



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년03월16일
(11) 등록번호 10-1839629
(24) 등록일자 2018년03월12일

- (51) 국제특허분류(Int. Cl.)
HO4N 19/147 (2014.01) *HO4N 19/105* (2014.01)
HO4N 19/122 (2014.01) *HO4N 19/154* (2014.01)
HO4N 19/176 (2014.01) *HO4N 19/44* (2014.01)
HO4N 19/567 (2014.01)

(21) 출원번호 10-2012-7017703

(22) 출원일자(국제) 2010년12월02일
심사청구일자 2015년11월30일

(85) 번역문제출일자 2012년07월06일

(65) 공개번호 10-2012-0126067

(43) 공개일자 2012년11월20일

(86) 국제출원번호 PCT/US2010/003075

(87) 국제공개번호 WO 2011/071514
국제공개일자 2011년06월16일

(73) 특허권자
톰슨 라이센싱
프랑스 이씨레몰리노 르 잔 다르크 1-5 (우:
92130)

(72) 발명자
정, 윤페이
미국 캘리포니아주 95762 샌디에고 아파트먼트
5205 토레이 서클 4615

인, 평
미국 뉴욕주 14850 이타카 존 스트리트 6
(뒷면에 계속)

(74) 대리인
특허법이 아주

(30) 우선권주장
61/267 715 2009년12월08일 미국(US)

(56) 서해기숙조사문현

KR1020080019294 A*

T. K. Tan, et al. Intra prediction by averaged template matching predictors. 4th IEEE Consumer Comm. and Networking Conf. Jan. 2007, pp. 405-409.

T. K. Tan, et al. Intra prediction by template matching. IEEE Int'l Conf. on Image Processing, Oct. 2006, pp. 1693-1696.

JP2007300380 A

*는 심사관에 의하여 이용된 문헌

전체 청구항 수 : 총 16 항

심사관 : 이상래

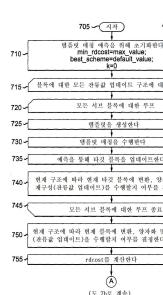
(54) 발명의 명칭 비디오 익코딩 및 디코딩을 위한 적응적 템플릿 매칭 예측 방법 및 장치

(57) 요약

비디오 인코딩 및 디코딩을 위한 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치가 제공된다. 일 실시예에서, 장치 및 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하되, 여기서 삼기 템플릿 매칭 예측은 삼기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트를 위한 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치가 제공된다.

(뒷면에 계속)

대 표 도



이트 구조들 중에서 선택하는 (715, 740, 750, 775, 935, 945) 것을 포함한다. 다른 실시예에서, 장치 및 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하되, 여기서 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 (720, 725, 730, 735, 915, 920, 925, 930, 1115, 1120, 1125, 1130, 1150, 1155, 1160, 1165) 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 (740, 750, 935, 945, 1135, 1175) 것을 포함한다.

(72) 발명자

솔, 조엘

미국 캘리포니아주 92037 라 졸라 유닛 106 빌라
라 졸라 드라이브 8722

루, 시아오안

미국 뉴저지주 08540 프린스턴 케네디 코트 30

수, 치안

미국 캘리포니아주 95762 엘 도라도 힐스 콘코디아
드라이브 4795

명세서

청구범위

청구항 1

템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 비디오 인코더(500)를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들(scheme) 중에서 선택하는 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 각각은 상기 블록을 복수의 서브 블록으로 분할하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 다른 하나는 상기 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료한 후에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 (740, 750, 935, 945, 1135, 1175) 것인 장치.

청구항 2

비디오 인코더에서 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (715, 740, 750, 775, 935, 945) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 각각은 상기 블록을 복수의 서브 블록으로 분할하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 다른 하나는 상기 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료한 후에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 (740, 750, 935, 945, 1135, 1175) 것인 방법.

청구항 3

제2항에 있어서, 상기 복수의 잔류값 업데이트 구조들 중 적어도 하나는 율-왜곡 비용(rate-distortion cost)에 기초하여 선택되는 (760, 765, 775) 것인 방법.

청구항 4

삭제

청구항 5

제2항에 있어서, 상기 복수의 잔류값 업데이트 구조들은 최종 비트스트림에 명시적으로 표시되는 (780) 것인 방법.

청구항 6

제2항에 있어서, 상기 복수의 잔류값 업데이트 구조들 중 적어도 하나는 상기 화상과 동일한 비디오 시퀀스에 대응하는 디코딩된 화상들, 상기 블록 중 적어도 하나의 블록의 양자화 파라미터 값, 상기 블록 내 적어도 하나의 서브 블록, 및 상기 블록이나 상기 적어도 하나의 서브 블록에 대하여 적어도 하나의 인접한 블록, 화상 컨텐츠 통계, 파티션 정보, 및 상기 적어도 하나의 인접한 블록의 코딩 정보 중 적어도 하나에 기초하여 선택되는 (910) 것인 방법.

청구항 7

제2항에 있어서, 상기 복수의 잔류값 업데이트 구조들 중 2개 이상이 상기 블록을 예측하기 위해 결합될 수 있

는 (1100) 것인 방법.

청구항 8

비디오 인코더에서 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (715, 740, 750, 775, 935, 945) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 중 2개 이상이 상기 블록을 예측하기 위해 결합될 수 있는 (1100) 것이며,

상기 복수의 잔류값 업데이트 구조들 각각은 서로에 대하여 독립적이고, 상기 블록의 최종 예측은 상기 복수의 잔류값 업데이트 구조들 중 적어도 2개의 구조의 출력의 결합인 (1300) 것인 방법.

청구항 9

비디오 인코더에서 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (715, 740, 750, 775, 935, 945) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 중 2개 이상이 상기 블록을 예측하기 위해 결합될 수 있는 (1100) 것이며,

상기 복수의 잔류값 업데이트 구조들은 서로 종속적이고, 상기 복수의 잔류값 업데이트 구조들 중 특정한 것의 하나의 출력은 상기 복수의 잔류값 업데이트 구조들 중 적어도 하나의 다른 특정한 것의 적어도 하나의 다른 출력에 의존하는 (1100) 것인 방법.

청구항 10

삭제

청구항 11

삭제

청구항 12

비디오 디코더에서, 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 디코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (810, 835, 845, 1015, 1040, 1050) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 각각은 상기 블록을 복수의 서브 블록으로 분할하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 다른 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료한 후에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 (835, 845, 1015, 1040, 1050) 것을 포함하는 것인 방법.

청구항 13

삭제

청구항 14

삭제

청구항 15

비디오 디코더에서, 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 디코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (810, 835, 845, 1015, 1040, 1050) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 중 2개 이상은 상기 블록을 예측하기 위해 결합될 수 있는 (1200) 것이고,

상기 복수의 잔류값 업데이트 구조들 각각은 서로에 대하여 독립적이고 상기 블록의 최종 예측은 상기 복수의 잔류값 업데이트 구조들 중 적어도 2개의 구조의 출력의 결합인 (1400) 것인 방법.

청구항 16

비디오 디코더에서, 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 디코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 (810, 835, 845, 1015, 1040, 1050) 것을 포함하는 것이고,

상기 복수의 잔류값 업데이트 구조들 중 2개 이상은 상기 블록을 예측하기 위해 결합될 수 있는 (1200) 것이고,

상기 복수의 잔류값 업데이트 구조들은 서로 종속적이며, 상기 복수의 잔류값 업데이트 구조들 중 특정한 것의 하나의 출력은 상기 복수의 잔류값 업데이트 구조들 중 적어도 하나의 다른 특정한 것의 적어도 하나의 다른 출력에 의존하는 (1200) 것인 방법.

청구항 17

삭제

청구항 18

템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하는 비디오 인코더(500)를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 것인 장치.

청구항 19

비디오 인코더에서 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 (720, 725, 730, 735, 915, 920, 925, 930, 1115, 1120, 1125, 1130, 1150, 1155, 1160, 1165) 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 (740, 750, 935, 945, 1135, 1175) 것을 포함하는 것인 방법.

청구항 20

템플릿 매칭 예측을 사용하여 화상에 있는 블록을 디코딩하는 비디오 디코더(600)를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 것인 장치.

청구항 21

비디오 디코더에서 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 디코딩하는 단계를 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 (815, 820, 825, 830, 1020, 1025, 1030, 1035, 1215, 1220, 1225, 1230, 1245, 1250, 1255, 1260) 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 (835, 845, 1040, 1050, 1235, 1270) 것을 포함하는 것인 방법.

청구항 22

인코딩된 비디오 신호 데이터를 구비하는 비-일시적인 컴퓨터 관독가능한 저장 매체로서,

템플릿 매칭 예측을 사용하여 인코딩된 화상 내 블록을 포함하되, 상기 템플릿 매칭 예측은 상기 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후 및 상기 템플릿 매칭 예측에 대

하여 상기 블록의 처리를 완료하기 전에 상기 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 것인 컴퓨터 판독가능한 저장 매체.

발명의 설명

기술 분야

- [0001] 본 출원은 그 전체 내용이 본 명세서에 참조 문헌으로 병합된 2009년 12월 8일에 출원된 미국 가출원 일련 번호 61/267,715호의 이익을 청구한다.
- [0002] 본 발명의 원리는 일반적으로 비디오 인코딩 및 디코딩에 관한 것이고, 보다 상세하게는 비디오 인코딩 및 디코딩을 위한 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치에 관한 것이다.

배경 기술

- [0003] 템플릿 매칭 예측(TMP: template matching prediction)은 움직임/변위 정보(움직임 벡터, 참조 색인 및 변위 벡터)의 전송을 회피하는 것에 의해 인터 예측 및 인트라 예측에 대한 코딩 효율을 증가시킬 수 있다. 그러나, TMP를 사용하는 것으로부터 얻어지는 예측은 타깃 블록(target block)과 그 인접한 픽셀(즉, 템플릿) 사이의 상관성에 매우 의존한다. 따라서, 템플릿 매칭은 통상적으로 처리 단위로 2x2와 같은 상대적으로 작은 블록을 사용하거나 심지어 하나의 픽셀을 사용한다. 많은 블록 기반 인코더 및 디코더에서, 기본 블록 사이즈는 템플릿 매칭에 의해 처리되는 블록 단위보다 통상 더 큰 변환 사이즈에 의해 결정된다. 기본 블록 사이즈는 템플릿 매칭 예측을 위해 서브 블록들로 더 분할될 수 있다. 템플릿 매칭이 검색하는데 사용되는 템플릿으로 인접한 픽셀(예를 들어, 상부 및 하부 인접 픽셀)을 사용하므로, 일부 서브 블록의 템플릿은 인접한 서브 블록으로부터의 픽셀 일부를 포함할 수 있다. 그러나, 기본 블록이 완전히 예측되었을 때에만 변환이 적용될 수 있으므로 전체 기본 블록이 예측된 후에야 잔류값(residual)이 예측에 추가될 수 있다. 따라서, 템플릿의 일부 부분은 잔류값을 추가하는 것에 의해 재구성되지 않는 일부 예측된 데이터를 포함할 수 있다. 템플릿에 예측된 데이터를 사용하는 것이 서브 블록의 경계면에 있을 수 있는 블록화 현상(blockiness)을 회피할 수 있으나, 이러한 사용은 예측된 데이터가 재구성된 데이터보다 더 많은 정보 손실을 가질 수 있어서 낮은 품질의 매칭 결과를 야기할 수 있다.

- [0004] 기본 템플릿 매칭 예측은 비디오 화상에 많은 반복 패턴이 존재한다는 가정에 기초한다. 이를 고려하여, 템플릿 매칭은 인접한 픽셀을 매칭하는 것에 의해 디코딩된 비디오 화상에서 이와 유사한 패턴을 검색한다. 역 예측 특성으로 인해, 템플릿 매칭은 움직임 벡터나 변위 벡터와 같은 오버헤드의 전송을 회피하여 코딩 성능을 개선할 수 있다. 나아가, 템플릿 매칭은 인터 및 인트라 예측에 모두 사용될 수 있다.

인터 예측에서 템플릿 매칭 예측

- [0006] 인터 예측에서 템플릿 매칭 예측은 움직임 벡터를 송신함이 없이 타깃 픽셀을 예측하는 하나의 방법이다. 프레임의 타깃 블록이 주어질 때, 블록 내 타깃 픽셀은 참조 샘플의 세트로부터 최적의 픽셀을 찾는 것에 의해 결정되며, 여기서 최적의 픽셀의 인접한 픽셀은 타깃 픽셀의 것과 최고 높은 상관성을 가진다. 타깃 픽셀의 이들 인접한 픽셀은 템플릿이라고 불리운다. 종래 기술에서, 이 템플릿은 타깃 픽셀의 재구성된 주변 픽셀로부터 통상 취해진다. 도 1을 참조하면 인터 예측을 위한 템플릿 매칭 예측 구조(scheme)의 일례가 일반적으로 참조 부호 100으로 표시된다. 템플릿 매칭 예측 구조(100)는 검색 영역(111), 이 검색 영역(111) 내 예측 영역(112) 및 이 예측 영역(112)에 대하여 인접한 영역(113)을 가지는 재구성된 참조 프레임(110)을 포함한다. 템플릿 매칭 예측 구조(100)는 또한 타깃 블록(151), 이 타깃 블록(151)에 대하여 템플릿 영역(152) 및 재구성된 영역(153)을 가지는 현재 프레임(150)을 포함한다. 인터-예측의 경우에, 템플릿 매칭 공정은 디코팅 측에서 움직임 벡터 검색으로 볼 수 있다. 여기서, 템플릿 매칭은 전통적인 움직임 추정 기술과 매우 유사하게 수행된다. 즉, 움직임 벡터는 참조 프레임에서 적절히 변위된 템플릿 형성된 영역에 대한 비용 함수를 계산하는 것에 의해 평가된다. 템플릿을 위한 최상의 움직임 벡터는 이후 타깃 영역을 예측하는데 사용된다. 재구성 또는 적어도 예측 신호가 이미 존재하는 이미지의 영역들만이 검색을 위해 액세스된다. 따라서, 디코더는 템플릿 매칭 공정을 실행하고 추가적인 부가 정보 없이 타깃 영역을 예측할 수 있다.

- [0007] 템플릿 매칭은 움직임 벡터의 전송 없이 타깃 블록 내 픽셀을 예측할 수 있다. 템플릿 매칭 예측의 예측 성능은 타깃 블록과 그 템플릿 사이의 상관성이 높은 경우 전통적인 블록 매칭 구조의 것과 필적할 수 있는 것으로 예상된다. 종래 기술에서, 템플릿은 타깃 픽셀의 재구성된 공간 인접 픽셀로부터 취해진다. 인접한 픽셀은 종종 타깃 픽셀과 낮은 상관성을 가진다. 따라서, 템플릿 매칭 예측의 성능은 전통적인 블록 매칭 구조보다 더 낮을

수 있다.

[0008] 인트라 예측에서 템플릿 매칭 예측

[0009] 인트라 예측에서 템플릿 매칭은 예측이 타깃 블록으로부터 먼 픽셀에 의해 생성될 수 있으므로 이용가능한 비-국부 예측 접근법들 중 하나의 접근법이다. 인트라 템플릿 매칭에서 템플릿의 정의는 인터 템플릿 매칭의 것과 유사하다. 그러나, 하나의 차이는 검색 영역이 현재 화상의 디코딩된 부분으로 제한된다는 것이다. 도 2를 참조하면, 인트라 예측을 위한 템플릿 매칭 예측 구조의 일례가 일반적으로 참조 부호 200으로 표시된다. 템플릿 매칭 예측 구조(200)는 화상(277)의 디코딩된 부분(210)을 포함한다. 화상(277)의 디코딩된 부분(210)은 검색 영역(211), 이 검색 영역(211) 내 후보 예측 영역(212) 및 이 후보 예측 영역(212)에 대하여 인접한 영역(213)을 가진다. 템플릿 매칭 예측 구조(200)는 또한 화상(277)의 디코딩되지 않은 부분(220)을 포함한다. 화상(277)의 디코딩되지 않은 부분(220)은 타깃 블록(221), 이 타깃 블록(221)에 대하여 템플릿 영역(222)을 구비한다. 간략화를 위해, 이후 설명은 인트라 템플릿 매칭에 기초한다. 그러나, 이 기술 분야 및 관련 분야에 통상의 지식을 가진 자라면 인터 템플릿 매칭으로 용이하게 확장될 수 있다는 것을 이해할 수 있을 것이다.

[0010] 템플릿 매칭에서 잔류값 업데이트 구조

[0011] 템플릿 매칭이 추가적인 오버헤드를 회피할 수 있으므로, 이것은 비디오 화상 내 상대적으로 입상(grainy)인 영역에 통상 요구되는 수반하는 추가적인 오버헤드를 최소화하는 것에 의해 이 비디오 화상 내 상대적으로 입상인 영역을 예측하는 장점을 구비한다. 템플릿 매칭의 장점을 이용하기 위하여, 예측 동안 타깃 블록은 통상 기본 코딩 블록보다 더 작다. 예를 들어, ISO/IEC(International Organization for Standardization/International Electrotechnical Commission) MPEG-4(Moving Picture Experts Group-4) Part 10 AVC (Advanced Video Coding) Standard/ITU-T(International Telecommunication Union, Telecommunication Sector) H.264 Recommendation (이후 "MPEG-4 AVC 표준")에서, 기본 코딩 블록 사이즈는 4x4 또는 8x8일 수 있다. 하나의 종래 기술의 접근법에서, 템플릿 매칭의 타깃 블록 사이즈는 기본 코딩 블록 4x4 사이즈의 절반인 2x2로 한정된다. 도 3을 참조하면, 인트라 템플릿 매칭 예측을 위한 서브 분할된 블록의 일례가 일반적으로 참조 부호 300으로 표시된다. 인트라 템플릿 매칭 예측은 화상(377)의 디코딩된 부분(310)을 포함한다. 화상(377)의 디코딩된 부분(310)은 검색 영역(311)과 이 검색 영역(311) 내 후보 예측 영역(312) 및 이 후보 예측 영역(312)에 대하여 인접한 영역(313)을 구비한다. 인트라 템플릿 매칭 예측은 또한 화상(377)의 디코딩되지 않은 부분(320)을 포함한다. 화상(377)의 디코딩되지 않은 부분(320)은 타깃 블록(321)(이는 또한 도 3의 예에서 인트라 템플릿 매칭 예측을 위한 서브 분할된 블록이라고도 언급된다)과 이 타깃 블록(321)에 대하여 템플릿 영역(322)을 구비한다.

[0012] 변환 사이즈는 기본 블록 사이즈(MPEG-4 AVC 표준에서는 4x4 변환 및 8x8 변환이 있다)와 통상 동일하므로, 이는 적어도 전체 기본 블록이 예측된 후에 변환된, 양자화된 및 업데이트된 예측 잔류값 만을 취할 수 있다는 것을 의미한다. 따라서, 더 작은 타깃 블록 사이즈를 가지는 템플릿 매칭에서 템플릿은 이전의 예측된 서브 블록으로부터 오는 일부 예측 데이터를 포함할 수 있다. 본 명세서에서 이 접근법은 "방법 1"이라고 언급된다.

[0013] 도 4a 내지 도 4d를 참조하면, 서브 블록 및 그 템플릿의 일례가 일반적으로 참조 부호 400으로 표시된다. 도 4의 예는 전술된 "방법 1"에 대응한다. 이 예에서, 기본 블록(410)은 4개의 서브 블록들(이는 각각 참조 부호 1, 2, 3 및 4로 표시된다)로 분할되며, 이 서브 블록들 각각은 템플릿 매칭 예측의 각 타깃 블록이다. 도 4a를 참조하면, 타깃 블록(1)의 템플릿은 참조 부호 411로 표시되며, (기본 블록에 적용된) 템플릿 매칭 예측의 타깃 블록은 참조 부호 412로 표시되며, 기본 블록 사이즈는 참조 부호 413으로 표시된다. 도 4b를 참조하면, 타깃 블록(2)의 템플릿은 참조 부호 421 및 422로 표시된 영역으로 표시된다. 도 4c를 참조하면, 타깃 블록(3)의 템플릿은 참조 부호 431 및 432로 표시된 영역으로 표시된다. 도 4d를 참조하면, 타깃 블록(4)의 템플릿은 참조 부호 441, 442 및 443으로 표시된 영역으로 표시된다. 4개의 타깃 블록들의 예측 공정은 다음 단계들을 포함한다:

- [0014] (1) 인접한 픽셀들을 사용하는 것에 의해 타깃 블록(예를 들어, 타깃 블록 1)에 대한 템플릿을 생성한다;
- [0015] (2) 매치(match)를 검색하고 이 타깃 블록에 대한 예측을 생성한다;
- [0016] (3) 그 예측을 통해 타깃 블록을 업데이트한다;
- [0017] (4) 타깃 블록 2,3 및 4에 대해 단계(1) 내지 단계(3)를 반복한다;
- [0018] (5) 전체 기본 블록의 잔류값 블록을 획득한다;

- [0019] (6) 전체 잔류값 블록을 변환하고, 계수를 양자화하고, 양자화된 계수를 역변환한다;
- [0020] (7) 예측 블록과 양자화된 잔류값 블록을 추가하는 것에 의해 재구성된 블록을 생성한다;
- [0021] (8) 기본 블록을 종료한다.
- [0022] 타깃 블록 1이 블록의 제1 서브 블록이므로, 그 템플릿에 있는 픽셀은 모두 재구성된 화상으로부터 온다. 그러나, 타깃 블록 2와 같은 다른 타깃 블록들에 대해서는 그 템플릿이 타깃 블록 1로부터 온 픽셀 일부를 가질 필요가 있다. 따라서, 타깃 블록 2를 예측할 때, 타깃 블록 1은 방금 예측되었지만 변환 및 양자화가 4개의 타깃 블록들이 단계(6)에서 예측될 때까지 수행되지 않았으므로 재구성되지 않았다. 따라서, 타깃 블록 2의 템플릿을 위해 그 템플릿의 일부 부분은 타깃 블록 1의 예측 데이터로부터 온다. 동일한 문제가 템플릿 매칭에 의하여 다른 타깃 블록(즉, 타깃 블록 1과는 다른 블록, 그 이유는 전술된 바와 같이, 타깃 블록 1에 대한 템플릿 내 픽셀이 모두 재구성된 화상으로부터 온 것이기 때문이다)을 예측할 때에도 존재한다.
- [0023] 이전의 타깃 블록의 예측 데이터를 현재 타깃 블록의 템플릿에 병합시키는 것은 전체 기본 블록의 예측을 평활하게 변하게 만들 수 있다. 나아가, 큰 변환(및 수반하는 양자화)을 적용하여 타깃 블록들 전부가 완전히 예측될 때에는 큰 변환을 사용하는 것으로부터 장점을 얻을 수 있다. 그러나, 매칭을 위한 템플릿에 예측 데이터를 병합하는 것은 예측 에러를 전파시켜 예측을 악화시킬 수 있다.

발명의 내용

- [0024] 종래 기술의 이들 및 다른 단점 및 불리함은 비디오 코딩을 위한 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치에 관한 본 발명의 원리에 의해 해결된다.
- [0025] 본 발명의 원리의 일 측면에 따라, 장치가 제공된다. 본 장치는 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 비디오 인코더를 포함한다. 템플릿 매칭 예측은 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 것을 포함한다.
- [0026] 본 발명의 원리의 다른 측면에 따라, 비디오 인코더에서 방법이 제공된다. 본 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 단계를 포함한다. 템플릿 매칭 예측은 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 것을 포함한다.
- [0027] 본 발명의 원리의 또 다른 측면에 따라, 장치가 제공된다. 본 장치는 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 디코딩하는 비디오 디코더를 포함한다. 템플릿 매칭 예측은 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 것을 포함한다.
- [0028] 본 발명의 원리의 더 다른 측면에 따라, 비디오 디코더에서 방법이 제공된다. 본 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 화상 데이터를 디코딩하는 단계를 포함한다. 템플릿 매칭 예측은 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 것을 포함한다.
- [0029] 본 발명의 원리의 다른 측면에 따라, 장치가 제공된다. 본 장치는 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하는 비디오 인코더를 포함한다. 템플릿 매칭 예측은 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함한다.
- [0030] 본 발명의 더 다른 측면에 따라, 비디오 인코더에서 방법이 제공된다. 본 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하는 단계를 포함한다. 템플릿 매칭 예측은 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함한다.
- [0031] 본 발명의 또 다른 측면에 따라, 장치가 제공된다. 본 장치는 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 디코딩하는 비디오 디코더를 포함한다. 템플릿 매칭 예측은 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함한다.
- [0032] 본 발명의 추가적인 측면에 따라, 비디오 디코더에서 방법이 제공된다. 본 방법은 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 디코딩하는 단계를 포함하되, 여기서 템플릿 매칭 예측은 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리

리를 완료하기 전에 복수의 서브 블록들 중에서 상기 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함한다.

[0033] 본 발명의 원리의 이들 및 다른 측면, 특징 및 장점은 첨부 도면과 연계하여 관독하는 예시적인 실시예의 이하 상세 설명으로부터 명백해질 것이다.

도면의 간단한 설명

[0034] 본 발명의 원리는 이하 예시적인 도면에 따라 더 잘 이해될 것이다.

도 1은 인터 예측을 위한 템플릿 매칭 예측 구조의 일례를 도시하는 다이아그램;

도 2는 인트라 예측을 위한 템플릿 매칭 예측 구조의 일례를 도시하는 다이아그램;

도 3은 인트라 템플릿 매칭 예측을 위한 서브 분할된 블록의 일례를 도시하는 다이아그램;

도 4a 내지 도 4d는 서브 블록과 그 템플릿의 일례를 집합적으로 도시하는 다이아그램;

도 5는 본 발명의 원리의 일 실시예에 따라 본 발명의 원리가 적용될 수 있는 예시적인 비디오 인코더를 도시하는 블록도;

도 6은 본 발명의 원리의 일 실시예에 따라 본 발명의 원리가 적용될 수 있는 예시적인 비디오 디코더를 도시하는 블록도;

도 7a 및 도 7b는 본 발명의 원리의 일 실시예에 따라 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 예시적인 방법을 도시하는 흐름도;

도 8은 본 발명의 원리의 일 실시예에 따라 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 예시적인 방법을 도시하는 흐름도;

도 9는 본 발명의 원리의 일 실시예에 따라 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 다른 예시적인 방법을 도시하는 흐름도;

도 10은 본 발명의 원리의 일 실시예에 따라 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 다른 예시적인 방법을 도시하는 흐름도;

도 11a 및 도 11b는 본 발명의 원리의 일 실시예에 따라 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이터를 위한 또 다른 예시적인 방법을 도시하는 흐름도;

도 12는 본 발명의 원리의 일 실시예에 따라 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 또 다른 예시적인 방법을 도시하는 흐름도;

도 13은 본 발명의 원리의 일 실시예에 따라 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 더 다른 예시적인 방법을 도시하는 흐름도;

도 14는 본 발명의 원리의 일 실시예에 따라 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 더 다른 예시적인 방법을 도시하는 흐름도.

발명을 실시하기 위한 구체적인 내용

[0035] 본 발명의 원리는 비디오 인코딩 및 디코딩에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치에 관한 것이다.

[0036] 본 설명은 본 발명의 원리를 예시한다. 따라서, 이 기술 분야에 통상의 지식을 가진 자라면 본 명세서에 명시적으로 기술되거나 도시되지는 않았을지라도 본 발명의 사상과 범위 내에 포함되고 본 발명의 원리를 구현하는 여러 배열을 고안할 수 있을 것이라는 것을 이해할 수 있을 것이다.

[0037] 본 명세서에 언급된 모든 예시와 조건적 언어들은 이 기술을 개선하려고 발명자(들)가 기여한 본 발명의 원리와 개념을 독자들이 이해하는 것을 돋기 위한 설명을 위한 목적으로 의도된 것이며, 그러한 구체적으로 언급된 예시와 조건으로 제한하려는 것이 아닌 것으로 해석되어야 한다.

[0038] 나아가, 본 발명의 원리, 측면 및 실시예 뿐만 아니라 특정 예시를 언급하는 모든 진술은 구조적 및 기능적으로 균등한 것을 포함하는 것으로 의도된 것이다. 부가적으로, 그러한 균등률은 현재 알려진 균등률 뿐만 아니라 미

래에 개발될 균등물, 즉 구조에 상관없이 동일한 기능을 수행하는 개발된 임의의 요소를 포함한다는 것으로 의도된다.

[0039] 따라서, 예를 들어, 이 기술 분야에 통상의 지식을 가진 자라면 본 명세서에 제시된 블록도가 본 발명의 원리를 구현하는 예시적인 회로의 개념적인 관점을 제시하는 것임을 이해할 수 있을 것이다. 이와 유사하게, 임의의 흐름도, 흐름 선도, 상태 전이도, 의사코드 등은 컴퓨터나 프로세서가 명시적으로 도시되지 않았을지라도 컴퓨터로 판독가능한 매체에 실질적으로 제공되고 컴퓨터나 프로세서에 의해 수행될 수 있는 여러 공정을 나타낸다는 것을 이해할 수 있을 것이다.

[0040] 도면에 도시된 여러 요소의 기능은 적절한 소프트웨어와 연관된 소프트웨어를 실행할 수 있는 하드웨어 뿐만 아니라 전용 하드웨어의 사용을 통해 제공될 수 있다. 프로세서에 의해 제공될 때, 그 기능은 단일 전용 프로세서에 의해, 단일 공유 프로세서에 의해 또는 일부가 공유될 수 있는 복수의 개별 프로세서에 의해 제공될 수 있다. 나아가, "프로세서" 또는 "제어기"라는 용어의 명시적인 사용이 소프트웨어를 실행할 수 있는 하드웨어만을 배타적으로 말하는 것으로 해석되어서는 아니되며, 디지털 신호 프로세서("DSP") 하드웨어, 소프트웨어를 저장하는 판독 전용 메모리("ROM"), 랜덤 액세스 메모리("RAM") 및 비휘발성 저장장치를 암시적으로 제한없이 포함할 수 있다.

[0041] 종래의 것이든 및/또는 주문형이든 상관없이 다른 하드웨어가 또한 포함될 수 있다. 이와 유사하게, 이 도면에 도시된 임의의 스위치는 단지 개념적인 것이다. 그 기능은 프로그램4 논리의 동작을 통해, 전용 논리를 통해, 프로그램 제어와 전용 논리의 상호작용을 통해 또는 심지어 수동으로 수행될 수 있으며, 특정 기술은 문맥으로부터 보다 구체적으로 이해되는 바와 같이 구현하는 자에 의해 선택될 수 있다.

[0042] 특허청구범위에서, 특정 기능을 수행하는 수단으로 표시된 임의의 요소는 예를 들어 a) 그 기능을 수행하는 회로 요소의 조합이나 b) 그 기능을 수행하기 위해 그 소프트웨어를 실행하는 적절한 회로와 결합된 펌웨어, 마이크로 코드 등을 포함하는 임의의 형태의 소프트웨어를 포함하여 그 기능을 수행하는 임의의 방법을 포함하는 것으로 의도된다. 특허청구범위에 의해 한정된 본 발명의 원리는 여러 언급된 수단으로 제공된 기능이 조합되며 특허청구범위가 요청하는 방식으로 서로 결합된 것에 존재한다. 따라서, 그 기능을 제공할 수 있는 임의의 수단은 본 명세서에 도시된 것과 균등한 것이라고 간주된다.

[0043] 명세서에서 본 발명의 원리의 "일 실시예" 또는 "실시예"라는 언급과 그 다른 변형 어구의 언급은 실시예와 연관하여 기술된 특정 특징, 구조, 특성 등이 본 발명의 원리의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서, 명세서 전체에 걸쳐 여러 곳에 나타나는 "하나의 실시예에서" 또는 "실시예에서"라는 어구의 표현과 그 임의의 다른 변형 어구는 반드시 동일한 실시예를 모두 언급하는 것이 아닐 수 있다. 나아가, 실시예라는 어구와 구현예라는 어구는 본 명세서에서 교환가능하게 사용된다는 것을 이해할 수 있을 것이다.

[0044] 예를 들어 "A/B", "A 및/또는 B" 그리고 "A 및 B 중 적어도 하나"에 있는 "/", " 및/또는" 및 "~ 중 적어도 하나" 중 어느 하나의 사용은 처음 나열된 옵션(A)만을 선택하거나 두 번째 나열된 옵션(B)만을 선택하거나 두 개의 옵션(A와 B)을 선택하는 것을 포함하는 것으로 의도된 것이라는 것을 이해하여야 한다. 다른 예로서, "A, B 및/또는 C" 그리고 "A, B 및 C 중 적어도 하나"의 경우에, 이 어구는 처음 나열된 옵션(A)만을 선택하거나, 두 번째 나열된 옵션(B)만을 선택하거나 세 번째 나열된 옵션(C)만을 선택하거나 처음과 두 번째 나열된 옵션(A와 B)만을 선택하거나, 처음과 세 번째 나열된 옵션(A와 C)만을 선택하거나 두 번째와 세 번째 나열된 옵션(B와 C)만을 선택하거나 3개의 옵션(A와 B와 C)을 모두 선택하는 것을 포함하는 것으로 의도된 것이다. 이것은 이 기술 분야 및 관련 기술 분야의 통상의 지식을 가진 자에게는 명백 바와 같이 많은 항목을 나열한 것에 대해서도 확장될 수 있다.

[0045] 또한, 본 명세서에 사용된 바와 같이, "화상" 및 "이미지"라는 용어는 상호 교환 가능하게 사용되며 비디오 시퀀스로부터 정지 이미지 또는 정지 화상을 말한다. 알려진 바와 같이, 화상은 프레임이나 필드일 수 있다.

[0046] 추가적으로, 본 명세서에 사용된 바와 같이, "신호"라는 용어는 대응하는 디코더로 가는 어떤 것을 나타내는 것을 말한다. 예를 들어, 인코더는, 인코더 측에서 사용된 특정 잔류값 업데이트 구조가 무엇인지 디코더에 알려주기 위하여 하나 이상의 잔류값 업데이트 구조를 신호 송신할 수 있다. 이런 방식으로, 동일한 잔류값 업데이트 구조가 인코더 측과 디코더 측에서 모두 사용될 수 있다. 따라서, 예를 들어, 인코더는 특정한 하나 이상의 잔류 업데이트 구조를 디코더에 송신하여 디코더가 동일한 특정 잔류 업데이트 구조를 사용하게 하거나 디코더가 다른 것과 함께 이미 특정한 하나 이상의 잔류값 업데이트 구조를 가지고 있다면 디코더로 하여금 잔류값 업데이트 구조의 더 큰 그룹으로부터 단순히 특정한 하나 이상의 잔류값 업데이트 구조를 알고 선택할 수 있게

하기 위해 (전송 없이) 신호 송신될 수 있다. 임의의 실제 잔류값 업데이트 구조의 전송을 회피하는 것에 의해 비트 절감이 실현될 수 있다. 이 신호 송신은 여러 방식으로 달성될 수 있다는 것을 이해할 수 있을 것이다. 예를 들어, 하나 이상의 구문 요소, 플래그 등이 대응하는 디코더에 정보를 신호 송신하는데 사용될 수 있다.

[0047] 나아가, 본 명세서에 사용된 바와 같이, "기본 블록(basic block)"이라는 어구는 인코더와 디코더에서 미리 한정된 변환이 적용되는 블록을 말한다.

[0048] 예시와 설명을 위하여, 예들은 본 명세서에서 설명을 위한 베이스라인으로 MPEG-4 AVC 표준을 사용하고 MPEG-4 AVC 표준을 넘어 개선과 확장을 설명하는 MPEG-4 AVC 표준에 비해 개선의 상황에서 설명된 것이다. 그러나, 본 발명의 원리는 MPEG-4 AVC 표준 및/또는 그 확장으로만 제한되는 것은 아니라는 것을 이해하여야 할 것이다. 본 명세서에 제공되는 본 발명의 원리의 개시 내용을 통해 이 기술 분야 및 관련 기술 분야에 통상의 지식을 가진 자라면 본 발명의 원리가 다른 표준의 확장에 적용될 때에나 아직 개발되지 않은 표준에 적용되거나 및/또는 이 표준에 포함될 때에도 동일하게 적용가능하고 적어도 유사한 이익을 제공할 수 있다는 것을 용이하게 이해할 수 있을 것이다. 나아가 본 발명의 원리는 또한 표준에 순응하지 않고 오히려 독점적 정의에 순응하는 비디오 인코더와 비디오 디코더에도 적용가능하다는 것을 더 이해할 수 있을 것이다.

[0049] 도 5를 참조하면, 본 발명의 원리가 적용될 수 있는 예시적인 비디오 인코더가 일반적으로 참조 부호 500으로 표시된다. 비디오 인코더(500)는 가산기(585)의 비반전 입력과 신호 통신가능한 출력을 구비하는 프레임 정렬 버퍼(510)를 포함한다. 가산기(585)의 출력은 변환기 및 양자화기(525)의 제1 입력과 신호 통신가능하게 연결된다. 변환기 및 양자화기(525)의 출력은 엔트로피 코더(545)의 제1 입력과 역변환기 및 역양자화기(550)의 제1 입력과 신호 통신가능하게 연결된다. 엔트로피 코더(545)의 출력은 가산기(590)의 제1 비반전 입력과 신호 통신가능하게 연결된다. 가산기(590)의 출력은 출력 버퍼(535)의 제1 입력과 신호 통신가능하게 연결된다.

[0050] 인코더 제어기(505)의 제1 출력은 프레임 정렬 버퍼(510)의 제2 입력, 역변환기 및 역양자화기(550)의 제2 입력, 화상 유형 결정 모듈(515)의 입력, 매크로블록 유형(MB 유형) 결정 모듈(520)의 제1 입력, 인트라 예측 모듈(560)의 제2 입력, 디블록킹 필터(565)의 제2 입력, 움직임 보상기(570)의 제1 입력, 움직임 추정기(575)의 제1 입력, 및 참조 화상 버퍼(580)의 제2 입력과 신호 통신가능하게 연결된다.

[0051] 인코더 제어기(505)의 제2 출력은 SEI(Supplemental Enhancement Information) 삽입기(530)의 제1 입력, 변환기 및 양자화기(525)의 제2 입력, 엔트로피 코더(545)의 제2 입력, 출력 버퍼(535)의 제2 입력, 및 SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set) 삽입기(540)의 입력과 신호 통신가능하게 연결된다.

[0052] SEI 삽입기(530)의 출력은 가산기(590)의 제2 비반전 입력과 신호 통신가능하게 연결된다.

[0053] 화상 유형 결정 모듈(515)의 제1 출력은 프레임 정렬 버퍼(510)의 제3 입력과 신호 통신가능하게 연결된다. 화상 유형 결정 모듈(515)의 제2 출력은 매크로블록 유형 결정 모듈(520)의 제2 입력과 신호 통신가능하게 연결된다.

[0054] SPS 및 PPS 삽입기(540)의 출력은 가산기(590)의 제3 비반전 입력과 신호 통신가능하게 연결된다.

[0055] 역양자화기 및 역변환기(550)의 출력은 가산기(519)의 제1 비반전 입력과 신호 통신가능하게 연결된다. 가산기(519)의 출력은 인트라 예측 모듈(560)의 제1 입력과 디블록킹 필터(565)의 제1 입력과 신호 통신가능하게 연결된다. 디블록킹 필터(565)의 출력은 참조 화상 버퍼(580)의 제1 입력과 신호 통신가능하게 연결된다. 참조 화상 버퍼(580)의 출력은 움직임 추정기(575)의 제2 입력과 움직임 보상기(570)의 제3 입력과 신호 통신가능하게 연결된다. 움직임 추정기(575)의 제1 출력은 움직임 보상기(570)의 제2 입력과 신호 통신가능하게 연결된다. 움직임 추정기(575)의 제2 출력은 엔트로피 코더(545)의 제3 입력과 신호 통신가능하게 연결된다.

[0056] 움직임 보상기(570)의 출력은 스위치(597)의 제1 입력과 신호 통신가능하게 연결된다. 인트라 예측 모듈(560)의 출력은 스위치(597)의 제2 입력과 신호 통신가능하게 연결된다. 매크로블록 유형 결정 모듈(520)의 출력은 스위치(597)의 제3 입력과 신호 통신가능하게 연결된다. 스위치(597)의 제3 입력은 스위치의 "데이터" 입력(제어 입력, 즉 제3 입력과 대조적으로)이 움직임 보상기(570)에 의해 제공되는지 또는 인트라 예측 모듈(560)에 의해 제공되는지를 결정한다. 스위치(597)의 출력은 가산기(519)의 제2 비반전 입력과 가산기(585)의 반전 입력과 신호 통신가능하게 연결된다.

[0057] 프레임 정렬 버퍼(510)의 제1 입력과 인코더 제어기(505)의 입력은 입력 화상을 수신하기 위해 인코더(500)의 입력으로 이용가능하다. 나아가, SEI 삽입기(530)의 제2 입력은 메타데이터를 수신하기 위해 인코더(500)의 입

력으로 이용가능하다. 출력 버퍼(535)의 출력은 비트스트림을 출력하기 위해 인코더(500)의 출력으로 이용가능하다.

[0058] 도 6을 참조하면, 본 발명의 원리가 적용될 수 있는 예시적인 비디오 디코더가 일반적으로 참조 부호 600으로 표시된다. 비디오 디코더(600)는 엔트로피 디코더(645)의 제1 입력과 신호 통신가능하게 연결된 출력을 구비하는 입력 버퍼(610)를 포함한다. 엔트로피 디코더(645)의 제1 출력은 역변환기 및 역양자화기(650)의 제1 입력과 신호 통신가능하게 연결된다. 역변환기 및 역양자화기(650)의 출력은 가산기(625)의 제2 비반전 입력과 신호 통신가능하게 연결된다. 가산기(625)의 출력은 디블록킹 필터(665)의 제2 입력과 인트라 예측 모듈(660)의 제1 입력과 신호 통신가능하게 연결된다. 디블록킹 필터(665)의 제2 출력은 참조 화상 버퍼(680)의 제1 입력과 신호 통신가능하게 연결된다. 참조 화상 버퍼(680)의 출력은 움직임 보상기(670)의 제2 입력과 신호 통신가능하게 연결된다.

[0059] 엔트로피 디코더(645)의 제2 출력은 움직임 보상기(670)의 제3 입력, 디블록킹 필터(665)의 제1 입력, 및 인트라 예측기(660)의 제3 입력과 신호 통신가능하게 연결된다. 엔트로피 디코더(645)의 제3 출력은 디코더 제어기(605)의 입력과 신호 통신가능하게 연결된다. 디코더 제어기(605)의 제1 출력은 엔트로피 디코더(645)의 제2 입력과 신호 통신가능하게 연결된다. 디코더 제어기(605)의 제2 출력은 역변환기 및 역양자화기(650)의 제2 입력과 신호 통신가능하게 연결된다. 디코더 제어기(605)의 제3 출력은 디블록킹 필터(665)의 제3 입력과 신호 통신가능하게 연결된다. 디코더 제어기(605)의 제4 출력은 인트라 예측 모듈(660)의 제2 입력, 움직임 보상기(670)의 제1 입력, 및 참조 화상 버퍼(680)의 제2 입력과 신호 통신가능하게 연결된다.

[0060] 움직임 보상기(670)의 출력은 스위치(697)의 제1 입력과 신호 통신가능하게 연결된다. 인트라 예측 모듈(660)의 출력은 스위치(697)의 제2 입력과 신호 통신가능하게 연결된다. 스위치(697)의 출력은 가산기(625)의 제1 비반전 입력과 신호 통신가능하게 연결된다.

[0061] 입력 버퍼(610)의 입력은 입력 비트 스트림을 수신하기 위해 디코더(600)의 입력으로 이용가능하다. 디블록킹 필터(665)의 제1 출력은 출력 화상을 출력하기 위해 디코더(600)의 출력으로 이용가능하다.

[0062] 전술된 바와 같이 본 발명의 원리는 비디오 인코딩 및 디코딩을 위한 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 방법 및 장치에 관한 것이다. 일 실시예에서는 더 작은 블록 레벨에서 잔류값을 업데이트하며, 이는 큰 변환 및 수반하는 양자화의 효율을 희생하며 더 우수한 예측을 제공할 수 있다. 코딩 성능을 개선하면서 잔류값 업데이트 구조의 장점을 제공하는 다른 실시예에서는, 특정 코딩 블록에 대한 최상의 구조(best scheme)를 선택하기 위해 적응적 스위칭 접근법을 또한 제안한다. 구체적으로, 인코딩될 현재 블록에 대한 템플릿 매칭에서 서로 다른 잔류값 업데이트 구조들을 선택할 것을 제안한다. 스위치 상태의 불명확성(uncertainty)이 최종 비트스트림에서 명시적으로 전송될 수 있으며 또는 예를 들어 디코딩된 화상 정보로부터 암시적으로 유도될 수 있다. 이러한 적응적 스위칭 구조의 목적은 서로 다른 잔류값 업데이트 방법을 사용하는 것에 의해 템플릿 매칭을 위한 예측과 변환 사이에 최상의 트레이드오프(tradeoff: 절충점)를 발견하여 코딩 효율을 개선시키는 것이다. 제안된 기술은 인트라 예측과 인터 예측 모두에 적용될 수 있다.

[0063] 따라서, 본 발명의 원리에 따라, 이제 더 작은 변환 사이즈를 사용하여 예측 데이터를 개선시키는 "방법 2"라고 언급하는 방법을 개시하고 기술한다. 4개의 타깃 블록의 예측 공정은 다음 단계를 포함한다:

[0064] (1) 인접한 픽셀들을 사용하는 것에 의해 (기본 블록 내에 있는 및/또는 그렇지 않으면 기본 블록에 대응하는) 타깃 블록에 대한 템플릿을 생성한다;

[0065] (2) 매치를 검색하고 이 타깃 블록에 대한 예측을 생성한다;

[0066] (3) 그 예측을 통해 타깃 블록을 업데이트한다;

[0067] (4) 타깃 블록의 잔류값을 변환하여 변환 계수를 얻고, 계수를 양자화하고, 양자화된 계수를 역변환한다;

[0068] (5) 예측 타깃 블록과 양자화된 잔류값 타깃 블록을 더하는 것에 의해 재구성된 타깃 블록을 생성한다;

[0069] (6) 그 재구성을 통해 타깃 블록을 업데이트한다;

[0070] (7) 타깃 블록 2,3 및 4에 대해 단계(1) 내지 단계(6)을 반복한다;

[0071] (8) 기본 블록을 종료한다.

[0072] 변환이 더 작은 블록에 기초하면, 인접한 서브 블록은 더 일찍 재구성될 수 있다. 따라서, 차후 예측된 타깃 블

록은 예측 데이터를 사용하는 대신 템플릿으로 재구성된 픽셀을 사용할 수 있으며 이는 예측 성능을 개선시킬 수 있다.

[0073] 적응적 잔류값 업데이트

제1 잔류값 업데이트 구조에서는 (예를 들어, 더 큰 변환을 사용하는 것에 의해) 더 효율적인 구조가 변환에 사용되지만, 예측은 더 악화될 수 있는 반면, 제2 잔류값 업데이트는 (제1 구조보다) 더 우수한 예측을 가질 수 있으나 큰 변환을 사용하는 것에 수반되는 효율을 희생시킬 수 있다. 본 발명의 원리에 따라, 본 발명에서는 코딩 효율을 개선시키기 위해 두 방법의 장점을 이용하는 것에 의해 변환 및 양자화의 효율과 예측 정밀도 사이에 우수한 트레이드오프를 달성하기 위하여 이를 2개의 구조를 적응적으로 잔류값 업데이트를 수행하는 프레임워크로 조합한다.

[0075] 상이한 타깃 사이즈와 상이한 이용가능한 변환 사이즈는 여러 잔류값 업데이트 구조를 초래할 수 있다. 본 발명의 원리에 따른 최적의 잔류값 업데이트 구조는 모든 장점을 이용하고 그 단점을 적응적으로 우회할 수 있다.

[0076] 본 발명의 원리에 따라 본 발명에서는 잔류값 업데이트 구조들을 통합하는 프레임워크를 형성한다. 이 프레임워크에서, 템플릿 매칭을 위해 많은 상이한 잔류값 업데이트 구조들이 있을 수 있다. 이들 구조들 중에서 불명확성이 최종 비트스트림에서 명시적으로 송신되거나 또는 예를 들어 이미 코딩된 화상에 기초하여 암시적으로 유도될 수 있다. 예시와 간략화를 위하여, 본 명세서에서는 선택하는데 이용가능한 2개의 잔류값 업데이트 구조들이 있다고 가정하고 있지만, 본 발명의 원리는 이 기술 분야 및 관련 분야에서 통상의 지식을 가진 자에 의해 2를 초과하는 잔류값 업데이트 구조를 포함하는 것으로 용이하게 확장될 수 있다는 것을 주지해야 할 것이다.

[0077] 실시예 1: 명시적인 스위칭 방법

[0078] 전술된 2개를 초과하는 잔류값 업데이트 구조들이 있을 때에는 이러한 구조들 사이에 불명확성이 해결될 필요가 있다. 이 실시예에서는, 예를 들어 어느 구조가 선택될 수 있는지를 나타내는 플래그 또는 표시자를 사용하는 것에 의한 명시적인 스위칭 방법을 제안한다. 인코더에서는 모든 잔류값 업데이트 구조들을 평가하고 예를 들어 율-왜곡(RD: rate-distortion) 비용을 고려하여 최상의 잔류값 업데이트 구조를 선택하여 플래그나 표시자를 대응하는 디코더에 송신할 수 있다. 디코더에서는 비트스트림으로부터 오는 플래그나 표시자를 판독하는 것에 의해 불명확성을 해결하며 디코딩할 때 대응하는 구조를 적용한다.

[0079] 도 7a 및 도 7b를 참조하면, 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 예시적인 방법이 일반적으로 참조 부호 700으로 표시된다. 방법(700)은 제어를 기능 블록(710)으로 전달하는 시작 블록(705)을 포함한다. 기능 블록(710)은 min_rdcost=max_value, best_scheme=default_value, k=0로 설정하는 것에 의해 템플릿 매칭 예측을 위한 초기화를 수행하며 제어를 루프 제한 블록(715)으로 전달한다. 루프 제한 블록(715)은 블록을 위한 모든 잔류값 업데이트 구조에 대한 루프를 시작하며 제어를 루프 제한 블록(720)으로 전달한다. 루프 제한 블록(720)은 모든 서브 블록에 대한 루프를 시작하며 제어를 기능 블록(725)으로 전달한다. 기능 블록(725)은 템플릿을 생성하며 제어를 기능 블록(730)으로 전달한다. 기능 블록(730)은 템플릿 매칭을 수행하며 제어를 기능 블록(735)으로 전달한다. 기능 블록(735)은 그 예측을 통해 타깃 블록을 업데이트하며 제어를 기능 블록(740)으로 전달한다. 기능 블록(740)은 현재 구조에 따라 현재 타깃 블록에 변환, 양자화 및 재구성(잔류값 업데이트)을 수행할지 여부를 결정하고 제어를 루프 제한 블록(745)으로 전달한다. 루프 제한 블록(745)은 모든 서브 블록에 대한 루프를 종료하며 제어를 기능 블록(750)으로 전달한다. 기능 블록(750)은 현재 구조에 따라 현재 블록에 변환, 양자화 및 재구성(잔류값 업데이트)을 수행할지 여부를 결정하며 제어를 기능 블록(755)으로 전달한다. 기능 블록(755)은 기능 블록(750)에 의해 얻어진 결정에 대응하는 율-왜곡(RD) 비용을 계산하며 제어를 결정 블록(760)으로 전달한다. 결정 블록(760)은 rdcost>= min_rdcost 인지를 결정한다. 그렇다면, 제어는 루프 제한 블록(768)으로 전달된다. 그렇지 않다면, 제어는 기능 블록(765)으로 전달된다. 기능 블록(765)은 min_rdcost 그리고 best_scheme = k로 설정하며 제어를 기능 블록(768)으로 전달한다. 기능 블록(768)은 k=k+1로 설정하고 제어를 루프 제한 블록(770)으로 전달한다. 루프 제한 블록(770)은 모든 잔류값 업데이트 구조에 대한 루프를 종료하며 제어를 기능 블록(775)으로 전달한다. 기능 블록(775)은 템플릿 매칭 예측의 출력으로 best_scheme의 예측을 선택하며 best_scheme으로 tmp_res_upd_ind를 설정하며 제어를 기능 블록(780)으로 전달한다. 기능 블록(780)은 모든 구문(tmp_res_upd_ind를 포함하는)과 포함된 계수 정보를 인코드하며 제어를 종료 블록(799)으로 전달한다.

[0080] 도 8을 참조하면, 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 예시적인 방법이 일반적으로 참조 부호 800으로 표시된다. 방법(800)은 제어를 기능 블록(810)으로 전달하는 시작 블록(805)을 포함

한다. 기능 블록(810)은 입력 비트스트림을 파싱(parse)하고, tmp_res_upd_ind로 best_scheme를 설정하며 제어를 루프 제한 블록(815)으로 전달한다. 루프 제한 블록(815)은 모든 서브 블록에 대해 루프를 시작하여 제어를 기능 블록(820)으로 전달한다. 기능 블록(820)은 템플릿을 생성하며 제어를 기능 블록(825)으로 전달한다. 기능 블록(825)은 템플릿 매칭을 수행하여 제어를 기능 블록(830)으로 전달한다. 기능 블록(830)은 그 예측을 통해 타깃 블록을 업데이트하며 제어를 기능 블록(835)으로 전달한다. 기능 블록(835)은 best_scheme에 따라 현재의 타깃 블록을 재구성(잔류값 업데이트)할지 여부를 결정하며 제어를 루프 제한 블록(840)으로 전달한다. 루프 제한 블록(840)은 모든 서브 블록에 대한 루프를 종료하며 제어를 기능 블록(845)으로 전달한다. 기능 블록(845)은 best_scheme에 따라 현재 블록을 재구성(잔류값 업데이트) 할지 여부를 결정하며 제어를 종료 블록(899)으로 전달한다.

[0081] 실시예 2: 암시적인 스위칭 방법

명시적인 스위칭 방법은 코딩 성능을 저하시킬 수 있는 추가적인 부가 정보를 위한 비용을 지불하는 것에 의해 인코더에서 최상의 구조(best scheme)를 항상 선택할 수 있다. 전술된 바와 같이, 추가적인 부가 정보를 전송하는 것을 회피하기 위해, 상이한 구조들 중에서 불명확성이 암시적으로 유도될 수 있다. 여기에서는 파티션 정보, 인접한 블록 코딩 정보(예를 들어, 변환 사이즈, 계수, 모드 및/또는 등), 국부 화상 컨텐츠 통계(예를 들어, 평균, 분산 및/또는 등) 등을 이용할 수 있다. 이러한 암시적인 스위칭 방법은 명시적인 스위칭 방법에 있는 추가적인 오버헤드를 회피할 수 있으며 그리하여 느리게 변하는 컨텐츠의 경우에 사용하는데 선호될 수 있다. 비디오 인코더에서 이러한 실시예를 포함하는 방법이 아래 도 9를 참조하여 기술되며, 비디오 디코더에서 이러한 실시예를 포함하는 방법이 아래 도 10을 참조하여 기술된다.

[0083] 실시예 3: 양자화 파라미터에 의존하는 스위칭 방법

잔류값이 특정 서브 블록에 더 빨리 업데이트될수록, (특정 서브-블록에 대해) 후속하는 서브 블록들이 더 우수한 예측을 얻을 수 있을 가능성이 더 많다. 나아가, 전체 블록의 예측이 더 우수해질 수 있다. 블록이 큰 양자화 파라미터(QP: quantization parameter) 값으로 코딩된 때에는 잔류값이 이러한 경우에 작은 양의 정보만을 제공하므로 더 우수한 예측이 사용하는데 이용가능한 것이 선호된다. 한편, 블록이 상대적으로 더 작은 양자화 파라미터 값으로 코딩되면, 잔류값은 매우 중요하며 상당한 양의 예측 에러를 보상할 수 있다. 따라서, 본 발명에서는 앞선 사항을 고려하여 양자화 파라미터에 의존하는 스위칭 방법을 제안한다. 예를 들어, 2개의 전술된 구조만을 도입했다고 가정하면, 양자화 파라미터 값이 큰 것일 때에는, 더 우수한 예측을 얻기 위해 방법 2를 사용할 수 있다. 그렇지 않으면 큰 변환을 사용하는 장점을 유지하기 위해 방법 1을 사용한다.

이하 도 9 및 도 10은 본 명세서에 기술된 두 실시예 2 및 3에 관한 것이다.

도 9를 참조하면, 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 다른 예시적인 방법이 일반적으로 참조 부호 900으로 표시된다. 방법(900)은 제어를 기능 블록(910)으로 전달하는 시작 블록(905)을 포함한다. 기능 블록(910)은 양자화 파라미터(QP), 파티션 정보, 인접한 블록 정보, 디코딩된 컨텐츠 통계 및/또는 등에 의해 best_scheme를 유도하며 제어를 루프 제한 블록(915)으로 전달한다. 루프 제한 블록(915)은 모든 서브 블록에 대해 루프를 시작하여 제어를 기능 블록(920)으로 전달한다. 기능 블록(920)은 템플릿을 생성하며 제어를 기능 블록(925)으로 전달한다. 기능 블록(925)은 템플릿 매칭을 수행하여 제어를 기능 블록(930)으로 전달한다. 기능 블록(930)은 예측을 통해 타깃 블록을 업데이트하며 제어를 기능 블록(935)으로 전달한다. 기능 블록(935)은 best_scheme에 따라 현재 타깃 블록에 변환, 양자화 및 재구성(잔류값 업데이트)을 수행할지 여부를 결정하며 제어를 루프 제한 블록(940)으로 전달한다. 루프 제한 블록(940)은 모든 서브 블록에 대해 루프를 종료하며 제어를 기능 블록(945)으로 전달한다. 기능 블록(945)은 현재 구조에 따라 현재 블록에 변환, 양자화, 및 재구성(잔류값 업데이트)을 수행할지 여부를 결정하며 제어를 기능 블록(950)으로 전달한다. 기능 블록(950)은 모든 구문과 포함된 계수 정보를 인코드하며 제어를 종료 블록(999)으로 전달한다.

도 10을 참조하면, 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 다른 예시적인 방법이 일반적으로 참조 부호 1000으로 표시된다. 방법(1000)은 제어를 기능 블록(1010)으로 전달하는 시작 블록(1005)을 포함한다. 기능 블록(1010)은 입력 비트스트림을 파싱하여 제어를 기능 블록(1015)으로 전달한다. 기능 블록(1015)은 양자화 파라미터(QP), 파티션 정보, 인접한 블록 정보, 디코딩된 컨텐츠 통계 및/또는 등에 의해 best_scheme를 유도하며 제어를 루프 제한 블록(1020)으로 전달한다. 루프 제한 블록(1020)은 모든 서브 블록에 대해 루프를 시작하여 제어를 기능 블록(1025)으로 전달한다. 기능 블록(1025)은 템플릿을 생성하여 제어를 기능 블록(1030)으로 전달한다. 기능 블록(1030)은 템플릿 매칭을 수행하여 제어를 기능 블록(1035)으로 전달한다. 기능 블록(1035)은 예측을 통해 타깃 블록을 업데이트하여 제어를 기능 블록(1040)으로 전달한다. 기능

블록(1040)은 best_scheme에 따라 현재 타깃 블록을 재구성(잔류값 업데이트)할지 여부를 결정하며 제어를 루프 제한 블록(1045)으로 전달한다. 루프 제한 블록(1045)은 모든 서브 블록에 대한 루프를 종료하며 제어를 기능 블록(1050)으로 전달한다. 기능 브록(1050)은 best_scheme에 따라 현재 블록을 재구성(잔류값 업데이트)할지 여부를 결정하며 제어를 종료 블록(1099)으로 전달한다.

[0088] 구문(syntax)

표 1은 본 발명의 원리의 실시예에 따라 실시예 1, 2, 및 3을 위한 예시적인 슬라이스 헤드 구문을 도시한다.

표 1

slice_header() {	Descriptor
...	
use_tmp_res_upd_flag	u(1)
if(use_tmp_res_upd_flag == 1){	
use_exp_tmp_res_upd_flag	u(1)
if (use_exp_tmp_res_upd_flag == 0){	
use_qp_tmp_res_upd_flag	u(1)
}	
}	
...	
}	
marco_block_header() {	
...	
if (use_exp_tmp_res_upd_flag == 1){	
tmp_res_upd_ind	u(v)
}	
....	
}	

[0091]

표 1의 구문 요소들 중 일부의 의미는 다음과 같다:

[0092]

1인 **use_tmp_res_upd_flag**는 템플릿 매칭에서 복수의 잔류값 업데이트 구조의 사용을 나타낸다; 0인 **use_tmp_res_upd_flag**는 단 하나의 잔류값 업데이트 구조의 사용을 나타낸다.

[0093]

1인 **use_exp_tmp_res_upd_flag**는 명시적인 스위칭 방법의 사용을 나타내고; 0인 **use_exp_tmp_res_upd_flag**는 암시적인 스위칭 방법의 사용을 나타낸다.

[0094]

1인 **use_qp_tmp_res_upd_flag**는 양자화 파라미터에 의존하는 스위칭 방법의 사용을 나타내고; 0인 **use_qp_tmp_res_upd_flag**는 양자화 파라미터에 의존하는 스위칭 방법을 사용하지 않는 것을 나타낸다.

[0095]

1인 **tmp_res_upd_ind**는 현재 블록에 의해 사용되는 대응하는 잔류값 업데이트 구조를 나타낸다.

[0096]

tmp_res_upd_ind는 현재 블록에 의해 사용되는 대응하는 잔류값 업데이트 구조를 나타낸다.

[0097]

실시예 4: 다층 잔류값 업데이트 방법

[0098]

이 실시예에서는 새로운 다층 잔류값 업데이트 방법이 제안된다. 구체적으로, 기본 블록은 여기에 제공된 2개의 방법만이 존재한다고 가정하면 2번 코딩될 수 있다. 따라서, 2개를 초과하는 잔류값 업데이트 방법을 포함하는 다른 실시예에서는 기본 블록이 2번을 초과하여 코딩될 수 있다는 것을 이해할 수 있을 것이다.

[0099]

그럼에도 불구하고, 기본 블록을 2번 코딩하는 것을 포함하는 전술된 예에서, 먼저 블록은 상대적으로 높은 양자화 파라미터 값으로 방법 2에 의해 코딩된다. 이후 기본 블록의 재구성된 버전을 얻을 수 있다. 제5 단계에서, 블록은 상대적으로 낮은 양자화 파라미터 값으로 방법 1에 의해 코딩된다. 그러나, 제5 단계에서 템플릿이 제1 단계로부터 출력된 재구성된 블록에 기초하여 생성될 수 있다. 마지막으로, 최종 비트스트림에서 두

단계로부터 나온 잔류값을 전송할 수 있다. 디코더에서는 이와 유사한 단계들이 수행된다. 일 구현예에서, 실시 예 4는 이하 단계를 포함한다:

[0100] (1) 상대적으로 높은 양자화 파라미터 값을 설정한다;

[0101] (2) 작은 변환 사이즈를 사용하여 방법 2로 전체 기본 블록을 코딩한다;

[0102] (3) 방법 2에 기초하여 재구성된 기본 블록을 얻는다;

[0103] (4) 상대적으로 낮은 양자화 파라미터를 설정한다;

[0104] (5) 템플릿이 단계(3)으로부터 얻어진 재구성된 기본 블록에 기초하여 생성될 수 있는 큰 변환을 사용하여 방법 1에 기초하여 기본 블록을 코딩한다;

[0105] (6) 단계(2)와 단계(5)로부터 잔류값 계수를 전송한다;

[0106] (7) 블록을 종료한다.

[0107] 이하 도 11a, 도 11b 및 도 12에서 단계(2), 단계(3) 및 단계(5)에 대한 언급은 실시 예 4에 관한 단계들의 바로 이전의 리스트에 속하는 것이다.

[0108] 도 11a 및 도 11b를 참조하면, 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 또 다른 예시적인 방법이 일반적으로 참조 부호 1100으로 표시된다. 방법(1100)은 제어를 기능 블록(1110)으로 전달하는 시작 블록(1105)을 포함한다. 기능 블록(1110)은 2개의 잔류값 업데이트 구조에 대한 디폴트 양자화 파라미터 값들, 즉 QP1 및 QP2에 의해 템플릿 매칭 예측을 위한 초기화를 수행하며 제어를 루프 제한 블록(1115)으로 전달한다. 루프 제한 블록(1115)은 모든 서브 블록에 대해 루프를 시작하며 제어를 기능 블록(1120)으로 전달한다. 기능 블록(1120)은 템플릿을 생성하며 제어를 기능 블록(1125)으로 전달한다. 기능 블록(1125)은 템플릿 매칭을 수행하며 제어를 기능 블록(1130)으로 전달한다. 기능 블록(1130)은 예측을 통해 타깃 블록을 업데이트하며 제어를 기능 블록(1135)으로 전달한다. 기능 블록(1135)은 QP1으로 현재 타깃 블록에 대한 변환, 양자화 및 재구성(잔류값 업데이트)을 수행하며 제어를 루프 제한 블록(1140)으로 전달한다. 루프 제한 블록(1140)은 모든 서브 블록에 대해 루프를 종료하며 제어를 기능 블록(1145)으로 전달한다. 기능 블록(1145)은 단계(2)에 대한 계수를 저장하며 제어를 루프 제한 블록(1150)으로 전달한다. 루프 제한 블록(1150)은 모든 서브 블록에 대해 루프를 시작하며 제어를 기능 블록(1155)으로 전달한다. 기능 블록(1155)은 단계(3)로부터 재구성된 블록이나 디코딩된 화상에 기초하여 템플릿을 생성하며 제어를 기능 블록(1160)으로 전달한다. 기능 블록(1160)은 템플릿 매칭을 수행하며 제어를 기능 블록(1165)으로 전달한다. 기능 블록(1165)은 예측을 통해 타깃 블록을 업데이트하며 제어를 루프 제한 블록(1170)으로 전달한다. 루프 제한 블록(1170)은 모든 서브 블록에 대해 루프를 종료하며 제어를 기능 블록(1175)으로 전달한다. 기능 블록(1175)은 QP2로 현재 블록에 대한 변환, 양자화 및 재구성(잔류값 업데이트)을 수행하며 제어를 기능 블록(1180)으로 전달한다. 기능 블록(1180)은 단계(5)에 대한 계수를 저장하며 제어를 기능 블록(1185)으로 전달한다. 기능 블록(1185)은 모든 구문, 단계(2) 및 단계(5)의 계수 등을 인코드하며 제어를 종료 블록(1199)으로 전달한다.

[0109] 도 12을 참조하면, 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 또 다른 예시적인 방법이 일반적으로 참조 부호 1200으로 표시된다. 방법(1200)은 제어를 기능 블록(1210)으로 전달하는 시작 블록(1205)을 포함한다. 기능 블록(1210)은 입력 비트스트림을 파싱하며 제어를 루프 제한 블록(1215)으로 전달한다. 루프 제한 블록(1215)은 모든 서브 블록에 대해 루프를 시작하며 제어를 기능 블록(1220)으로 전달한다. 기능 블록(1220)은 템플릿을 생성하며 제어를 기능 블록(1225)으로 전달한다. 기능 블록(1225)은 템플릿 매칭을 수행하며 제어를 기능 블록(1230)으로 전달한다. 기능 블록(1230)은 예측을 통해 타깃 블록을 업데이트하며 제어를 기능 블록(1235)으로 전달한다. 기능 블록(1235)은 단계(2)의 계수로 현재 타깃 블록에 재구성(잔류값 업데이트)을 수행하며 제어를 루프 제한 블록(1240)으로 전달한다. 루프 제한 블록(1240)은 모든 서브 블록에 대해 루프를 종료하며 제어를 기능 블록(1245)으로 전달한다. 기능 블록(1245)은 모든 서브 블록에 대해 루프를 시작하며 제어를 기능 블록(1250)으로 전달한다. 기능 블록(1250)은 단계(3)로부터 재구성된 블록이나 디코딩된 화상에 기초하여 템플릿을 생성하며 제어를 기능 블록(1255)으로 전달한다. 기능 블록(1255)은 템플릿 매칭을 수행하며 제어를 기능 블록(1260)으로 전달한다. 기능 블록(1260)은 예측을 통해 타깃 블록을 업데이트 하며 제어를 루프 제한 블록(1265)으로 전달한다. 루프 제한 블록(1265)은 모든 서브 블록에 대해 루프를 종료 하며 제어를 기능 블록(1270)으로 전달한다. 기능 블록(1270)은 단계(5)의 계수로 현재 블록에 재구성(잔류값 업데이트)을 수행하며 제어를 종료 블록(1299)으로 전달한다.

[0110] 구문

[0111] 표 2는 본 발명의 원리의 일 실시예에 따라 실시예 4를 위한 예시적인 슬라이스 헤더 구문을 도시한다.

[0112] 표 2

slice_header() {	Descriptor
...	
use_tmp_res_upd_flag	u(1)
...	
}	
marco_block_header() {	
...	
if (use_tmp_res_upd_flag == 1){	
use_multi_layer_tmp_res_upd_flag	u(1)
}	
...	
}	

[0113]

[0114] 표 2의 구문 요소들 중 일부의 의미는 다음과 같다:

[0115] 1인 **use_tmp_res_upd_flag**는 템플릿 매칭에서 복수의 잔류값 업데이트 구조의 사용을 나타낸다; 0인 **use_tmp_res_upd**는 단 하나의 잔류값 업데이트 구조의 사용을 나타낸다.

[0116] 1인 **use_multi_layer_tmp_res_upd_flag**는 다층 잔류값 업데이트 방법의 사용을 나타내고; 0인 **use_multi_layer_tmp_res_upd_flag**는 다층 잔류값 업데이트 방법을 사용하지 않는 것을 나타낸다.

[0117]

실시예 5: 다중 결합된 잔류값 업데이트 방법

[0118] 이 실시예에서, 새로운 다중 결합된 잔류값 업데이트 방법이 제안된다. 구체적으로, 기본 블록은 여기서 제공된 단 2개의 잔류값 업데이트 방법만이 존재한다고 가정하면 2번 코딩될 수 있다. 따라서, 2개를 초과하는 잔류값 업데이트 방법을 포함하는 다른 실시예에서는 기본 블록이 2번을 초과하여 코딩될 수 있다는 것을 이해할 수 있을 것이다.

[0119]

그럼에도 불구하고, 기본 블록을 2번 코딩하는 것을 포함하는 전술된 예에서, 먼저 블록은 방법 2로 코딩된다. 이후 기본 블록의 재구성된 버전을 얻을 수 있다. 이후 블록은 방법 1로 코딩된다. 마지막으로, 최종 재구성된 블록을 형성하기 위해 재구성된 블록을 결합한다. 2개의 방법에서 잔류값은 최종 비트스트림으로 전송될 수 있다. 디코더에서는 이와 유사한 단계들이 수행된다.

[0120]

도 13을 참조하면, 비디오 인코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 더 다른 예시적인 방법이 일반적으로 참조 부호 1300으로 표시된다. 방법(1300)은 제어를 기능 블록(1310)과 기능 블록(1315)으로 전달하는 시작 블록(1305)을 포함한다. 이 기능 블록(1310, 1315)은 2개의 독립적인 잔류값 업데이트 구조들, 즉 방법 1 및 방법 2를 각각 사용하는 것에 의해 현재 블록의 2개의 재구성된 버전을 얻기 위해 템플릿 매칭과 인코딩을 수행한다. 이후 기능 블록(1310, 1315)은 제어를 기능 블록(1320)으로 전달한다. 기능 블록(1320)은 기능 블록(1310, 1315)으로부터 얻어진 2개의 재구성된 버전을 결합하여 현재 블록의 최종 재구성된 블록을 형성하며 제어를 기능 블록(1325)으로 전달한다. 기능 블록(1325)은 두 방법으로부터 모든 구문과 잔류값을 인코드하며 제어를 종료 블록(1330)으로 전달한다.

[0121]

도 14를 참조하면, 비디오 디코더에서 템플릿 매칭 예측의 적응적 잔류값 업데이트를 위한 더 다른 예시적인 방법이 일반적으로 참조 부호 1400으로 표시된다. 방법(1400)은 제어를 기능 블록(1410)으로 전달하는 시작 블록(1405)을 포함한다. 기능 블록(1410)은 비트스트림을 파싱하며 제어를 기능 블록(1415)과 기능 블록(1420)으로 전달한다. 기능 블록(1415, 1420)은 2개의 독립적인 방법, 즉 방법 1 및 방법 2를 각각 사용하는 것에 의해 현재 블록의 2개의 재구성된 버전을 얻기 위해 템플릿 매칭 예측과 디코딩을 수행한다. 기능 블록(1415, 1420)은 제어를 기능 블록(1425)으로 전달한다. 기능 블록(1425)은 블록(1415, 1420)으로부터 얻어진 재구성된 블록을

결합하여 현재 블록의 최종 재구성된 블록을 형성하며 제어를 종료 블록(1430)으로 전달한다.

[0122] 구문

표 3은 본 발명의 원리의 일 실시예에 따라 실시예 5를 위한 예시적인 슬라이스 헤드 구문을 보여준다.

표 3

slice_header() {	Descriptor
...	
use_tmp_res_upd_flag	u(1)
...	
}	
macro_block_header() {	
...	
if (use_tmp_res_upd_flag == 1){	
use_comb_multi_tmp_res_upd_flag	u(1)
}	
....	
}	

[0125]

표 2의 구문 요소들 중 일부의 의미는 다음과 같다:

[0126]

1인 **use_tmp_res_upd_flag**는 템플릿 매칭에서 복수의 잔류값 업데이트 구조의 사용을 나타낸다; 0인 **use_tmp_res_upd_flag**는 단 하나의 잔류값 업데이트 구조의 사용을 나타낸다.

[0127]

1인 **use_comb_multi_tmp_res_upd_flag**는 다중 결합된 잔류값 업데이트 방법의 사용을 나타내고; 0인 **use_comb_multi_tmp_res_upd_flag**는 다중 결합된 잔류값 업데이트 방법을 사용하지 않는 것을 나타낸다.

[0128]

이제 일부가 전술된 본 발명의 많은 부수적인 장점/특징들 중 일부에 대한 설명이 제공된다. 예를 들어, 하나의 장점/특징은 템플릿 매칭 예측을 사용하여 화상에 있는 적어도 하나의 블록에 대한 화상 데이터를 인코딩하는 비디오 인코더를 구비하되, 여기서 상기 템플릿 매칭 예측은 블록을 인코딩하는데 사용하기 위한 복수의 잔류값 업데이트 구조들 중에서 선택하는 것을 포함하는 것인 장치이다.

[0129]

다른 장점/특징은 복수의 잔류값 업데이트 구조들 중 적어도 하나가 율-왜곡 비용에 기초하여 선택되는 것인 전술된 비디오 인코더를 구비하는 장치이다.

[0130]

또 다른 장점/특징은 복수의 잔류값 업데이트 구조들 각각이 블록을 복수의 서브 블록으로 분할하는 것을 포함하며, 상기 복수의 잔류값 업데이트 구조들 중 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하며, 복수의 업데이트 구조들 중 다른 하나는 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료한 후에 복수의 서브 블록들 중에서 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 것인 전술된 비디오 인코더를 구비하는 장치이다.

[0131]

더 다른 장점/특징은 복수의 잔류값 업데이트 구조들이 최종 비트스트림에 명시적으로 표시되는 것인 전술된 비디오 인코더를 구비하는 장치이다.

[0132]

나아가, 다른 장점/특징은 복수의 잔류값 업데이트 구조들 중 적어도 하나가 화상과 동일한 비디오 시퀀스에 대응하는 디코딩된 화상, 블록 중 적어도 하나의 블록의 양자화 파라미터 값, 블록 내 적어도 하나의 서브 블록, 및 블록이나 적어도 하나의 서브 블록에 대하여 적어도 하나의 인접한 블록, 화상 컨텐츠 통계, 파티션 정보, 및 적어도 하나의 인접한 블록의 코딩 정보 중 적어도 하나에 기초하여 선택되는 것인 전술된 비디오 인코더를 구비하는 장치이다.

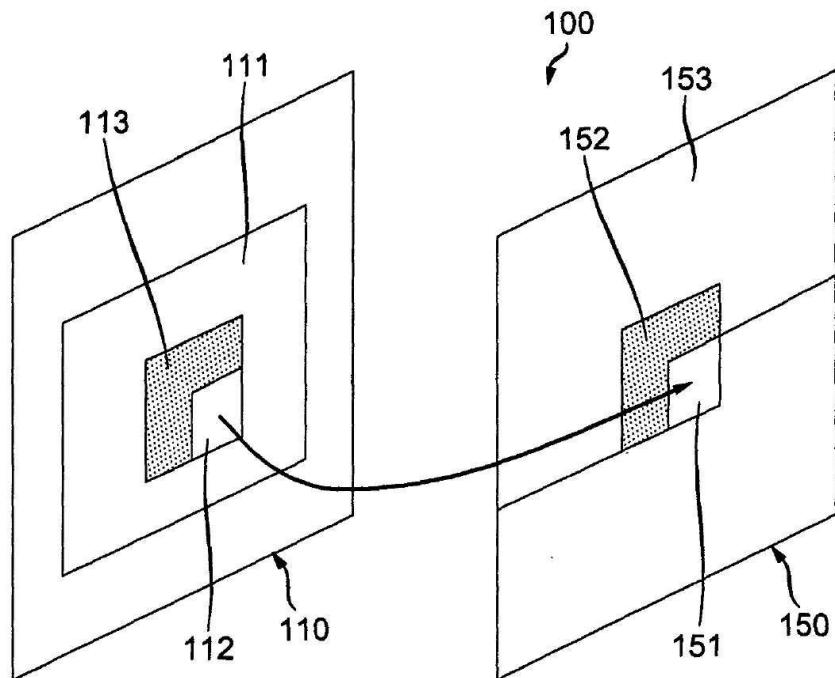
[0133]

더욱이, 다른 장점/특징은 복수의 잔류값 업데이트 구조들 중 2개 이상이 블록을 예측하기 위해 결합될 수 있는 것인 전술된 비디오 인코더를 구비하는 장치이다.

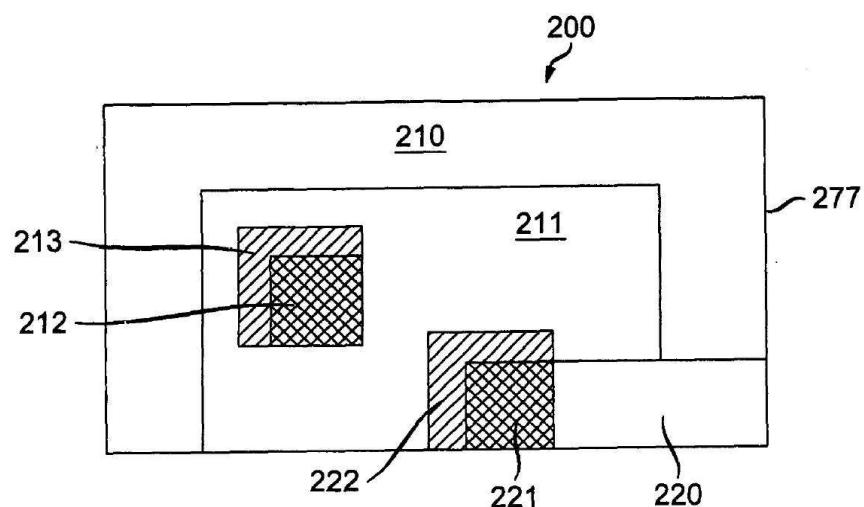
- [0135] 또한, 다른 장점/특징은 복수의 잔류값 업데이트 구조들 중 2개 이상이 전술된 바와 같이 블록을 예측하기 위해 결합될 수 있고, 복수의 잔류값 업데이트 구조들 각각은 서로에 대하여 독립적이고, 블록의 최종 예측은 복수의 잔류값 업데이트 구조들 중 적어도 2개의 구조의 출력의 결합인 것인 비디오 인코더를 구비하는 장치이다.
- [0136] 추가적으로, 다른 장점/특징은 복수의 잔류값 업데이트 구조들 중 2개 이상이 전술된 바와 같이 블록을 예측하기 위해 결합될 수 있고, 복수의 잔류값 업데이트 구조들이 서로 종속적이고, 복수의 잔류값 업데이트 구조들 중 특정한 것의 하나의 출력은 복수의 잔류값 업데이트 구조들 중 적어도 하나의 다른 특정한 것의 적어도 하나의 다른 출력에 종속하는 것인 비디오 인코더를 구비하는 장치이다.
- [0137] 나아가, 다른 장점/특징은 템플릿 매칭 예측을 사용하여 화상에 있는 블록을 인코딩하는 비디오 인코더를 구비하되, 여기서 템플릿 매칭 예측은 블록을 복수의 서브 블록으로 분할하는 것과, 특정한 서브-블록 블록에 대한 예측을 검색한 후에 그리고 템플릿 매칭 예측에 대하여 블록의 처리를 완료하기 전에 복수의 서브 블록들 중에서 특정한 서브 블록의 잔류값에 변환을 적용하는 것을 포함하는 것인 장치이다.
- [0138] 본 발명의 원리의 이들 및 다른 특징 및 장점은 본 명세서에 있는 교시 내용에 기초하여 이 기술 분야에 통상의 지식을 가진 자라면 용이하게 인식할 수 있을 것이다. 본 발명의 교시 내용이 하드웨어, 소프트웨어, 펌웨어, 특정 목적 프로세서 또는 이들의 조합의 여러 형태로 구현될 수 있다는 것을 이해할 수 있을 것이다.
- [0139] 가장 바람직하게는 본 발명의 원리의 교시 내용은 하드웨어와 소프트웨어의 조합으로 구현된다. 나아가, 소프트웨어는 프로그램 저장 장치에 유형적으로 구현되는 애플리케이션 프로그램으로 구현될 수 있다. 애플리케이션 프로그램은 임의의 적절한 이키텍처를 포함하는 기계에 업로딩되고 이 기계에 의해 실행될 수 있다. 바람직하게는 이 기계는 하나 이상의 중앙 처리 장치("CPU"), 램덤 액세스 메모리("RAM") 및 입력/출력("I/O") 인터페이스와 같은 하드웨어를 구비하는 컴퓨터 플랫폼에 구현된다. 컴퓨터 플랫폼은 운영 시스템 및 마이크로명령 코드를 더 포함할 수 있다. 본 명세서에 기술된 여러 공정과 기능은 CPU에 의해 실행될 수 있는 마이크로명령 코드의 일부나 애플리케이션 프로그램의 일부 또는 이들의 임의의 조합일 수 있다. 나아가, 여러 다른 주변 장치들이 추가적인 데이터 저장 장치와 프린팅 장치와 같은 컴퓨터 플랫폼에 연결될 수 있다.
- [0140] 첨부 도면에 도시된 구성 시스템 요소 및 방법의 일부는 바람직하게는 소프트웨어로 구현될 수 있으므로, 시스템 요소나 공정 기능 블록들 사이의 실제 연결은 본 발명의 원리가 프로그래밍되는 방식에 따라 상이할 수 있다는 것을 더 이해할 수 있을 것이다. 본 명세서에 있는 교시 내용에 따라 이 기술 분야에 통상의 지식을 가진 자라면 본 발명의 원리의 이들 및 이와 유사한 구현이나 구성을 구상할 수 있을 것이다.
- [0141] 첨부 도면을 참조하여 본 명세서에서 예시적인 실시예들이 기술되었으나, 본 발명의 원리는 이들 정확한 실시예로 제한되는 것은 아니고 본 발명의 원리의 범위나 사상을 벗어남이 없이 이 기술 분야에 통상의 지식을 가진 자라면 여러 변경과 변형이 이루어질 수 있다는 것을 이해할 수 있을 것이다. 그러므로, 모든 이러한 변경과 변형은 첨부된 청구범위에 개시된 본 발명의 원리의 범위 내에 있는 것으로 의도된다.

도면

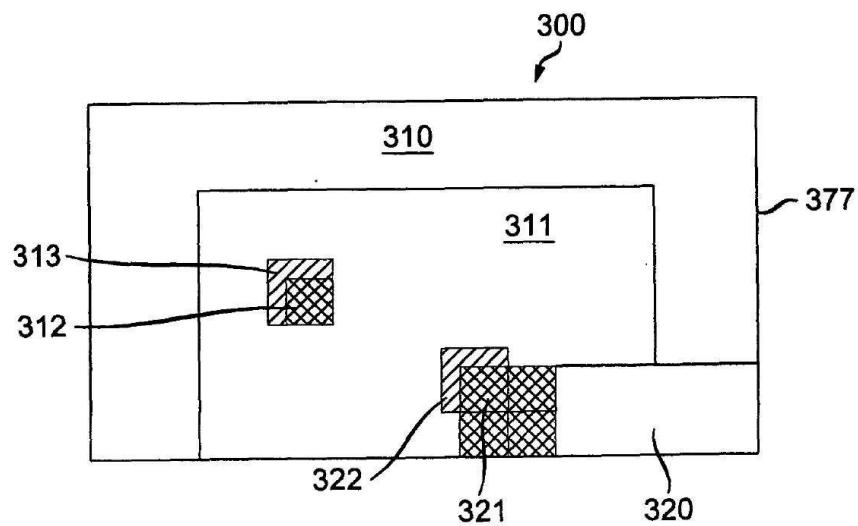
도면1



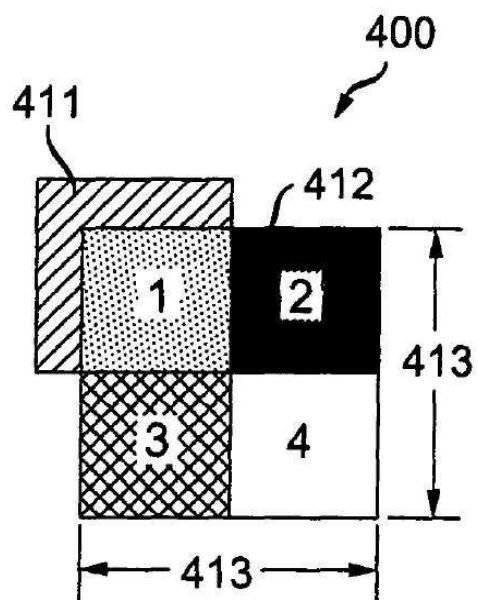
도면2



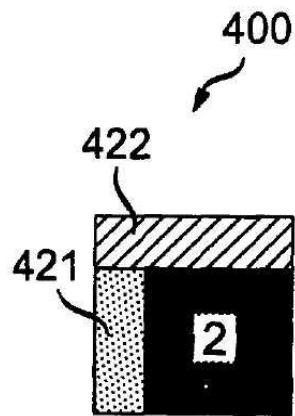
도면3



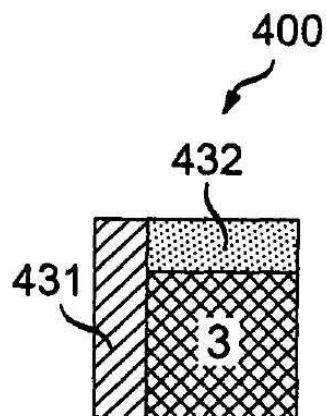
도면4a



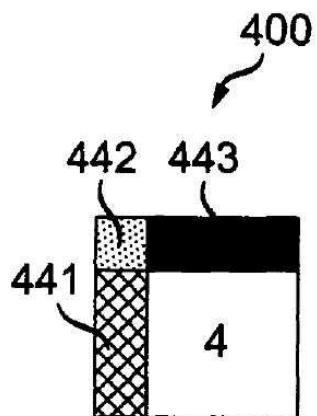
도면4b



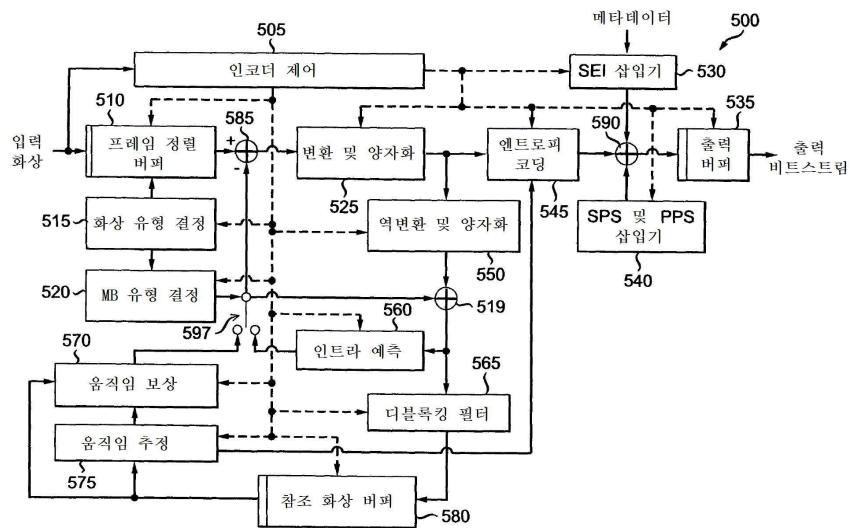
도면4c



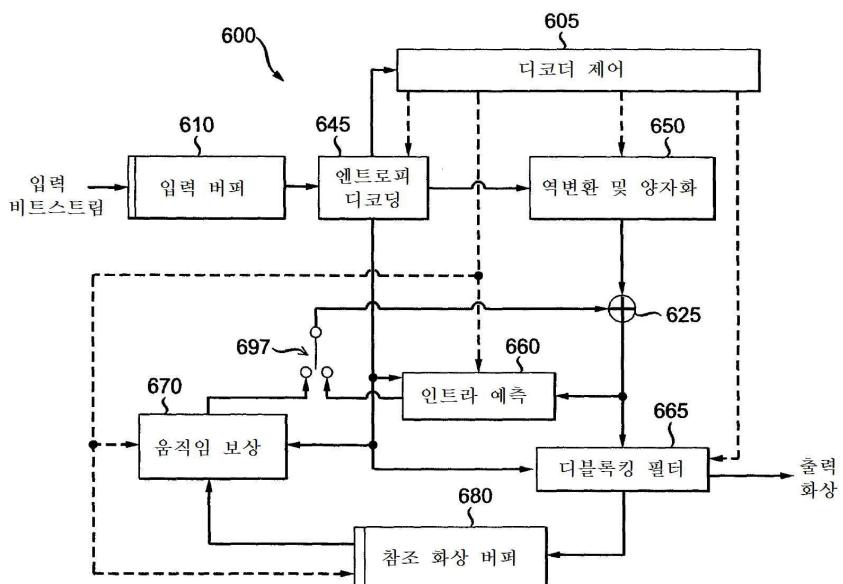
도면4d



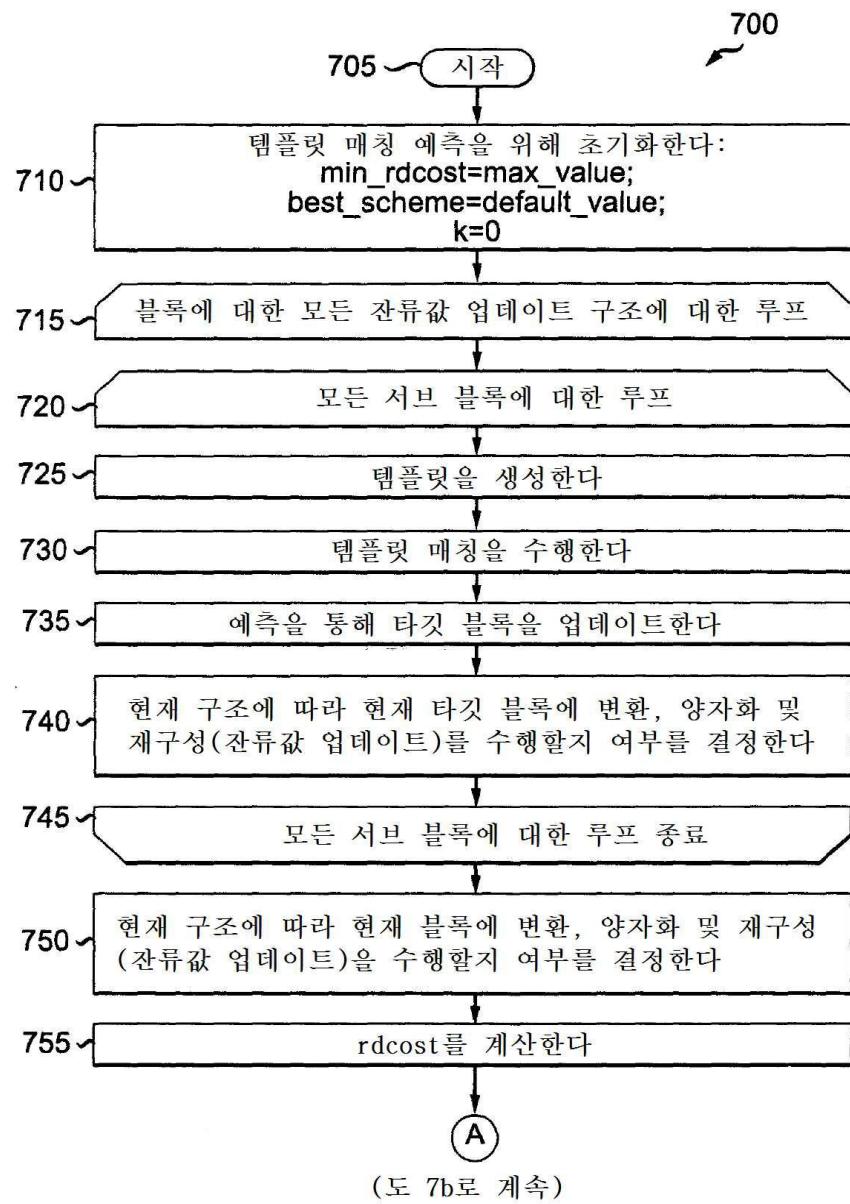
도면5



도면6



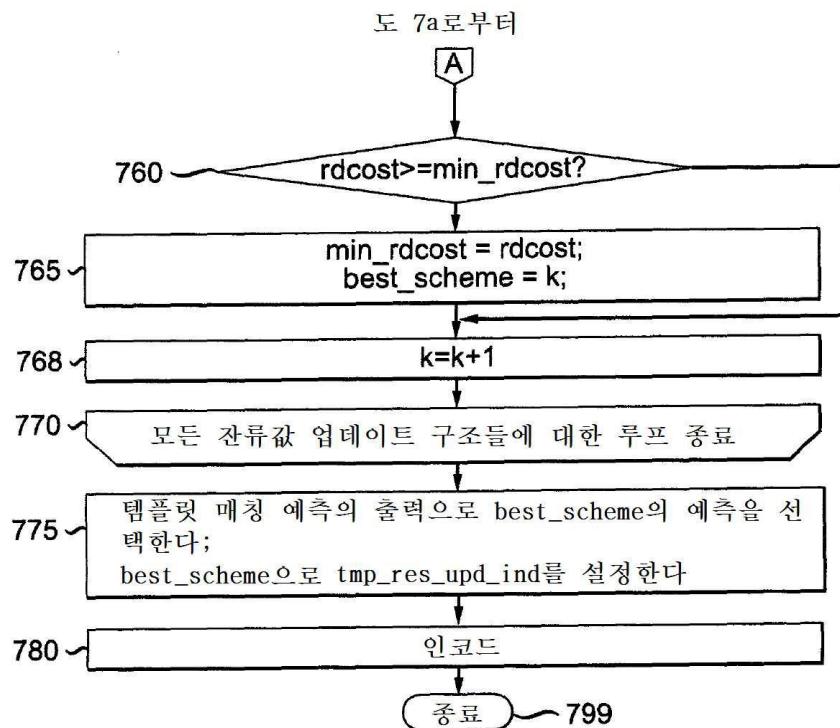
도면7a



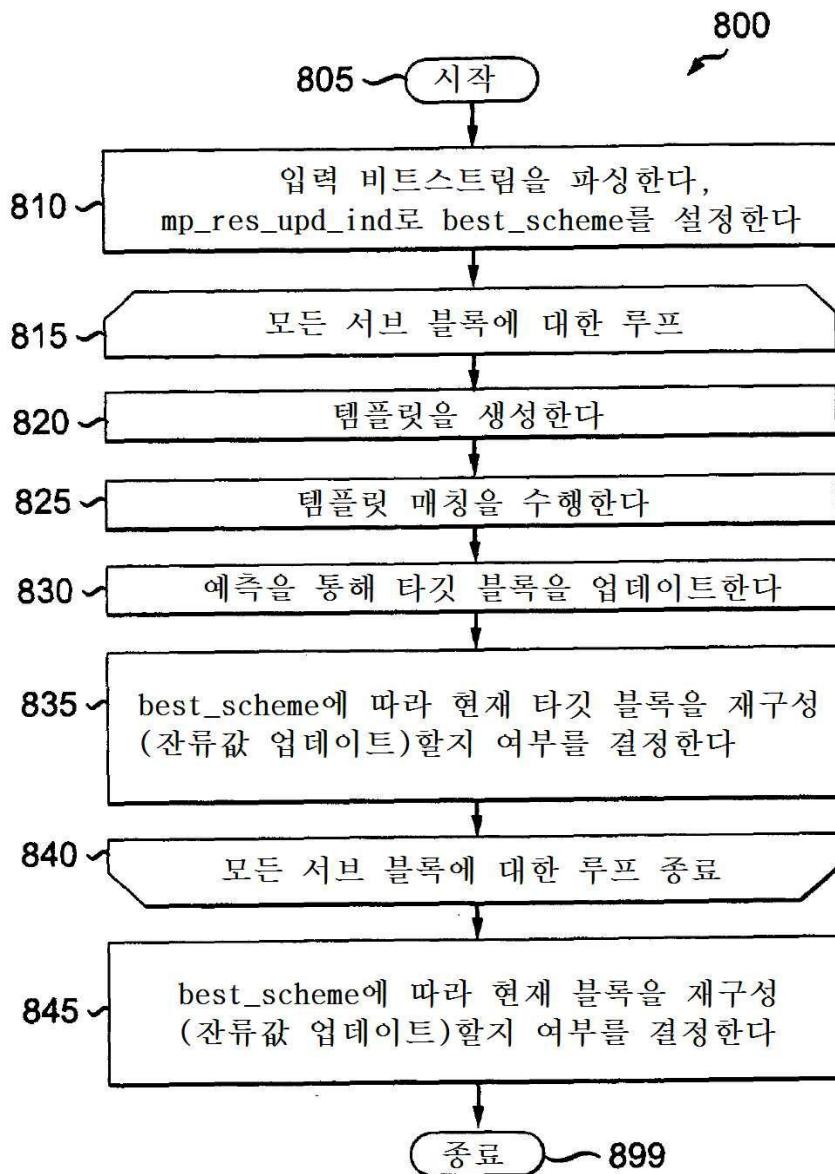
도면7b

700

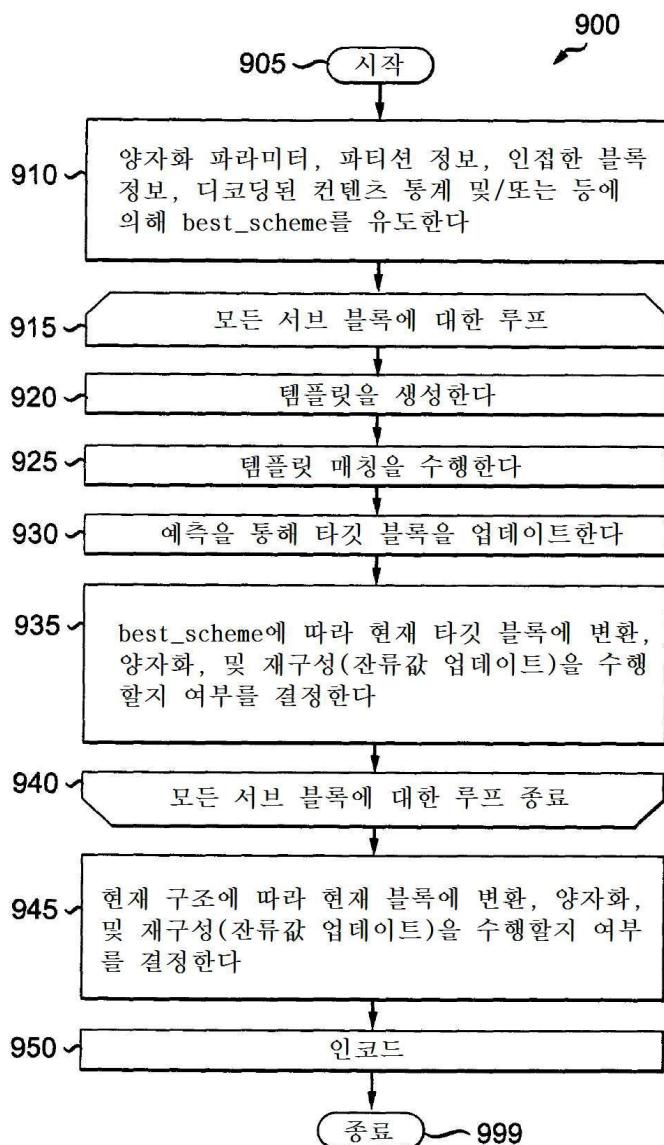
(도 7a에서 계속)



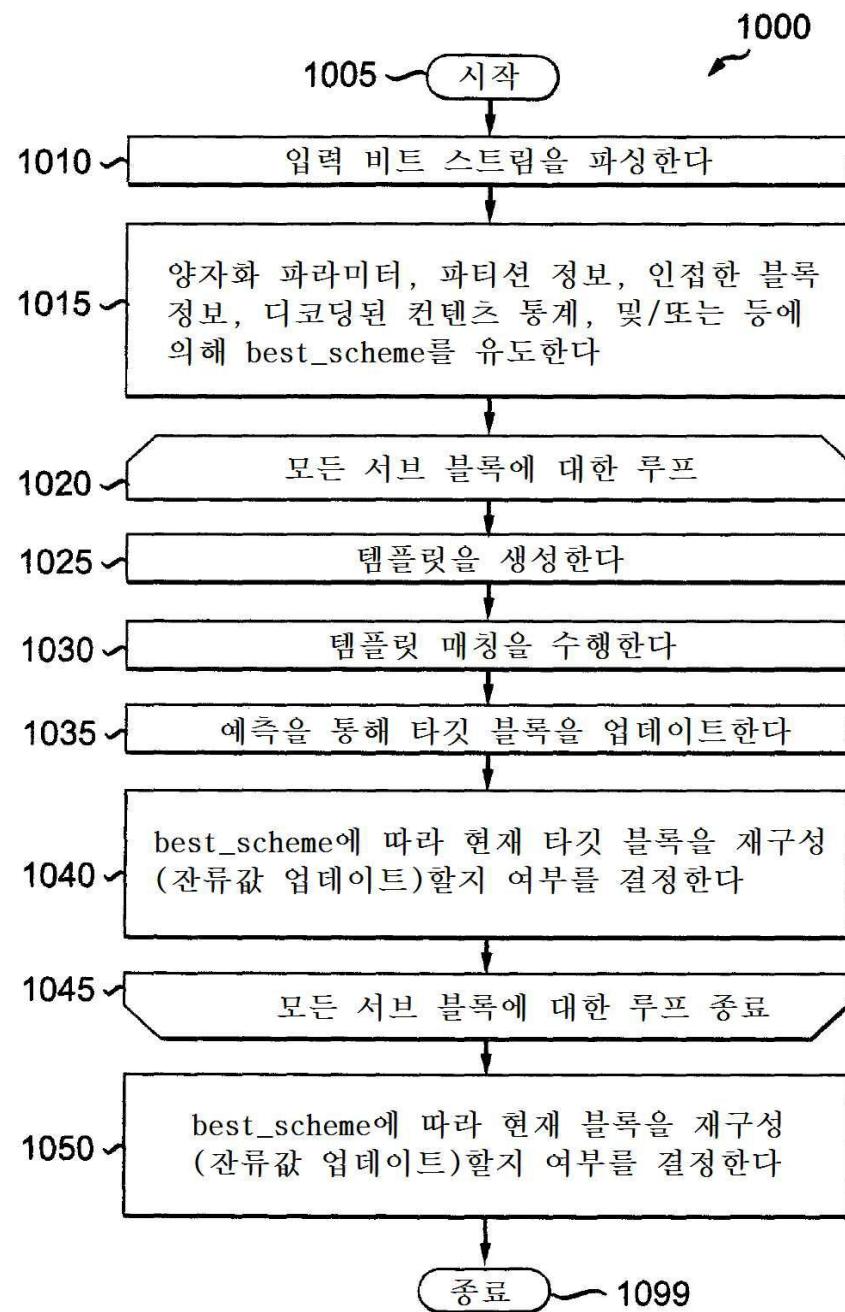
도면8



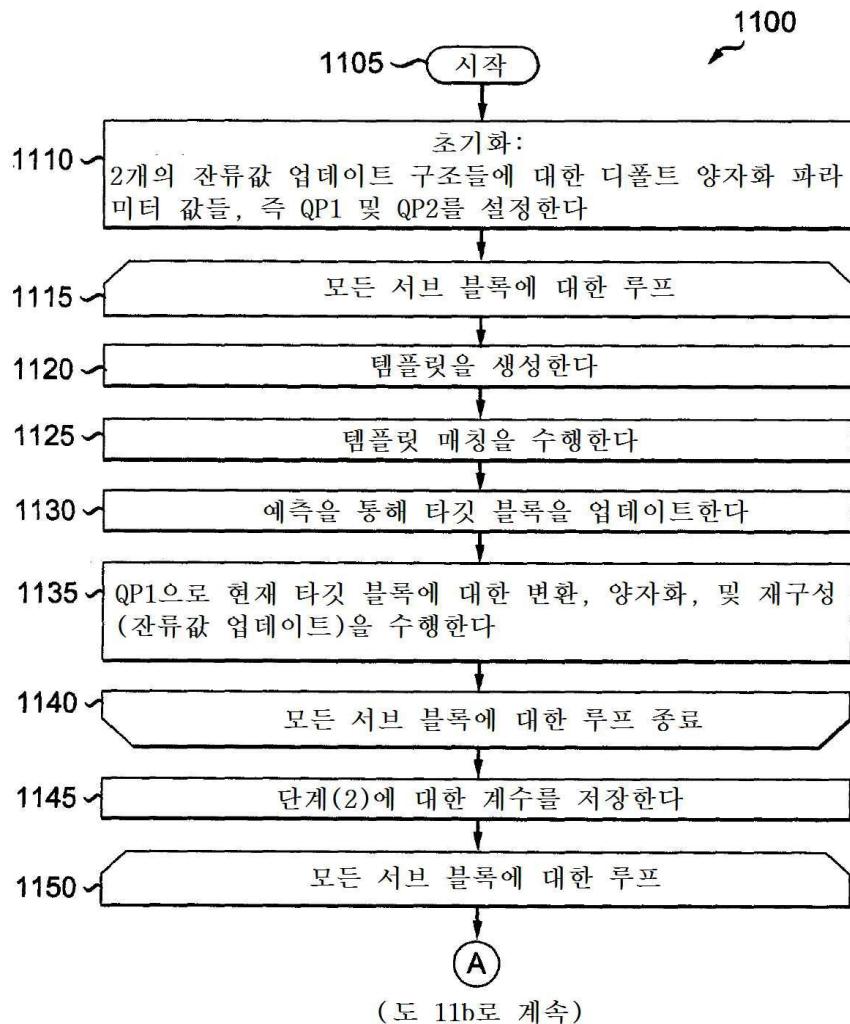
도면9



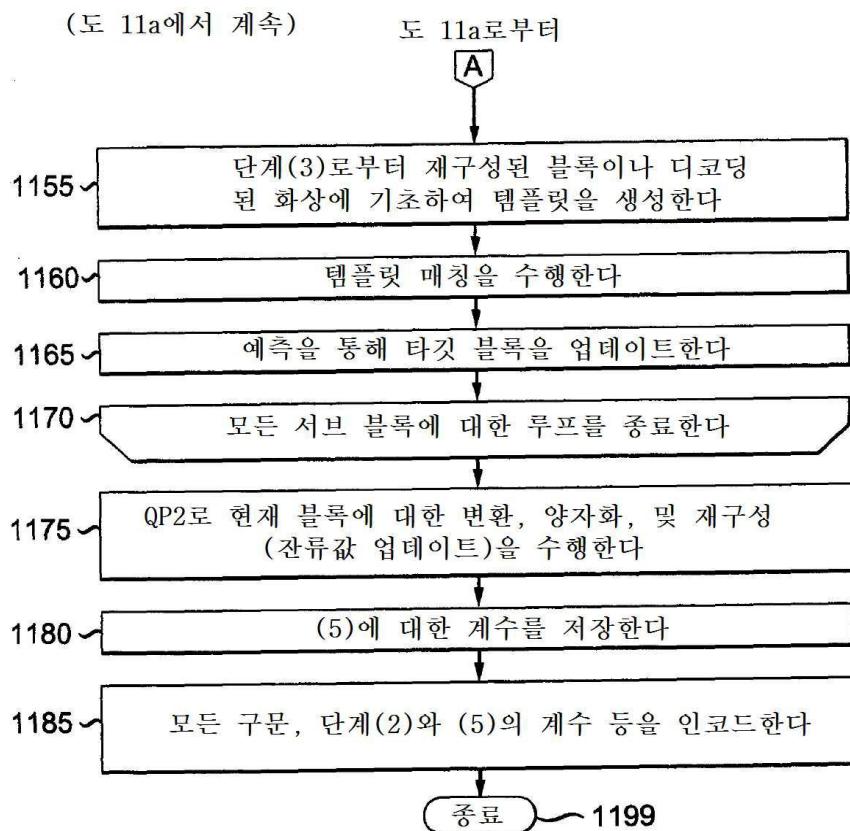
도면10



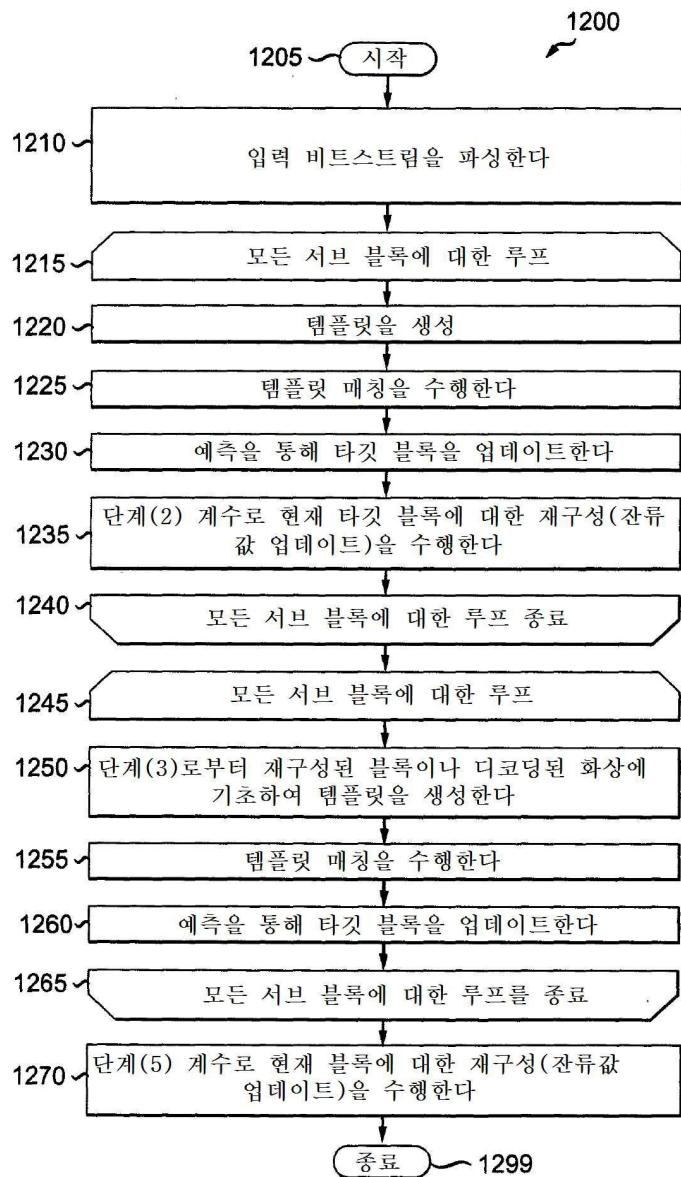
도면11a



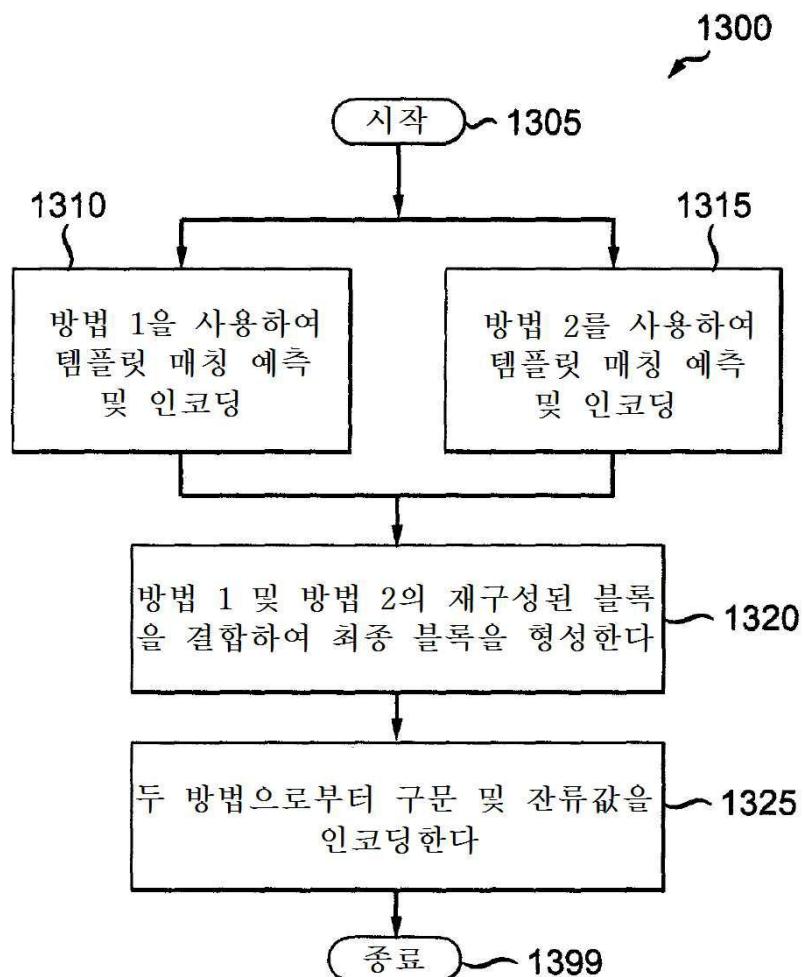
도면11b



도면12



도면13



도면14

