]$ 이 인코딩된 LPC 필터 계수들로부터 쉽게 도출될 수 있으며, 이는, 예를 들면, 비트스트림 내의 각각의 인덱스로 표현된다.

[0381] 프로세싱 스텝(1870)에서 수행되는, $x_i[n]$ 의 도출이 위에서 논의되었음을 알아야 한다. 유사하게, $X_0[k]$ 의 계산이 위에서 논의되었다. 유사하게, 프로세싱 스텝(1890)에서 수행되는, $x_i[n]$ 선형-예측-도메인 이득 값들 $g[k]$ 의 계산이 위에서 논의되었다.

[0382] 11. 스펙트럼-성형을 위한 다른 해법

[0383] 스펙트럼-성형에 대한 개념이 위에서 논의되었는데, 이는 선형-예측-도메인 내에 인코딩된 오디오 프레임들에 대하여 적용되고, 또한 이는 LPC 필터 계수 $\hat{w}_0[n]$ 의 스펙트럼 표현 $X_0[k]$ 으로의 변환에 기초하며, 이로부 터 선형-예측-도메인 이득 값들이 도출된다. 상술한 것처럼, LPC 필터 계수들 $\hat{w}[n]$ 이 주파수-도메인 표현 $X_0[k]$ 로, 64개의 동일-간격 주파수 빈들을 갖는 오드(odd) 이산 푸리에 변환을 사용하여, 변환된다. 그러나, 당연히 동일한 주파수 간격을 갖는, 주파수-도메인 값들 $x_0[k]$ 을 획득할 필요는 없다. 오히려, 때로는 주파수-도메인 값들 $x_0[k]$ 을 사용하는 것이 권장될 수 있는데, 이는 비-선형으로 주파수 간격을 갖는다. 예를 들면, 주파수-도메인 값들 $x_0[k]$ 는 대수로 주파수 간격을 갖거나 또는 바크(Bark) 스케일에 따라서 주파수 간격을 갖는다. 주파수 도메인 값들 $X_0[k]$ 및 선형-예측-도메인 이득 값들 $g[k]$ 의 비-선형 간격은 특히 청취감과 계산 복잡성간의 양호한 트레이드-오프에 기인할 수 있다. 그럼에도 불구하고, 선형-예측-도메인 이득 값들의 비-단일 주파수 간격의 개념을 구현할 필요는 없다.

[0384] 12. 강화된 전이 개념

[0385] 다음으로, 주파수 도메인 내에 인코딩된 오디오 프레임과 선형-예측-도메인 내에 인코딩된 오디오 프레임 간의 전이에 대한 개선된 개념이 기술된다. 이러한 개선된 개념은 소위 선형-예측 모드 시작 윈도우를 사용하는데, 이는 다음에 설명될 것이다.

[0386] 먼저, 도 17a 및 17b를 참조하며, 상대적으로 짧은 우측 전이 슬로프를 갖는 종래의 윈도우들이 선형-예측 모드로 인코딩되는 오디오 프레임에 대한 전이가 일어날때 주파수-도메인 모드로 인코딩되는 오디오 프레임의 시간-도메인 샘플들에 적용됨을 볼 수 있다. 도 17a에 도시된 것처럼, 윈도우 타입 "LONG_START_SEQUENCE", 윈도우 타입 "EIGHT_SHORT_SEQUENCE", 윈도우 타입 "STOP_START_SEQUENCE"는 일반적으로 선형-예측-도메인으로 인코딩되는 오디오 프레임 앞에 적용된다. 따라서, 일반적으로, 주파수-도메인 인코딩된 오디오 프레임으로부터의 직접적인 전이의 가능성은 없으며, 여기에 상대적으로 긴 우측 슬로프를 갖는 윈도우가, 선형-예측 모드로 인코딩되는 오디오 프레임에 적용된다. 이는 상대적으로 긴 우측 전이 슬로프가 적용되는 주파수-도메인 인코딩된 오디오 프레임의 긴 시간-도메인 에일리어싱 부분에 의해 야기되는 심각한 문제가 있다는 사실 때문이다. 도 17a를 참조하면, 일반적으로 윈도우 타입 "only_long_sequence"가 연관되는 오디오 프레임으로부터, 또는 윈도우 타입 "long_stop_sequence"가 연관되는 오디오 프레임으로부터, 선형-예측 모드로 인코딩되는 연속하는 오디오 프레임으로의 전이는 가능성이 없다.

[0387] 그러나, 본 발명에 따른 몇 실시예에서, 새로운 타입의 오디오 프레임, 즉 선형-예측 모드 시작 윈도우가 연관되는 오디오 프레임이 사용된다.

[0388] 새로운 타입의 오디오 프레임(또한 간략하게 선형-예측 모드 시작 프레임으로 지시되는)이 선형-예측-도메인 모드의 TCX 서브-모드로 인코딩된다. 선형-예측 모드 시작 프레임은 싱글 TCX 프레임(즉, TCX 서브프레임으로 세분되지 않는)을 포함한다. 이에 따라, 1024개의 MDCT 계수들이 비트스트림내에, 선형-예측 모드 시작 프레임에 대하여, 인코딩된 형태로 포함된다. 즉, 선형-예측 시작 프레임에 연관되는 MDCT 계수들의 수는 주파수-도메인 인코딩된 오디오 프레임으로, 윈도우 타입 "only_long_sequence"의 윈도우에 연관되는 오디오 프레임에 연관되는 MDCT 계수들의 수와 동일하다. 또한, 선형-예측 모드 시작 프레임에 연관되는 윈도우는 윈도우 타입 "LONG_START_SEQUENCE"일 수 있다. 따라서, 선형-예측 모드 시작 프레임은 윈도우 타입 "long_start_sequence"에 연관되는 주파수-도메인 인코딩된 프레임과 매우 유사할 수 있다. 그러나, 선형-예측 모드 시작 프레임은, 스펙트럼-성형이, 스케일 팩터 값들에 따라 수행되기보다는, 선형-예측 도메인 이득 값들에 따라 수행되는, 그런 주파수-도메인 인코딩된 프레임과는 다르다. 따라서, 인코딩된 선형-예측-코딩 필터 계수들은 선형-예측-모드 시작 프레임에 대하여 비트스트림에 포함된다.

[0389] 역 MDCT(1354, 1382)가 주파수-도메인 모드로 인코딩된 오디오 프레임과 선형-예측 모드로 인코딩된 오디오 프레임 모두에 대하여 동일한 도메인(상술한 것처럼) 내에 적용됨에 따라, 양호한 시간-에일리어싱-제거 특징을 갖는 시간-도메인-에일리어싱-제거 오버랩-및-가산 연산이 주파수-도메인 모드로 인코딩된 상대적으로 긴 우측 전이 슬로프(예를 들면, 1024 샘플들 중의)를 갖는 선형 오디오 프레임과 상대적으로 긴 좌측 전이 슬로프(예를 들면, 1024 샘플들 중의)를 갖는 선형-예측 모드 시작 프레임간에 수행될 수 있으며, 전이 슬로프는 시간-에일

리어싱 제거에 매칭된다. 따라서, 선형-예측 모드 시작 프레임은 선형-예측 모드로 인코딩되고(즉, 선형-예측-코딩 필터 계수들을 사용하여) 또한 다른 선형-예측 모드로 인코딩된 오디오 프레임보다도 훨씬 더 긴(예를 들면, 적어도 팩터 2 만큼, 또는 적어도 팩터 4 만큼, 또는 적어도 팩터 8 만큼) 좌측 전이 슬로프를 포함하여, 추가적인 전이 가능성을 창출한다. 이에 따라, 선형-예측 모드 시작 프레임은 윈도우 타입 "long_sequence"를 갖는 주파수-도메인 인코딩된 오디오 프레임을 대체할 수 있다. 선형-예측 모드 시작 프레임은, MDCT 필터 계수들이 선형-예측 모드 시작 프레임에 대하여 전송되는 이점을 포함하는데, 이는 선형-예측 모드로 인코딩된 연속 오디오 프레임에 대하여 가용하다. 따라서, 연속 선형-예측-모드-인코딩된 오디오-프레임의 디코딩을 위한 초기화 정보를 갖기 위하여 추가 LPC 필터 계수 정보를 비트스트림으로 포함할 필요는 없다.

[0390] 도 14는 이러한 개념을 보여준다. 도 14는 4개 오디오 프레임들(1410, 1412, 1414, 1416)의 그래픽 표현을 보여주는데, 이들은 2048 오디오 샘플의 길이를 포함하고, 또한 대략 50% 정도로 오버래핑된다. 첫번째 오디오 프레임(1410)은 "only_long_sequence" 윈도우(1420)를 사용하여 주파수-도메인 모드로 인코딩되고, 두번째 오디오 프레임(1412)은 "long_start_sequence" 윈도우와 동일한, 선형-예측 모드 시작 윈도우를 사용하는 선형-예측 모드로 인코딩되고, 세번째 오디오 프레임(1414)은, 예를 들면, $\text{mod}[x]=3$ 의 값에 대하여 위에 정의된 것처럼 $\hat{W}[n]$ 윈도우(1424)를 사용하여 선형-예측 모드 시작 윈도우를 사용하는 선형-예측 모드로 인코딩된다. 선형-예측 모드 시작 윈도우(1422)는 길이 1024 오디오 샘플들의 좌측 전이 슬로프와 길이 256 샘플들의 우측 전이 슬로프를 포함한다. 윈도우(1424)는 길이 256 샘플들의 좌측 전이 슬로프와 길이 256 오디오 샘플들의 우측 전이 슬로프를 포함한다. 네번째 오디오 프레임(1416)은 "long_stop_sequence" 윈도우(1426)를 사용하여 주파수-도메인 모드로 인코딩되고, 길이 256 샘플들의 좌측 전이 슬로프와 길이 1024 샘플들의 우측 전이 슬로프를 포함한다.

[0391] 도 14를 참조하면, 오디오 프레임에 대한 시간-도메인 샘플들이 역 변경된 이산 코사인 변환(1460, 1462, 1464, 1466)에 의해 제공된다. 주파수-도메인 모드로 인코딩된 오디오 프레임들(1410, 1416)에 대하여, 스펙트럼-성형이 스케일 팩터들 및 스케일 팩터 값들에 따라 수행된다. 선형-예측 모드로 인코딩된, 오디오 프레임들(1412, 1414)에 대하여, 스펙트럼-성형이 인코딩된 선형 예측 코딩 필터 계수들로부터 도출된 선형-예측 도메인 이득 값들에 따라 수행된다. 어느 경우건, 스펙트럼 값들은 디코딩(또한, 선택적으로, 역 양자화)에 의해 제공된다.

[0392] 13. 결론

[0393] 요약하면, 본 발명에 따른 실시예들은 스위칭된 오디오 코더에 대한 주파수-도메인 내에 적용되는 LPC-기반의 노이즈-성형을 사용한다.

[0394] 본 발명에 따른 실시예들은 스위칭된 오디오 코덱의 맥락에서 서로 다른 코더들 간의 전이를 용이하게 하는 주파수-도메인 내의 LPC-기반의 필터를 적용한다.

[0395] 몇몇 실시예들은, 따라서, 세 개의 코딩 모드들, 주파수-도메인 코딩, TCX(변환-코딩된-여기 선형-예측-도메인) 및 ACELP(대수-코드-여기된(excited) 선형예측) 간의 효율적인 전이를 디자인하는 문제를 해결한다. 그러나, 몇몇 다른 실시예들은, 상술한 모드들 중 단지 두 개, 예를 들면, 주파수-도메인 코딩과 TCX 모드를 갖는 것으로 충분하다.

[0396] 본 발명에 따른 실시예들은 다음의 다른 해결방법보다 우수하다:

[0397] - 주파수-도메인 코더와 선형-예측 도메인 코더 간의 비-결정적으로 샘플링된 전이들(예를 들면, 참고문헌[4] 참조):

[0398] - 비-결정적 샘플링, 오버래핑 사이즈와 오버헤드 정보간의 트레이드-오프를 생성하며, MDCT의 용량을 모두 사용하지 않는다(시간-도메인-에일리어싱 제거 TDAC)

[0399] - 주파수-도메인 코더로부터 LPD 코더로 갈 때 추가 LPC 계수 세트를 전송해야 한다.

[0400] - 시간-도메인-에일리어싱 제거(TDAC)를 서로 다른 도메인 내에 적용(예를 들면, 참고문헌[5] 참조), LPC 필터링이 폴딩(folding)과 DCT 사이의 MDCT 내부에서 수행된다:

[0401] - 시간-도메인 에일리어싱된 신호가 필터리에 적절하지 않으며; 그리고

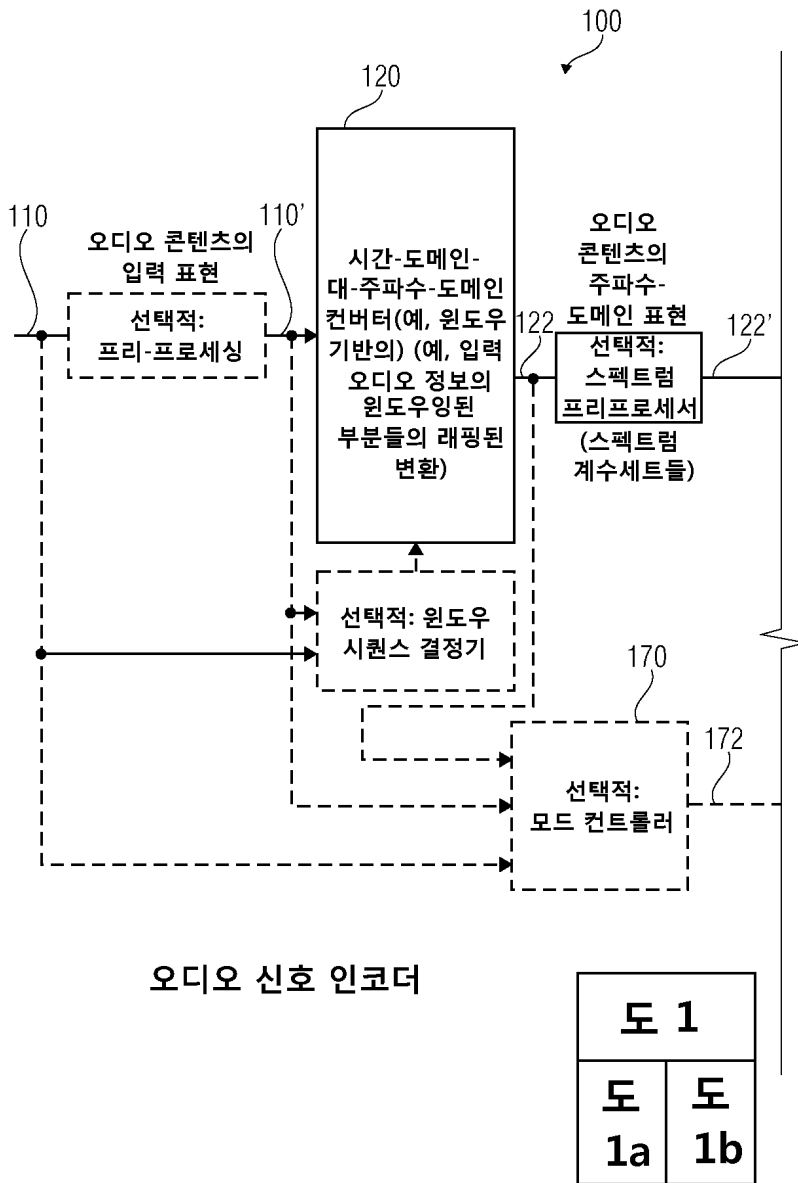
[0402] - 주파수-도메인 코더로부터 LPD 코더로 갈 때 추가 LPC 계수 세트를 전송해야 한다.

- [0403] - MDCT 도메인 내의 LPC 계수들을 비-스위칭된 코더 (T_{winVQ}) 에 대하여 계산한다(예를 들면, 참고문헌[6] 참조).
- [0404] - LPC 만을 스펙트럼을 평평하게 하는 스펙트럼 인벨로프(envelope) 표현으로 사용한다. 또 다른 오디오 코더로 스위칭할 때 전이를 용이하게 하기 위하여 LPC 나 양자화 노이즈를 성형을 이용하지 않는다.
- [0405] 본 발명에 따른 실시예들은 동일 도메인 내의 주파수-도메인 코더와 LPC 코더 MDCT를 수행하는 한편 LPC를 MDCT 도메인 내의 양자화 에러를 성형하는데 이용한다. 이는 수많은 이점을 갖는다.
- [0406] - LPC가 여전히 ACELP 같은 스피치-코더로 스위칭하도록 사용될 수 있다.
- [0407] - 시간-도메인 에일리어싱 제거(TDAC)는 TCX 로부터 주파수-도메인 코더로(또한 그 반대로)의 전이 동안 가능하며, 그때, 결정적 샘플링이 유지된다.
- [0408] - LPC는 여전히 ACELP의 주위에서 노이즈-성형으로서 사용되며, 이는 동일한 대상 함수를 사용하여 TCX와 ACELP 모두에 대하여(예를 들면, 폐쇄-루프 결정 프로세스 내의 LPC-기반의 가중된 세그먼트 SNR을) 최대화할 수 있도록 한다.
- [0409] 결론으로, 다음은 중요한 측면이다.
- [0410] 1. 변환-코딩된-여기(TCX)와 주파수 도메인(FD) 간의 전이는 선형-예측-코딩을 주파수 도메인 내에 적용함으로써 상당히 단순화되고/통합된다.
- [0411] 2. TCX 경우에 LPC 계수들의 전송을 유지함으로써, TCX와 ACELP 간의 전이가 다른 구현에서처럼 이점으로 실현될 수 있다(LPC 필터를 시간 도메인 내에 적용할 때).
- [0412] 구현 대안들
- [0413] 비록 어떤 실시예는 장치의 맥락에서 기술되었지만, 이와 같은 측면은 또한 대응하는 방법을 표현함이 명백하며, 이때, 블록이나 장치가 방법의 단계 또는 방법의 단계의 특징에 대응한다. 유사하게, 방법의 단계의 맥락에서 기술된 측면은 대응하는 블록이나 아이템 또는 대응하는 장치의 특징의 기술을 표현한다. 방법의 일부 또는 전체 단계는 하드웨어 장치, 예를 들면, 마이크로프로세서, 프로그램 가능한 컴퓨터, 또는 전자 회로에 의해 (또는 사용하여) 실행될 수 있다. 몇몇 실시예에서, 하나 이상의 가장 중요한 단계가 장치에 의해 실행될 수 있다.
- [0414] 본 발명의 인코딩된 오디오 신호는 디지털 저장 매체에 저장되거나 무선 전송 매체나 인터넷 같은 유선 전송 매체와 같은 전송 매체 상에 전송될 수 있다.
- [0415] 특정 구현 요건에 따라, 본 발명의 실시예는 하드웨어 또는 소프트웨어로 구현될 수 있다. 구현은 디지털 저장 매체, 예를 들면 플로피 디스크, DVD, 블루-레이, CD, ROM, PROM, EPROM, EEPROM, 또는 플래시 메모리와 같이, 내부에 전자적으로 판독 가능한 제어 신호를 갖고, 각 방법이 수행되는 프로그래머블 컴퓨터 시스템과 같이 협업하는(또는 협업할 수 있는), 저장매체를 사용하여 수행될 수 있다.
- [0416] 본 발명에 따른 몇몇 실시예는 전자적으로 판독 가능한 제어 신호를 갖는 데이터 캐리어를 포함하며, 이는 프로그래머블 컴퓨터 시스템과 협업하여, 여기서 기술된 방법이 수행되도록 할 수 있다.
- [0417] 일반적으로, 본 발명의 실시예들은 프로그램 코드를 갖는 컴퓨터 프로그램 제품으로 구현될 수 있고, 프로그램 코드는 컴퓨터 프로그램 제품이 컴퓨터에서 수행될 때 상기 방법들 중의 하나를 수행하도록 동작할 수 있다. 프로그램 코드는 예를 들면, 기계 판독가능한 캐리어 상에 저장된다.
- [0418] 다른 실시예들은 여기서 기술된 방법들 중의 하나를 수행하는 컴퓨터 프로그램을 포함하며, 이는 기계 판독 가능한 캐리어 상에 저장된다.
- [0419] 즉, 본 발명의 방법의 실시예는, 따라서, 컴퓨터 프로그램이 컴퓨터상에서 수행될 때, 여기서 기술된 방법들 중의 하나를 수행하는 프로그램 코드를 갖는 컴퓨터 프로그램이다.
- [0420] 본 발명의 방법의 다른 실시예들은, 따라서, 데이터 캐리어(또는 디지털 저장 매체, 또는 컴퓨터-판독가능한 매체)이며, 이는 여기서 기술된 방법들 중의 하나를 수행하는 컴퓨터 프로그램을 기록하여 포함한다.

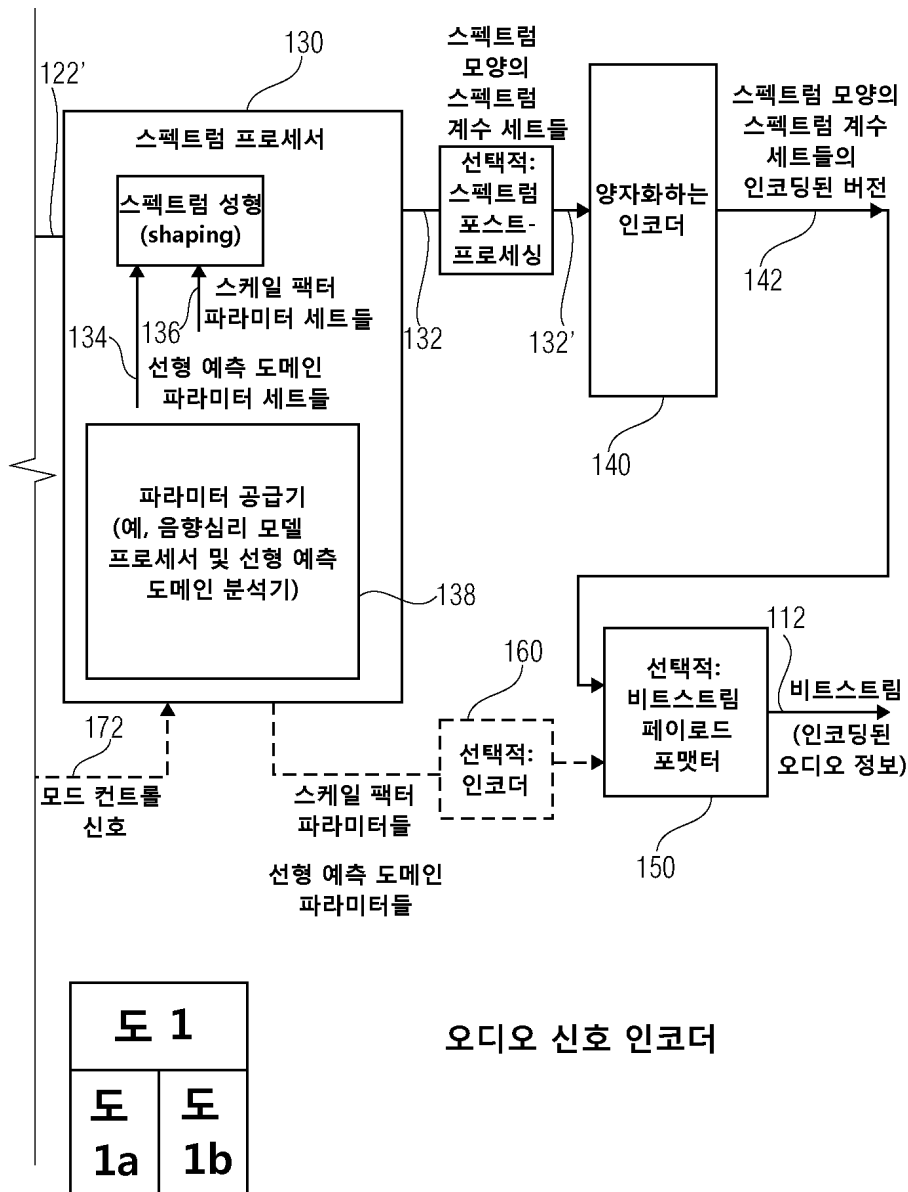
- [0421] 본 발명의 또 다른 실시예들은, 따라서, 데이터 스트림 또는 여기서 기술된 방법들 중의 하나를 수행하는 컴퓨터 프로그램을 표현하는 일련의 신호이다. 데이터 스트림과 일련의 신호는 예를 들면 데이터 통신 연결, 예를 들면 인터넷, 을 통해 전송되도록 구성될 수 있다.
- [0422] 또 다른 실시예는 처리 수단, 예를 들면 컴퓨터, 또는 여기서 기술된 방법들 중의 하나를 수행하도록 적용된, 프로그래머블 논리 장치를 포함한다.
- [0423] 또 다른 실시예는 여기에서 기술된 방법들 중의 하나를 수행하는 컴퓨터 프로그램이 설치된 컴퓨터를 포함한다.
- [0424] 본 발명에 따른 또 다른 실시예는 여기에서 기술된 방법들 중의 하나를 수행하는 컴퓨터 프로그램을 수신기로 전송(예를 들면, 전자적으로나 광적으로)하도록 구성된 장치나 시스템을 포함한다. 수신기는, 예를 들면, 컴퓨터, 모바일 장치, 메모리 장치 등일 수 있다. 장치나 시스템은 예를 들면, 컴퓨터 프로그램을 수신기로 전송하는 파일 서버를 포함할 수 있다.
- [0425] 어떤 실시예에서, 프로그래머블 논리 장치(예를 들면 필드 프로그래머블 게이트 어레이)는 여기서 기술된 방법의 기능성 일부 또는 전부를 수행하도록 사용될 수 있다. 어떤 실시예에서, 필드 프로그래머블 게이트 어레이는 마이크로프로세서와 협업하여 여기서 기술된 방법들 중의 하나를 수행할 수 있다. 일반적으로, 상기 방법은 어느 하드웨어 장치에 의해서도 바람직하게 수행될 수 있다.
- [0426] 상술한 실시예들은 단지 본 발명의 사상을 보여줄 뿐이다. 여기서 기술된 배열과 세부사항에 대한 수정 및 변경들이 당업자들에게 명백함이 이해되어야 한다. 하기의 특허 청구범위의 영역에 의해서만 제한되며 상술한 실시예의 기술이나 설명의 방법에 의해서 표현되는 특정 사항에 의하여 제한되지 않음이 의도된다.
- [0427] 참고문헌:
- [0428] [1] "Unified speech and audio coding scheme for high quqlity at low bitrates", Max Neuendorf et al., in iEEE Int, Conf. Acoustics, Speech and Signal Processing, ICASSP, 2009
- [0429] [2] Generic Coding of Moving pictures and Associated Audio: Advanced Audio Coding International Standard 13818-7, ISO/IEC JTC1/SC29/WG11 Moving Pictures Expert Group, 1977
- [0430] [3] "Extended Adaptive Multi-Rate - Wideband(AMR-WB+) codec", 3GPP TS 26.290 V6.3, 2005-06, Technical Specification
- [0431] [4] "Audio Encoder and Decoder for Encoding and Decoding Audio Samples", FH080703PUS, F49510, incorporated by reference,
- [0432] [5] "Apparatus and Method for Encoding/Decoding an Audio Signal Usign and Alasing Switch Scheme", FH080715PUS, F49522, incorporated by reference
- [0433] [6] "High-quality audio-coding at less than 64 kbits/s "by using transform-domain weighted interleave vector quantization(Twin VQ)", N.Iwakami and T. Moriya and S. Miki, IEEEICASSP, 1995

도면

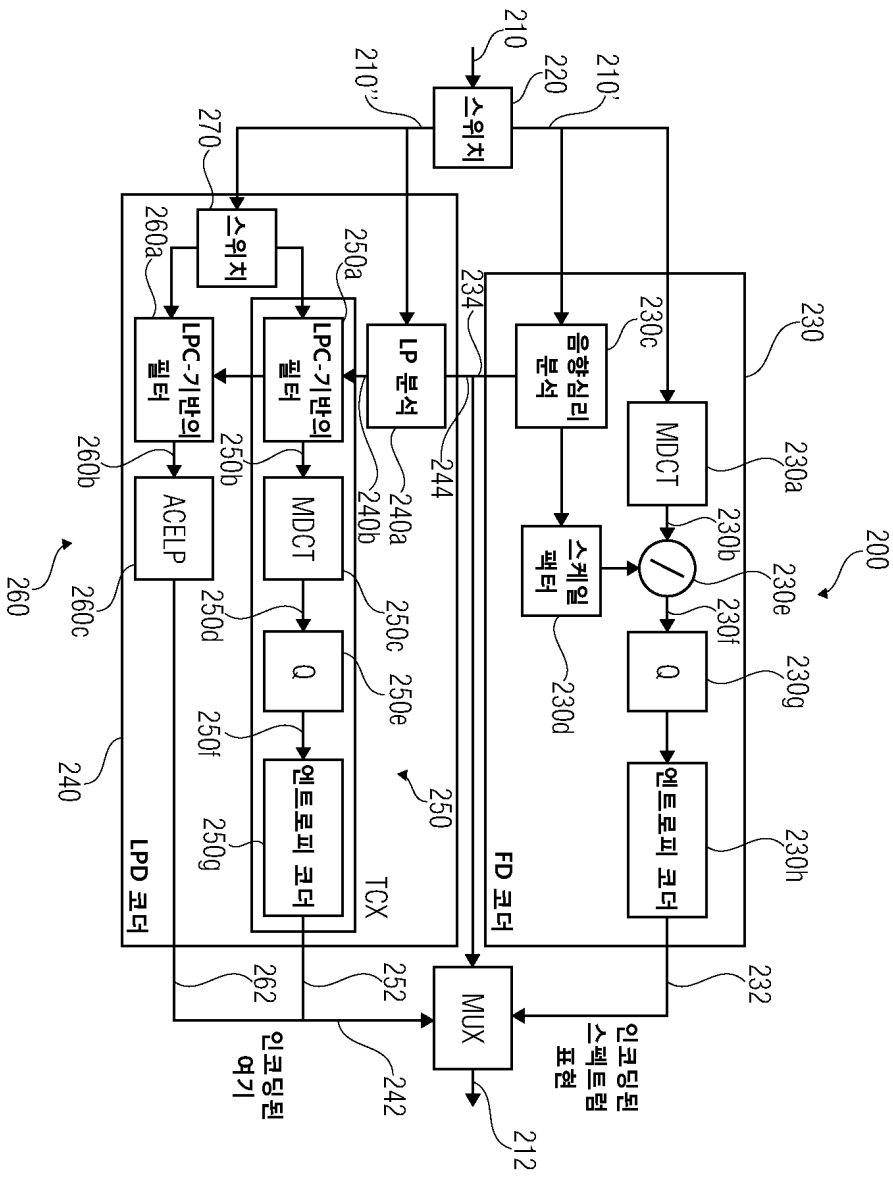
도면1a



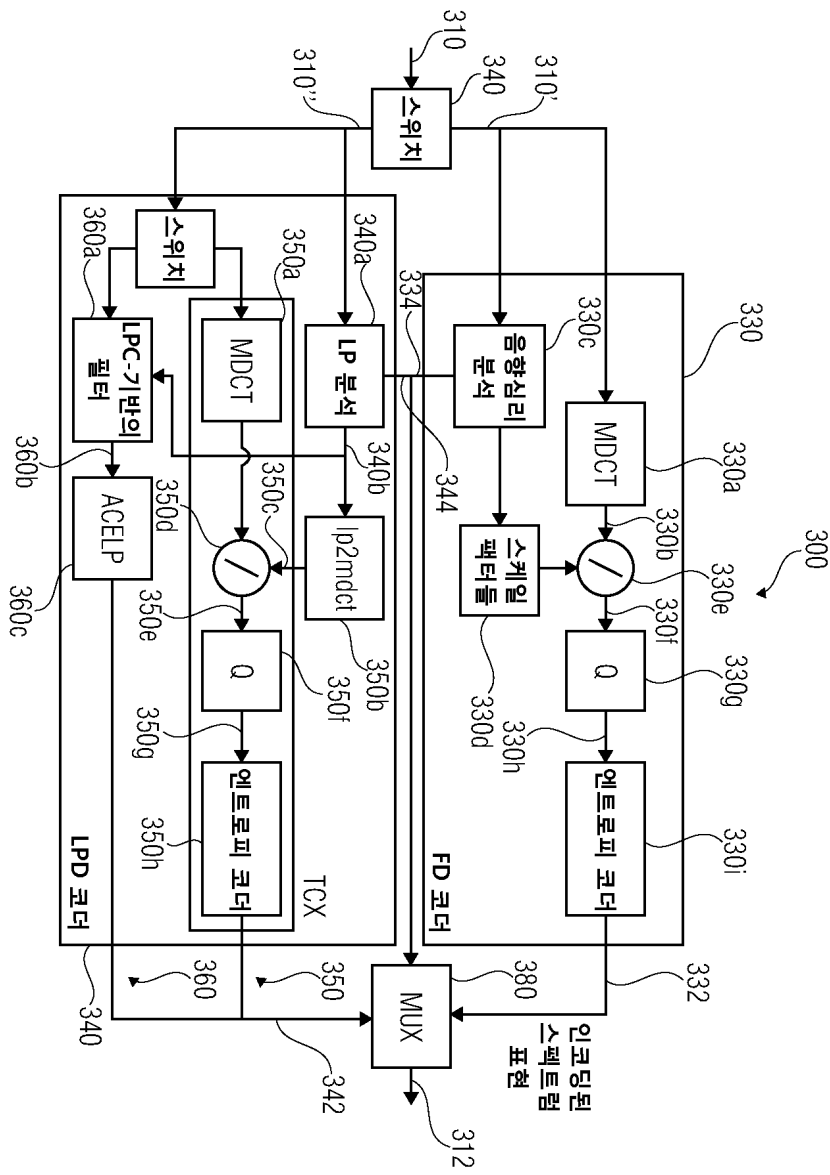
도면1b



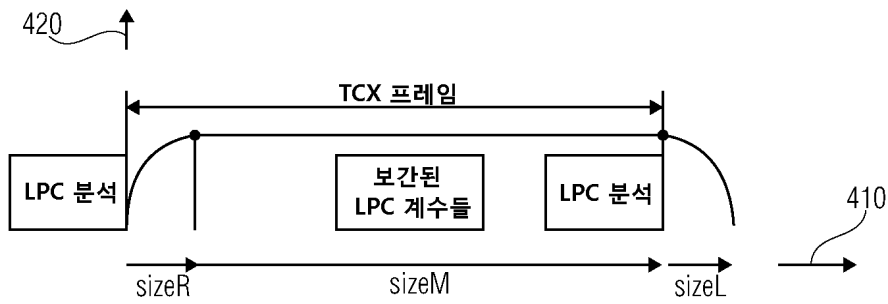
도면2



도면3



도면4



```

Function lpc2mdct(input lpc_coefs, lpc_order,
                 input sizeR, sizeM, sizeL,
                 output mdct_scaleFactors){
    int sizeN = 2 * (sizeL/2 + sizeM + sizeR/2);
    float InRealData[sizeN];
    float OutRealData[sizeN];
    float InImagData[sizeN];
    float OutImagData[mdct_size];

    /*ODFT*/
    for(i=0; i < lpc_order; i++){
        InRealData[i] = lpcCoefs[i] * cos(i*PI/(float)(sizeN));
        InImagData[i] = -lpcCoefs[i] * sin(i*PI/(float)(sizeN));
    }
    for(i <= sizeN; i++){
        InRealData[i] = 0.f;
        InImagData[i] = 0.f;
    }
}
Complex_fft(InRealData, InImagData, OutRealData, OutImagData, sizeN);
/*Get amplitude*/
for(i=0; i <= sizeN/2; i++){
    mdct_scaleFactors[i] = sqrt(1/(OutRealData[i]*OutRealData[i] + OutImagData[i]*OutImagData[i]));
}
}

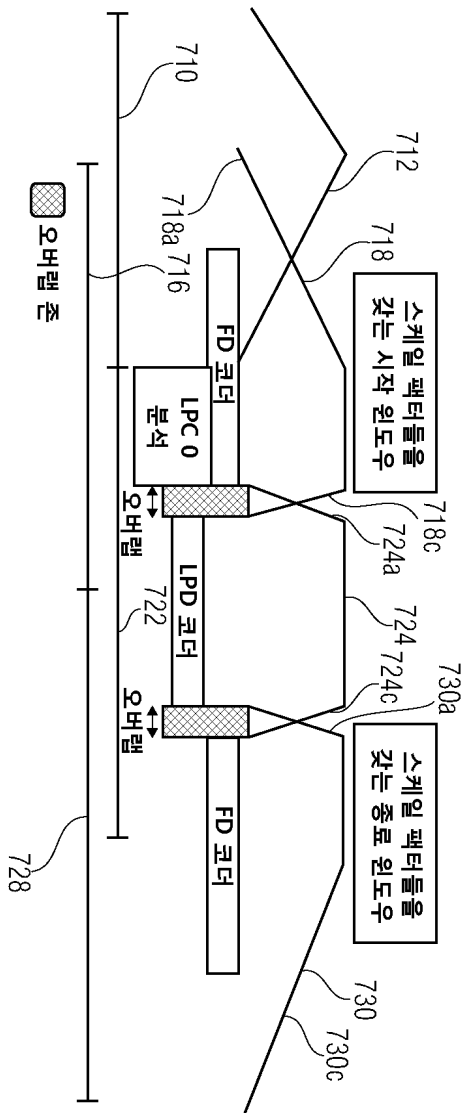
```

도면5

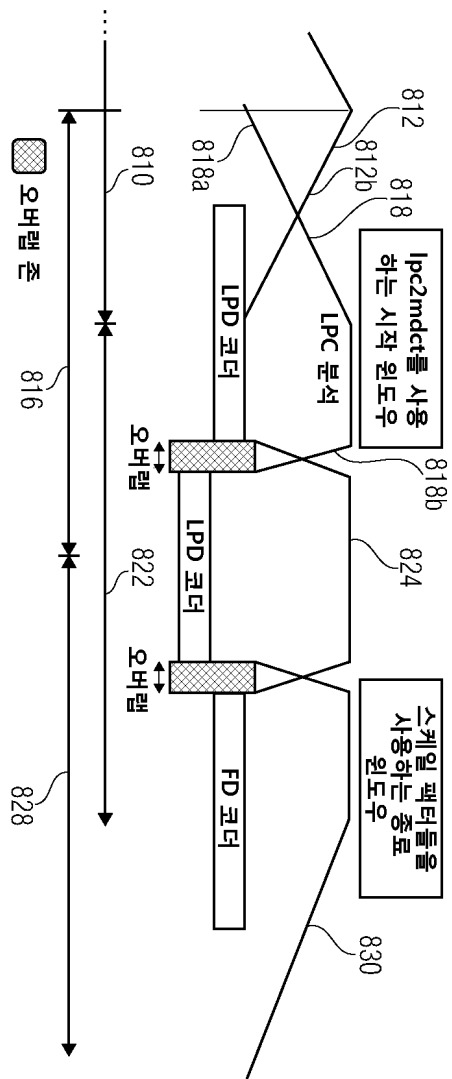
도면6

```
For(i=0;i<sizeL/2+sizeM+sizeR/2;i++){  
    mdcf[i]=mdcf[i]/mdct_scaleFactors[i];  
}
```

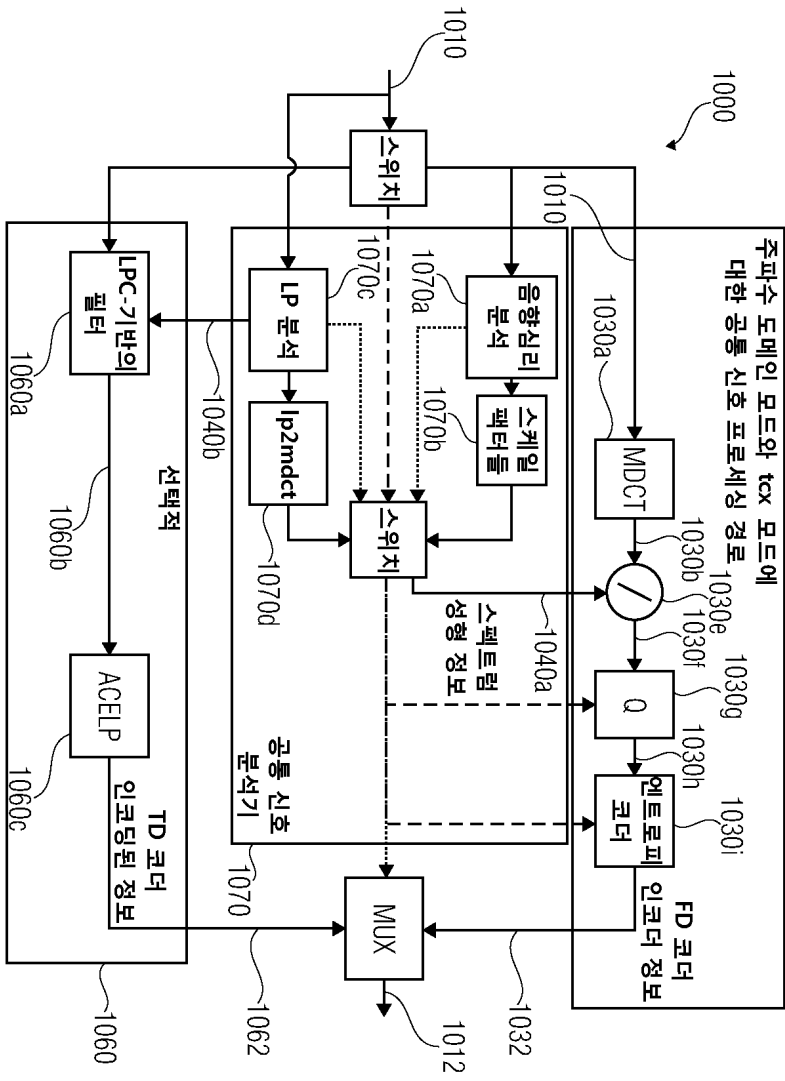

도면7



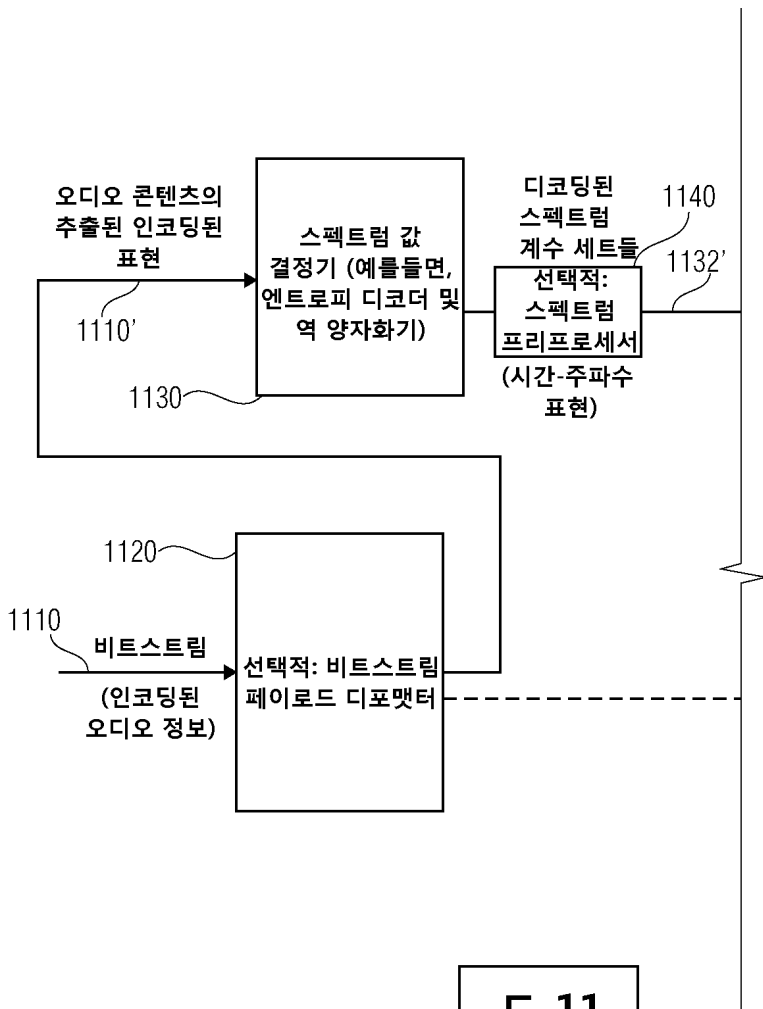
도면8



도면10

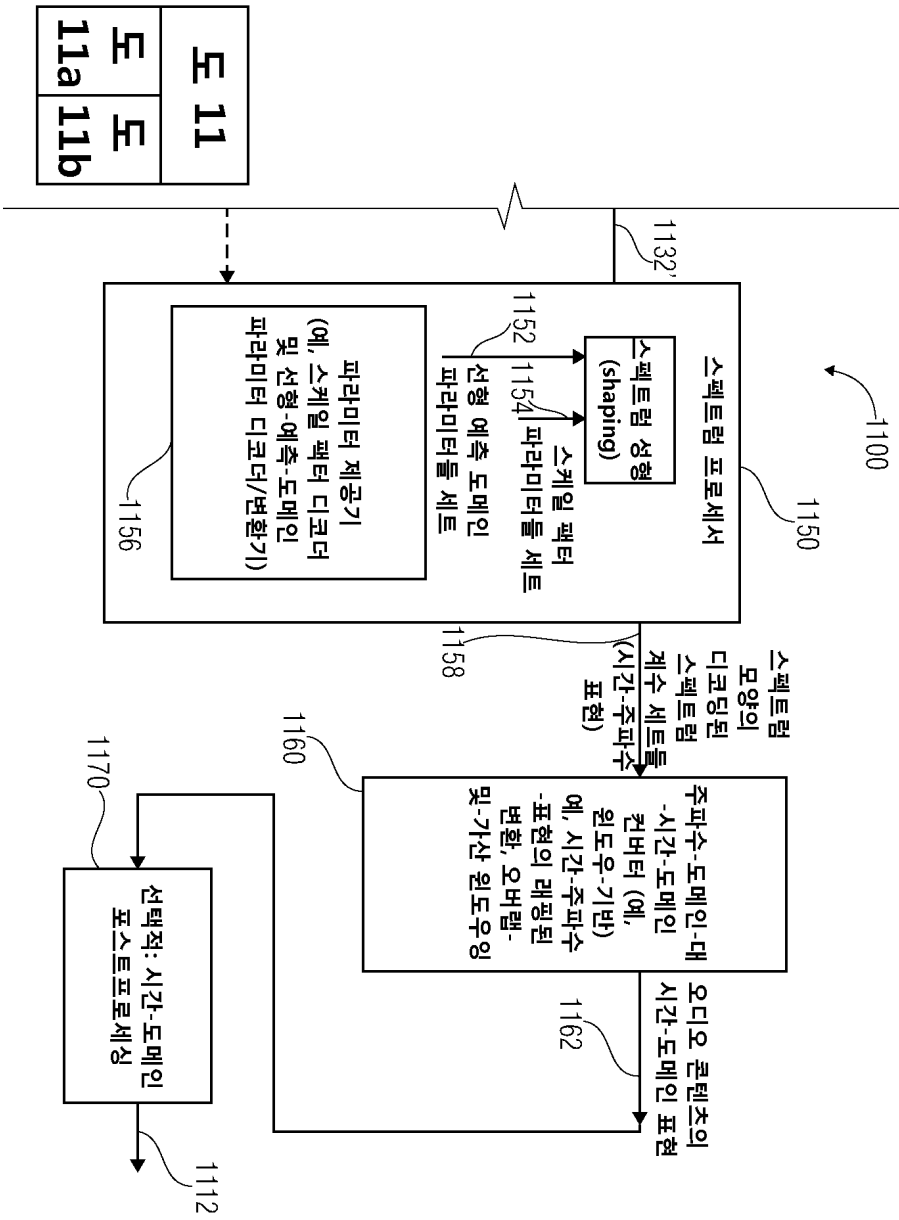


도면11a

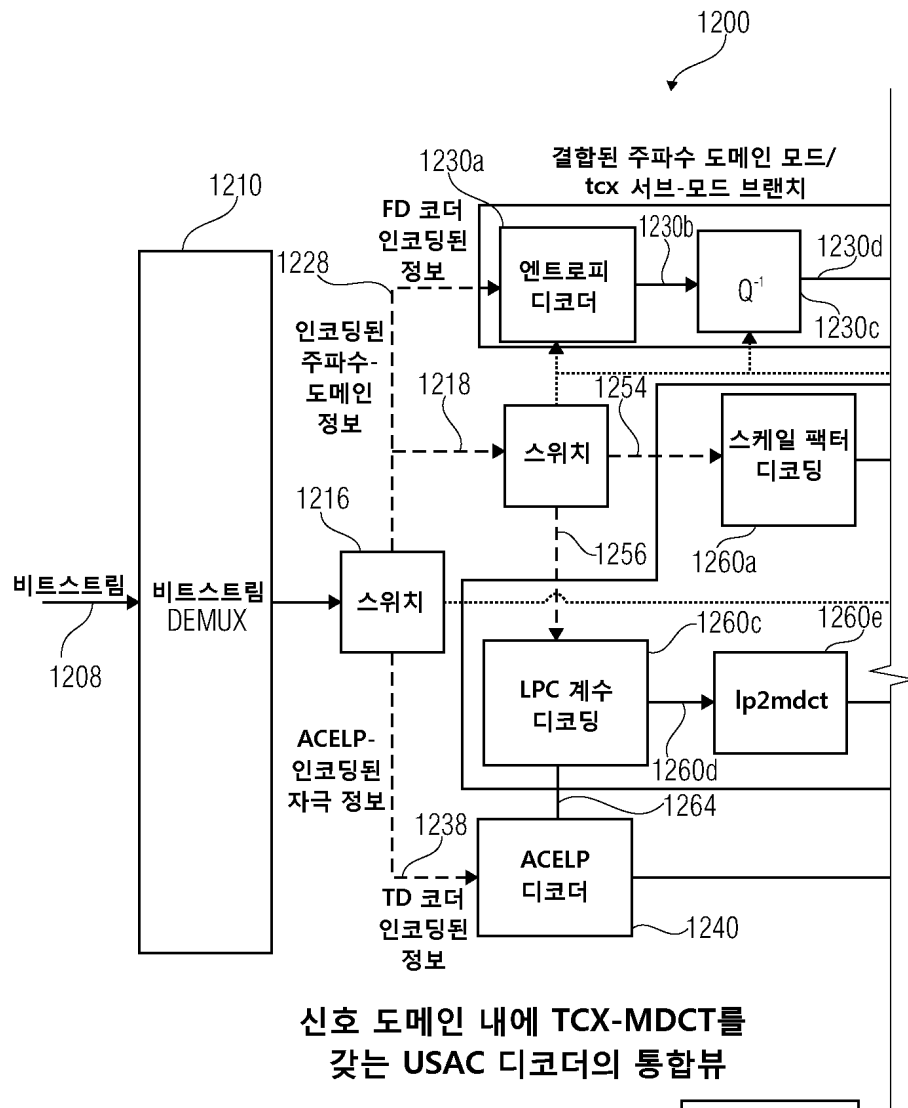


| | |
|----------|----------|
| 도 11 | |
| 도 11a | 도 11b |

도면11b



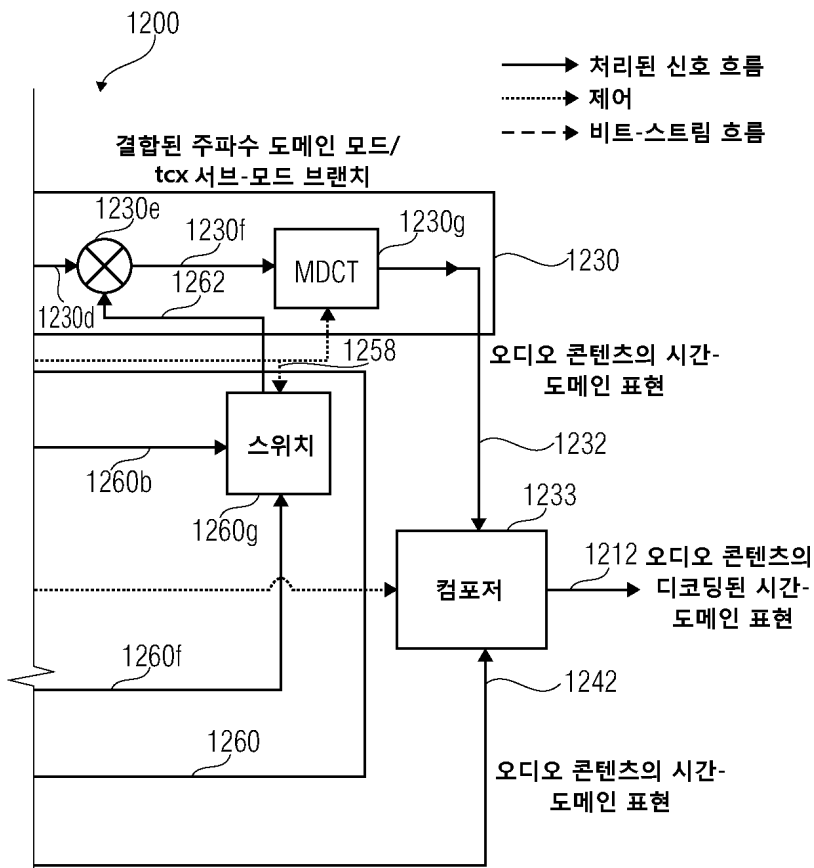
도면12a



도 12

| | |
|----------|----------|
| 도 12a | 도 12b |
|----------|----------|

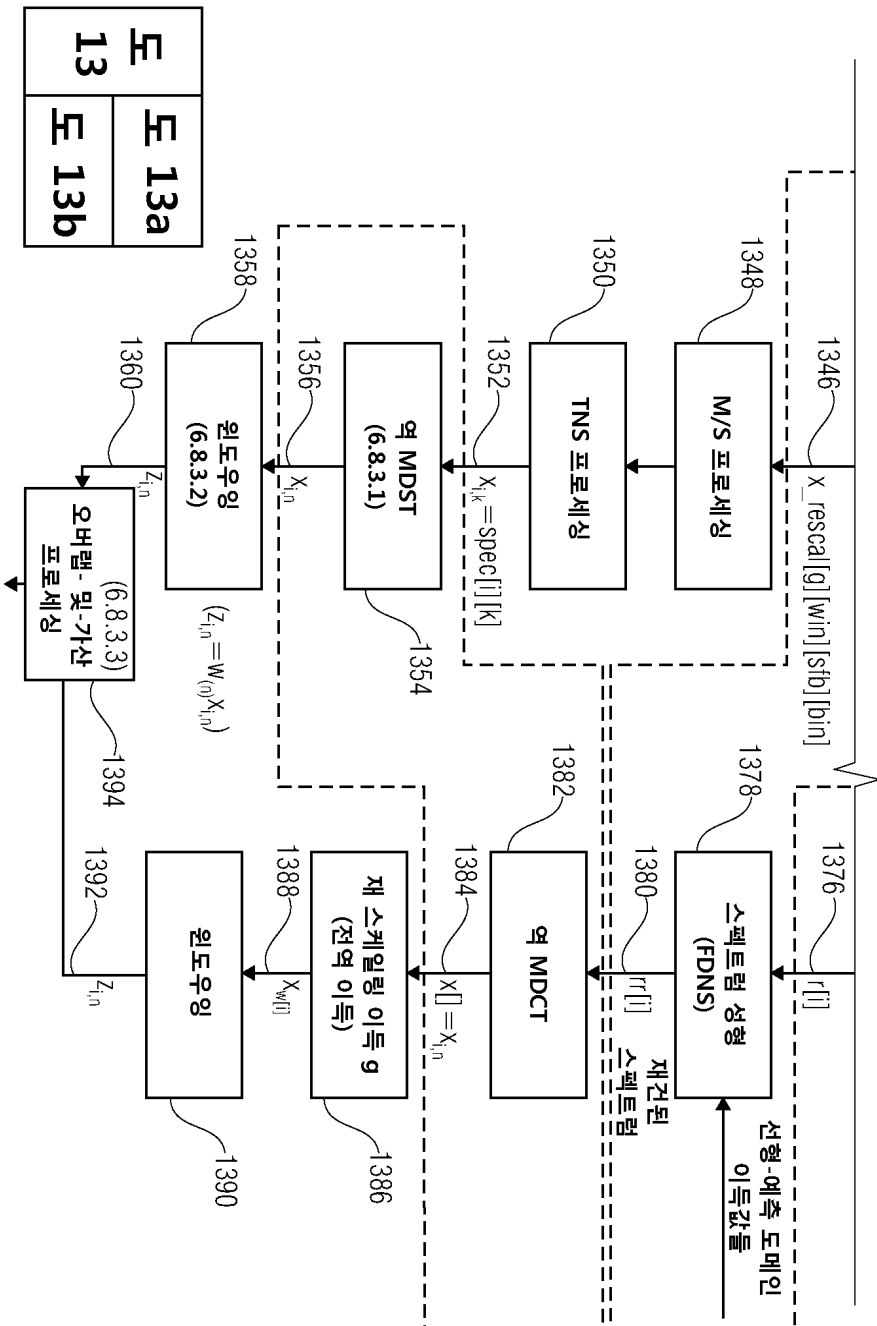
도면12b



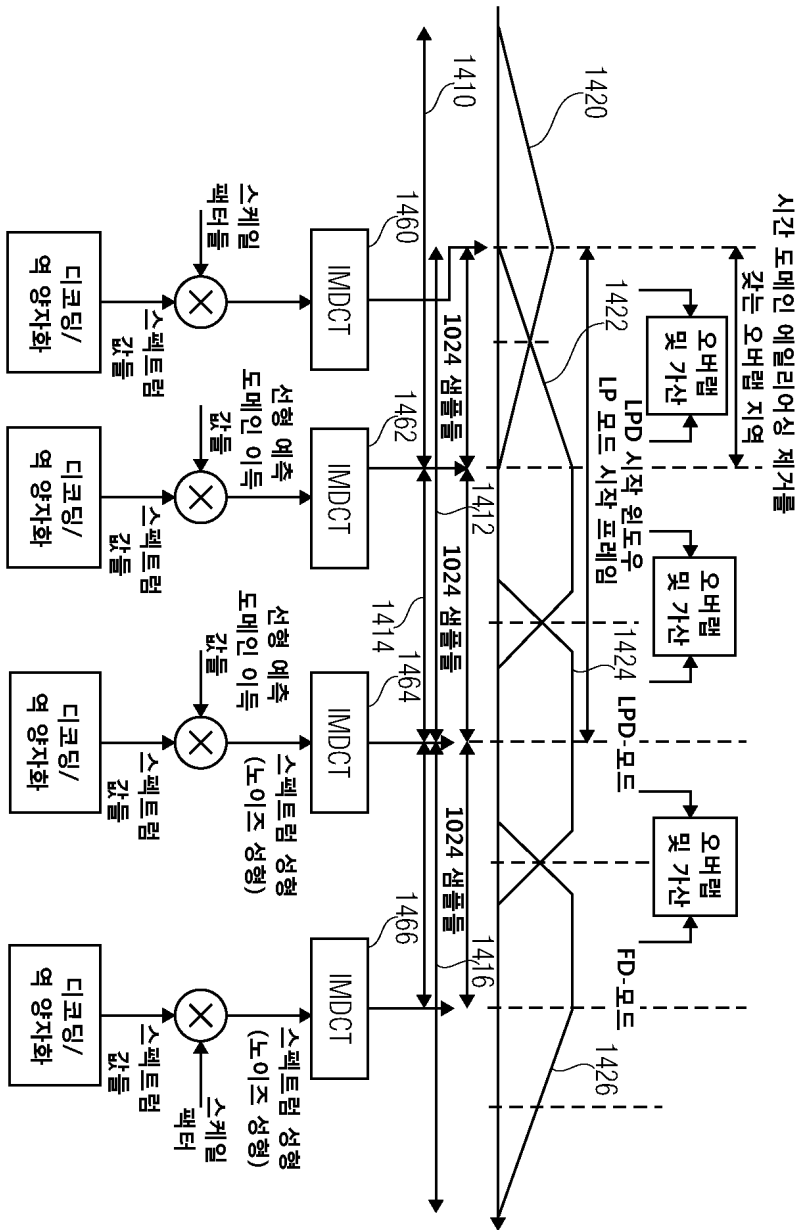
신호 도메인 내의 TCX-MDCT를 갖는 USAC 디코더의 통합뷰

| | |
|-------|-------|
| 도 12 | |
| 도 12a | 도 12b |

도면13b



도면14





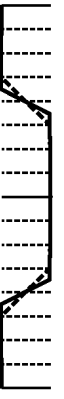


도면15

mod []의 함수로서 스펙트럼 계수의 수

| mod [x]의 값 | 스펙트럼 계수들의 수 N _g | ZL | L | M | R | ZR | 서브프레임 또는 프레임당 시간 도메인 샘플들의 수 (오버랩 포함) |
|------------|----------------------------|-----|-----|-----|-----|-----|--------------------------------------|
| 1 | 256 | 0 | 256 | 0 | 256 | 0 | 512 (서브프레임) |
| 2 | 512 | 128 | 256 | 256 | 256 | 128 | 1024 (서브프레임) |
| 3 | 1024 | 384 | 256 | 768 | 256 | 384 | 2048 (서브프레임) |

윈도우 시퀀스와 변환 윈도우들

| 값 | 윈도우 | #계수 | 윈도우 모양 |
|---|---|-------------|---|
| 0 | ONLY_LONG_SEQUENCE =LONG_WINDOW | 1024/960 |  |
| 1 | LONG_START_SEQUENCE =LONG_START_WINDOW | 1024/960 |  |
| 2 | EIGHT_SHORT_SEQUENCE =8*SHORT_WINDOW | 8*(128/120) |  |
| 3 | LONG_STOP_SEQUENCE =LONG_STOP_WINDOW | 1024/960 |  |
| 1 | STOP_START_SEQUENCE =STOP_START_WINDOW | 1024/960 |  |

도면16

점선은 인접한 윈도우 시퀀스가 LPD-SEQUENCE 일때 윈도우 모양을 지시한다

도면17a

허용된 윈도우 전이

| 윈도우 시퀀스 부터 ↓ 까지 → | ONLY_LONG_SEQUENCE | LONG_START_SEQUENCE | EIGHT_SHORT_SEQUENCE | LONG_STOP_SEQUENCE | STOP_START_SEQUENCE | LPD_SEQUENCE |
|----------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| ONLY_LONG_SEQUENCE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | |
| LONG_START_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| EIGHT_SHORT_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| LONG_STOP_SEQUENCE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | |
| STOP_START_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| LPD_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

도면17b

허용된 윈도우 전이

| 윈도우 시퀀스 부터 ↓ 까지 → | ONLY_LONG_SEQUENCE | LONG_START_SEQUENCE | EIGHT_SHORT_SEQUENCE | LONG_STOP_SEQUENCE | STOP_START_SEQUENCE | LPD_SEQUENCE | LPD_START_SEQUENCE |
|----------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| ONLY_LONG_SEQUENCE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | <input checked="" type="checkbox"/> |
| LONG_START_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| EIGHT_SHORT_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| LONG_STOP_SEQUENCE | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | | | | | <input checked="" type="checkbox"/> |
| STOP_START_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| LPD_SEQUENCE | | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| LPD_START_SEQUENCE | | | | | | <input checked="" type="checkbox"/> | |

도면18

