



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2015년06월05일  
(11) 등록번호 10-1526349  
(24) 등록일자 2015년06월01일

- (51) 국제특허분류(Int. Cl.)  
H04N 19/117 (2014.01) H04N 19/147 (2014.01)  
H04N 19/156 (2014.01) H04N 19/80 (2014.01)
- (21) 출원번호 10-2013-7001815
- (22) 출원일자(국제) 2011년09월30일  
심사청구일자 2013년01월23일
- (85) 번역문제출일자 2013년01월23일
- (65) 공개번호 10-2013-0030294
- (43) 공개일자 2013년03월26일
- (86) 국제출원번호 PCT/CN2011/080408
- (87) 국제공개번호 WO 2012/045269  
국제공개일자 2012년04월12일
- (30) 우선권주장  
13/093,068 2011년04월25일 미국(US)  
(뒷면에 계속)
- (56) 선행기술조사문헌  
W02009126915 A1\*  
JP2011035619 A  
\*는 심사관에 의하여 인용된 문헌

- (73) 특허권자  
미디어텍 인크.  
중화민국 타이완, 신추, 신추 사이언스 파크, 더  
싱 1 로드, 넘버 1
- (72) 발명자  
첸 칭-예  
대만 타이페이 씨티 난강 디스트릭트 종샤오 이.  
로드 섹션 5 넘버 809 4층  
후 치-밍  
대만 흐신추 씨티 시양산 디스트릭트 종후아 로드  
섹션 6 라인 647 엘리 31 넘버 115  
(뒷면에 계속)
- (74) 대리인  
유미특허법인

전체 청구항 수 : 총 15 항

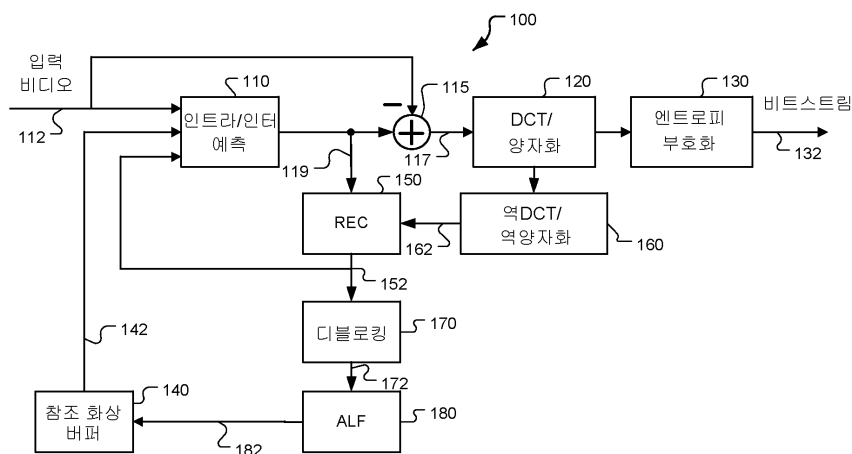
심사관 : 김영태

(54) 발명의 명칭 **영역 기반의 적응 루프 필터링의 방법 및 장치**

(57) 요약

블록 기반 움직임 보상 시스템에서, 손실성 연산으로 인하여 부호화 잡음이 발생할 수 있다. 적응 루프 필터(ALF)는 화질을 향상시키기 위해 이용되는 기술이다. 픽셀을 분류하기 위해 변형 라플라시안 합 측정(SLM : Sum-modified Laplacian Measure)을 이용하고 SLM에 따라 각각의 픽셀에 각각의 ALF 필터를 적용하는 픽셀 적응  
(뒷면에 계속)

대표도 - 도1



(PA) ALF 방법이 개발되었다. PA ALF가 종래의 단일 필터(SF) 기반 ALF 이상으로 더 우수한 성능을 달성하지만, PA ALF는 더 높은 복잡도를 야기하고, 높은 복잡도 및 픽셀마다의(per-pixel) 필터 스위칭으로 인해 더 많은 파워를 소비한다. 이에 따라, 영역별로(on a area by area basis) 적응 필터 선택을 허용하고 픽셀마다의 필터 스위칭을 요구하지 않는 영역 기반 ALF 방식이 개시된다. 본 발명에 따른 일실시예에서, 화상은 M×N 영역으로 분할된다. 영역 기반 ALF는 또한 성능을 향상시키기 위해 M×N 영역에 대해서 영역 병합을 허용한다. 본 발명의 또 다른 특징에서, 최적의 ALF 설계는 또한 레이트-왜곡-복잡도 최적화(RDCO)를 이용함으로써 시스템 복잡도를 고려한다. RDCO 기술은 또한 샘플 적응 오프셋(SAO) 설계에도 적용된다.

(72) 발명자

**차이 시아-양**

대만 뉴 타이페이 시티 중해 디스트릭트 민행 스트리트 라인 35 엘리 2 넘버 10 2층

**황 유-웬**

대만 타이페이 시티 종산 디스트릭트 룽지앙 로드 라인 298 넘버 23 8층

**레이 쇼우-민**

대만 호신쥬 카운티 쥬베이 시티 리우지아 6쓰 스트리트 넘버 32 6층

(30) 우선권주장

13/158,427 2011년06월12일 미국(US)

13/177,343 2011년07월06일 미국(US)

61/390,068 2010년10월05일 미국(US)

61/421,729 2010년12월10일 미국(US)

61/429,313 2011년01월03일 미국(US)

61/454,829 2011년03월21일 미국(US)

## 명세서

### 청구범위

#### 청구항 1

부호화된 비디오를 적응 루프 필터(ALF : adaptive loop filter)를 이용하여 처리하는 방법에 있어서,  
부호화된 비디오 데이터에 대응하는 재구성된 비디오 데이터(reconstructed video data)를 처리 유닛으로부터 수신하는 단계;

필터링된 비디오 데이터를 생성하기 위해 상기 재구성된 비디오 데이터에 적응 루프 필터링을 적용하는 단계로서, 상기 적응 루프 필터링의 적용은 영역 적응 플래그(region adaptation flag)에 따라 영역 기반 ALF(region based ALF) 또는 비영역 기반 ALF(non-region based ALF)를 선택하여 이루어지는, 단계; 및

상기 필터링된 비디오 데이터를 제공하는 단계

를 포함하는 것을 특징으로 하는 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

#### 청구항 2

제1항에 있어서,

상기 영역 기반 ALF가 선택되면, 상기 재구성된 비디오 데이터가  $M \times N$  영역으로 분할되며, 여기서 M과 N은 정수인, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

#### 청구항 3

제2항에 있어서,

각각의  $M \times N$  영역은 스캐닝 패턴에 따라 그룹 인덱스가 할당되며, 제1 그룹 인덱스  $i$ 를 갖는 제1 영역이 제2 그룹 인덱스  $i-1$ 을 갖는 제2 영역과 병합된다는 것을 나타내기 위해 1-D 병합 선택스(1-D merge syntax)가 이용되는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

#### 청구항 4

제3항에 있어서,

상기 스캐닝 패턴은 힐버트 곡선(Hilbert curve)의 변형, 수평 스네이크 스캔(horizontal snake scan), 수직 스네이크 스캔(vertical snake scan), 지그-재그 스캔(zig-zag scan), 나선형 스캔(spiral scan), 쿼드-트리 스캔(quad-tree scan), 및 래스터 스캔(raster scan)으로 이루어진 군에서 선택되는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

#### 청구항 5

제2항에 있어서,

상기  $M \times N$  영역의 제1 영역이 상기 제1 영역의 하나 이상의 인접 영역과 병합되는 것을 허용하기 위해 2-D 병합 선택스가 이용되며, 상기 하나 이상의 인접 영역은 상기  $M \times N$  영역에 속하는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

#### 청구항 6

제5항에 있어서,

상기 2-D 병합 선택스는 상기 제1 영역이 상기 하나 이상의 인접 영역과 병합되는지를 나타내기 위해 제1 플래그를 포함하며, 상기 2-D 병합 선택스는 상기 제1 영역이 상기 제1 플래그에 의해 나타낸 바와 같이 상기 하나 이상의 인접 영역과 병합된다면 병합 후보 플래그를 더 포함하는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

**청구항 7**

제6항에 있어서,

상기 병합 후보 플래그의 표시는 상기 하나 이상의 인접 영역의 이용 가능성에 맞추어지는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

**청구항 8**

제2항에 있어서,

영역 병합이 허용되는지의 여부를 나타내기 위해 병합 플래그가 이용되는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

**청구항 9**

제2항에 있어서,

상기  $M \times N$  영역의 어떠한 2개의 영역 간의 각각의 영역 경계가 가장 큰 부호화 단위(LCU : largest coding unit)의 경계 또는 LCU의 경계들인, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법.

**청구항 10**

부호화된 비디오를 적응 루프 필터(ALF)를 이용하여 처리하는 장치에 있어서,

부호화된 비디오 데이터에 대응하는 재구성된 비디오 데이터를 처리 유닛으로부터 수신하는 수단;

필터링된 비디오 데이터를 생성하기 위해 상기 재구성된 비디오 데이터에 적응 루프 필터링을 적용하는 수단으로서, 상기 적응 루프 필터링의 적용은 영역 적응 플래그(region adaptation flag)에 따라 영역 기반 ALF(region based ALF) 또는 비영역 기반 ALF(non-region based ALF)를 선택하여 이루어지는, 수단; 및

상기 필터링된 비디오 데이터를 제공하는 수단

을 포함하는 것을 특징으로 하는 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 장치.

**청구항 11**

제10항에 있어서,

상기 영역 기반 ALF가 선택되면, 상기 재구성된 비디오 데이터가  $M \times N$  영역으로 분할되며, 여기서 M과 N은 정수인, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 장치.

**청구항 12**

제11항에 있어서,

각각의  $M \times N$  영역은 스캐닝 패턴에 따라 그룹 인덱스가 할당되며, 제1 그룹 인덱스  $i$ 를 갖는 제1 영역이 제2 그룹 인덱스  $i-1$ 을 갖는 제2 영역과 병합된다는 것을 나타내기 위해 1-D 병합 신택스(1-D merge syntax)가 이용되는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 장치.

**청구항 13**

제11항에 있어서,

상기  $M \times N$  영역의 제1 영역이 상기 제1 영역의 하나 이상의 인접 영역과 병합되는 것을 허용하기 위해 2-D 병합 신택스가 이용되며, 상기 하나 이상의 인접 영역은 상기  $M \times N$  영역에 속하는, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 장치.

**청구항 14**

제11항에 있어서,

영역 병합이 허용되는지의 여부를 나타내기 위해 병합 플래그가 이용되는, 부호화된 비디오를 적응 루프 필터를

이용하여 처리하는 장치.

**청구항 15**

제11항에 있어서,

상기  $M \times N$  영역의 어떠한 2개의 영역 간의 각각의 영역 경계가 가장 큰 부호화 단위(LCU : largest coding unit)의 경계 또는 LCU의 경계들인, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 장치.

**청구항 16**

삭제

**청구항 17**

삭제

**청구항 18**

삭제

**청구항 19**

삭제

**청구항 20**

삭제

**청구항 21**

삭제

**청구항 22**

삭제

**청구항 23**

삭제

**청구항 24**

삭제

**청구항 25**

삭제

**청구항 26**

삭제

**청구항 27**

삭제

**청구항 28**

삭제

**청구항 29**

삭제

청구항 30

삭제

**발명의 설명**

**기술 분야**

[0001] 관련 출원의 상호 참조

[0002] 본 발명은 "Improved In-Loop Filter"를 발명의 명칭으로 하여 2010년 10월 5일자로 출원된 미국 가특허 출원 번호 61/390,068호, "Rate-Distortion-Complexity Optimization for Adaptive Loop Filter"를 발명의 명칭으로 하여 2010년 12월 10일자로 출원된 미국 가특허 출원 번호 61/421,729호, "Method and Apparatus of Region-Based Adaptive Loop Filtering"를 발명의 명칭으로 하여 2011년 7월 6일자로 출원된 미국 특허 출원 번호 13/177,343호, "Region-Based ALF"를 발명의 명칭으로 하여 2011년 3월 21일자로 출원된 미국 가특허 출원 번호 61/454,829호, "Method and Apparatus of Adaptive Loop Filtering"를 발명의 명칭으로 하여 2011년 4월 25일자로 출원된 미국 특허 출원 번호 13/093,068호, "Apparatus and Method of Sample Adaptive Offset for Video Coding"을 발명의 명칭으로 하여 2011년 6월 12일자로 출원된 미국 특허 출원 번호 13/158,427호, 및 "Media Tek's Adaptive Loop Filter(MTK\_ALF)"를 발명의 명칭으로 하여 2011년 1월 3일자로 출원된 미국 가특허 출원 번호 61/429,313호를 우선권으로 주장한다. 상기한 미국 특허 출원 및 미국 가특허 출원은 그 전체 내용이 원용에 의해 본 명세서에 통합되어 있다.

[0003] 발명의 분야

[0004] 본 발명은 비디오 부호화(video coding)에 관한 것으로, 보다 구체적으로는 적응 루프 필터링에 관한 것이다.

**배경 기술**

[0005] MPEG-1/2/4 및 H.261/H.263/H.264/AVC와 같은 다양한 부호화 표준에서 움직임 보상 인터 프레임 부호화(Motion compensated inter-frame coding)가 폭 넓게 채택되고 있다. 움직임 추정 및 보상은 후속 처리와 마찬가지로 블록 기반으로 수행된다. 압축 프로세스 동안, 양자화(quantization)와 같은 손실성 연산(lossy operation)의 적용으로 인해 부호화 잡음(coding noise)이 발생할 수 있다. 특히 블록 경계 또는 블록 경계의 부근에서의 재구성된 비디오 데이터(reconstructed video data)에서는 부호화 결함(coding artifact)이 현저하게 될 수 있다. 부호화 결함의 가시성을 경감시키기 위해, H.264/AVC 및 현재 개발 중인 고효율 비디오 부호화(HEVC : High Efficiency Video Coding) 시스템과 같은 신규의 부호화 시스템에서는 디블로킹(deblocking)으로 지칭되는 기술이 이용되고 있다. 디블로킹 프로세스는 이미지 선명도를 유지하면서 부호화 잡음으로 인한 블록 경계 및 블록 경계 부근에서의 커다란 트랜지션(transition)을 적응 방식으로 평활화하기 위해 블록 경계를 가로질러 필터링을 적용한다. 더욱이, 인터 프레임 부호화(inter-frame coding)의 성질로 인해, 디블로킹 프로세스는 루프내 연산(in-loop operation)을 위해 구성된다. 최근의 HEVC 개발에서, 디블로킹된 재구성 프레임의 처리하기 위해 적응 루프 필터링(ALF : Adaptive Loop Filtering)이 채택되고 있다. 적응 루프 필터링은 디블로킹에 추가하여 루프내 프로세싱으로서 이용되며, 재구성된 비디오 데이터의 디블로킹 후에 적용되는 경우가 많다.

[0006] 종래의 적응 루프 필터는 단지 필터링이 성능을 향상시키는데 도움을 주는 블록에만 적용된다. 적응 루프 필터링이 성능을 향상시키는데 도움을 주지 못하는 다른 블록에 대해서는, 적응 루프 필터링이 적용되지 않는다. ALF가 적용되는 때에, 슬라이스(slice) 내의 블록에 단일 필터(SF : Single Filter)가 적용된다. 단일 필터는 최소 왜곡, 최소 레이트 또는 최상의 R-D 성능과 같은 최상의 성능을 달성하기 위한 필터 후보의 군으로부터 선택된다. 이러한 ALF 방식은 SF(단일 필터) ALF로서 지칭된다. 또 다른 ALF 방식(QC\_ALF로 지칭되는)은 Qualcomm(2010년 4월 15-23에 독일 Dresden에서 개최된 Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 1st Meeting에서 Karczewicz 등에 의해 발표된 "Video coding technology proposal by Qualcomm Inc.", Document: JCTVC-A121)에 의해 제안되어 있다. QC\_ALF에 따라, ALF는 디블로킹된 비디오 데이터에 블록별로(on a block by block basis) 적용된다. 각각의 블록에 대해, 블록의 각각의 픽셀  $(i, j)$ 의 변형 라플라시안 합 측정(Sum-modified Laplacian Measure, SLM)  $SLM(i, j)$ 이 계산된다. 블록의 각각의 픽셀은 그 후 SLM 측정치에 따라 복수의 부류 또는 범주로 분류되며, 각각의 ALF 필터가 각각의 픽셀에 대해 선택된다. QC\_ALF 방식이 SF ALF 이상으로 성능을 향상시킬 수 있지만, 각각의 픽셀에 대한 SLM 계산 및 픽셀에서 픽셀로의 필터 스위칭은 계산 복잡도(computational complexity)를 더 높아지게 하고,

더 많은 파워를 소비하게 할 것이다. 높은 계산 복잡도 및 픽셀에서 픽셀로의 스위칭의 필요 없이 각각의 영역에 대한 복수의 필터 선택의 장점을 제공하는 ALF 방식을 개발하는 것이 바람직하다.

**발명의 내용**

**해결하려는 과제**

[0007]

영역 파티션(region partition) 및 모드 결정(mode decision)의 면에서의 ALF 설계 동안, 레이트-왜곡 최적화(RDO : rate-distortion optimization)로 지칭되는 최적화 기술이 영역 파티션 및 모드 결정을 가이드하기 위해 이용되는 경우가 있다. RDO 기술이 최상의 레이트-왜곡 성능을 달성하지만, 이 기술은 시스템 복잡도를 고려하지 않아서, 소중한 시스템 자원을 소비할 수도 있다. 레이트, 왜곡 및 복잡도의 면에서 최상의 성능을 달성하는 최적화 방식을 이용하는 것이 바람직하다. 따라서, ALF 설계를 위해 레이트-왜곡-복잡도 최적화(RDCO : rate-distortion-complexity optimization) 기술이 이용된다. RDCO 기술은 또한 최상의 레이트-왜곡-복잡도 성능을 달성하기 위해 샘플 적응 오프셋(SAO : sample adaptive offset) 설계에 적용된다.

**과제의 해결 수단**

[0008]

부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법 및 장치를 개시한다. 본 발명에 따른 일실시예에서, 부호화된 비디오를 적응 루프 필터를 이용하여 처리하는 방법 및 장치는 부호화된 비디오 데이터에 대응하는 재구성된 비디오 데이터(reconstructed video data)를 처리 유닛으로부터 수신하는 것과, 필터링된 비디오 데이터를 생성하기 위해 상기 재구성된 비디오 데이터에 적응 루프 필터링을 적용하는 것과, 상기 필터링된 비디오 데이터를 제공하는 것을 포함한다. 적응 루프 필터링은 영역 적응 플래그(region adaptation flag)에 따라 영역 기반 ALF(region based ALF) 또는 비영역 기반 ALF(non-region based ALF)가 될 수 있다. 상기 영역 기반 ALF가 선택되면, 상기 재구성된 비디오 데이터가 M×N 영역으로 분할될 수 있으며, 여기서 M과 N은 정수이다. M×N 영역의 영역들은 1-D 또는 2-D 인덱스 중의 하나를 이용하여 병합될 수 있다. 더욱이, 영역이 하나 이상의 인접 영역과 병합되는지를 나타내기 위해 병합 플래그가 이용될 수 있다. 1-D 인덱스가 이용될 때, M×N 영역의 각각에 그룹 인덱스를 할당하기 위해 스캐닝 패턴이 이용될 수 있으며, 스캐닝 패턴은 힐버트 곡선(Hilbert curve)의 변형, 수평 스네이크 스캔(horizontal snake scan), 수직 스네이크 스캔(vertical snake scan), 지그-재그 스캔(zig-zag scan), 나선형 스캔(spiral scan), 쿼드-트리 스캔(quad-tree scan), 및 래스터 스캔(raster scan)으로 이루어진 군에서 선택된다. 본 발명에 따른 또 다른 실시예에서, M×N 영역의 각각의 영역은 가장 큰 부호화 단위(LCU : largest coding unit)의 경계로 정렬된다. 본 발명에 따른 또 다른 실시예에서, 영역 병합이 허용되는지의 여부를 나타내기 위해 병합 플래그가 이용된다.

[0009]

적응 루프 필터(ALF) 설계 또는 샘플 적응 오프셋(SAO : sample adaptive offset) 설계를 위한 방법 및 장치가 개시된다. 본 발명에 따른 일실시예에서, 적응 루프 필터(ALF) 설계 또는 샘플 적응 오프셋(SAO) 설계를 위한 방법 및 장치는, 적응 루프 필터(ALF) 설계 또는 샘플 적응 오프셋(SAO) 설계를 위한 설계 특징(design feature)에 연관된 후보를 결정하는 것과, 레이트-왜곡-복잡도 최적화(RDCO : rate-distortion-complexity optimization)에 따라 상기 후보 중에서 최상의 후보를 선택하는 것을 포함한다. 상기 설계 특징은 모드 결정(mode decision) 또는 영역 파티션에 연관될 수 있다. ALF 설계를 위해서는, RDCO의 복잡도가 필터 계수의 개수에 관련되거나, 또는 필터 계수의 개수와 필터링될 픽셀의 개수의 조합에 관련되거나, 또는 필터 계수의 개수와 필터링될 픽셀의 개수와 모드 결정을 위한 후보 모드에 연관된 연산의 개수의 조합에 관련된다. SAO 설계를 위해서는, RDCO의 복잡도가 픽셀 분류 타입에 연관된 연산의 개수에 관련되거나, 픽셀 분류 타입에 연관된 연산의 개수와 오프셋 값으로 보상될 픽셀의 개수의 조합에 관련될 수 있다.

**도면의 간단한 설명**

[0010]

도 1은 재구성된 비디오 데이터에 적응 루프 필터가 적용되는 움직임 보상 예측을 기반으로 하는 비디오 부호화 시스템의 일례의 블록도이다.

도 2는 6×4 블록 내의 픽셀을 3개의 범주로 분류하기 위해 변형 라플라시안 합 측정(SLM)이 이용되는 픽셀 기반의 적응 루프 필터(pixel based adaptive loop filter)의 예를 도시하는 도면이다.

도 3은 416×240 픽셀로 구성되는 화상을 4×4 LCU-정렬 영역으로 분할하는 예를 도시하고 있으며, 여기서 각각의 정사각형은 64×64 픽셀로 이루어진 LCU이며, 최우측이 아니면서 바닥이 아닌 영역은 2×1 LCU로 구성된다.

도 4는 영역 기반 ALF가 이용되는지의 여부를 나타내기 위해 플래그 **region\_adaptation\_flag**를 통합하고 있는

영역 기반 ALF를 지원하기 위한 일례의 선택스 설계(syntax design)를 도시하는 도면이다.

도 5는 변형 힐버트 곡선(deformed Hilbert curve)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 6은 수평 스네이크 스캔(horizontal snake scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 7은 수직 스네이크 스캔(vertical snake scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 8은 지그-재그 스캔(zig-zag scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 9는 나선형 스캔(spiral scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 10은 쿼드-트리 스캔(quad-tree scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 11은 래스터 스캔(raster scan)에 따른 4×4 영역을 통한 스캐닝 순서를 도시하는 도면이다.

도 12는 영역을 5개의 영역, 즉  $F_0$ ,  $F_1$ ,  $F_2$ ,  $F_3$  및 노-필터 영역으로 나누는 영역 나눔(region splitting)의 예를 도시하는 도면이다.

도 13은 5개의 필터링될 영역을 3개의 영역  $F_0'$ ,  $F_1'$  및 노-필터 영역으로 병합하는 예를 도시하는 도면이다.

도 14는 2-D 영역 병합을 지원하기 위한 일례의 선택스 설계를 도시하는 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0011]

디지털 비디오 압축에 대해서는, 움직임 보상 인터 프레임 부호화가 효율적인 압축 기술이며, MPEG-1/2/4 및 H.261/H.263/H.264/AVC와 같은 다양한 부호화 표준에 폭 넓게 채택되고 있다. 움직임 보상 시스템에서, 움직임 추정/보상 및 후속 압축은 블록별로 수행되는 경우가 있다. 압축 프로세스 동안, 양자화와 같은 손실성 연산으로 인해 부호화 잡음이 발생할 수 있다. 특히 블록 경계 또는 블록 경계의 부근에서의 재구성된 비디오 데이터에서는 부호화 결함이 현저하게 될 수도 있다. 부호화 결함의 가시성을 완화시키기 위해, H.264/AVC 및 현재 개발 중인 고효율 비디오 부호화(HEVC) 시스템과 같은 신규의 부호화 시스템에서는 디블로킹(deblocking)으로 지칭되는 기술이 이용되고 있다. 디블로킹 프로세스는 이미지 선명도를 유지하면서 부호화 잡음으로 인한 블록 경계 및 블록 경계 부근에서의 커다란 트랜지션(transition)을 적응 방식으로 평활화하기 위해 블록 경계를 가로질러 필터링을 적용한다. 더욱이, 인터 프레임 부호화의 성질로 인해, 디블로킹 프로세스는 루프내 연산(in-loop operation)을 위해 구성된다.

[0012]

HEVC에서는, 디블로킹에 추가하여 적응 루프 필터링(ALF)으로 지칭되는 또 다른 루프내 필터링이 이용된다. 디블로킹 필터가 단지 블록 경계에 적용되지만, 적응 루프 필터는 프레임, 필드, 슬라이스 또는 화상 영역 내의 모든 픽셀에 적용될 수 있다. 종래의 적응 루프 필터는 단지 필터링이 성능을 향상시키는데 도움을 주는 블록에만 적용된다. 적응 루프 필터가 성능을 향상시키는데 도움을 주지 못하는 다른 블록에 대해서는, 적응 루프 필터링이 적용되지 않는다. ALF가 적용되는 때에, 슬라이스(slice) 내의 블록에 단일 필터 SF가 적용된다. 단일 필터는 최소 왜곡, 최소 레이트 또는 최상의 R-D 성능과 같은 최상의 성능을 달성하기 위한 필터 후보의 군으로부터 선택된다. 이러한 ALF 방식은 SF(싱글 필터) ALF로서 지칭된다. 선택된 필터에 연관된 정보는 디코더측에 보내져야 한다. 전송되거나 저장될 정보를 보존하기 위해, ALF 필터의 세트가 사전에 정해질 수 있으며, 필터 선택이 인덱스에 의해 지시될 수 있다. 이와 달리, 필터는 이미 재구성된 비디오 데이터에 기초한 시간 지연 배열(time-delayed arrangement)에서 얻어질 수 있다. 따라서, 보조 정보(side information)가 요구되지 않거나, 매우 적은 보조 정보가 요구된다. 필터 계수의 엔트로피 부호화(entropy coding) 및/또는 계수를 별도로(differentially) 전송하는 것과 같은, 필터에 연관된 정보를 감소시키기 위한 다른 수단이 이용될 수 있다.

[0013]

적응 루프 필터링 및 디블로킹을 통합하는 부호화 시스템에 대한 시스템 블록도가 도 1에 도시되어 있다. 도 1은 디블로킹을 갖는 움직임 보상 비디오 인코더의 시스템 블록도이다. 압축 시스템(100)은 압축 비디오 데이터를 생성하기 위해 인트라/인터-예측(Intra/Inter-prediction), 이산 코사인 변환(DCT) 및 엔트로피 부호화를 통합하는 대표적인 비디오 인코더를 예시한다. 입력 비디오 데이터는 입력 인터페이스(112)를 통해 인코더에 입력되며, 인트라/인터-예측 블록(110)을 이용하여 예측된다. 인트라 예측 모드에서, 인입되는 비디오 데이터는 이미 부호화된 동일한 프레임 화상 또는 필드 화상 내의 주변 데이터에 의해 예측된다. 인터 예측 모드에서, 예측은 재구성된 데이터(142)가 화상 버퍼(140)에 저장되는 시간적인 방향에서 이전에 재구성된 데이터를 기반



으로 하여 이루어진다. 인터 예측은, 디코딩 순서에 있어서 현재 화상보다 앞서고 있고 참조 화상의 제1 리스트에 있는 화상에 기초하여 예측이 이루어지는 리스트 0 예측 모드일 수 있다. 인터 예측은 또한 리스트 1 예측 모드일 수 있으며, 이 예측 모드에서, 인터 예측은 디코딩 순서에 있어서 현재 화상보다 앞서고 있고 현재 슬라이스가 B-슬라이스이면 참조 화상의 제2 리스트에 있는 화상에 기초하여 이루어진다. 인터 예측은 또한 양방향 모드일 수 있으며, 이 모드에서 인터 예측은 현재 슬라이스가 B-슬라이스이면 리스트 0 참조 화상 및 리스트 1 참조 화상에 기초하여 이루어진다. 인터 예측 모드에서, 인트라/인터 예측 블록(110)은 예측 데이터가 가산기(115)에 제공되어 원본 비디오 데이터(112)로부터 감산되게 할 것이다. 가산기(115)로부터의 출력은 예측 오차로서 지칭되며, 이 예측 오차가 이산 코사인 변환 및 양자화(Q)를 나타내는 DCT/Q 블록(120)에 의해 추가로 처리된다. DCT 및 양자화 블록(120)은 예측 오차(117)를 부호화된 심볼로 변환하며, 이 부호화된 심볼이 엔트로피 부호화(130)에 의한 추가로 처리되어 압축 비트스트림(132)을 발생하며, 이 압축 비트스트림이 저장되거나 전송된다. 인트라/인터 압축을 위한 예측 데이터를 제공하기 위해, DCT 및 양자화 블록(120)에 의해 처리된 예측 오차는 재구성된 예측 오차(162)를 제공하기 위해 역 DCT 및 역양자화(IDCT/IQ) 블록(160)에 의해 복원되어야 한다. 인터 예측 모드에서, 재구성된 예측 오차(162)는 현재 재구성된 프레임(152)을 형성하기 위해 재구성 블록(150)에 의해 이전에 재구성된 비디오 데이터(119)에 추가된다. 인트라 예측 모드에서, 재구성된 예측 오차(162)는 동일 화상에서의 이전에 재구성된 주변 데이터에 추가된다. 인트라/인터 예측 블록(110)은 이전에 재구성된 데이터(119)를 재구성 블록(150)에 라우팅하도록 구성되며, 여기서 재구성된 데이터(119)는 인트라/인트라 모드에 좌우되어 시간적인 방향에서의 이전에 재구성된 프레임에 대응하거나 또는 동일한 화상에서의 재구성된 주변 데이터에 대응할 수 있다.

[0014]

재구성된 데이터는 디블로킹 블록(170) 및 적응 루프 필터링 블록(180)에 의해 처리되며, 그리고 나서 후속 화상의 처리를 위한 참조 비디오 데이터로서 화상 버퍼(140)에 저장된다. HEVC를 위해 제안된 본래의 ALF는 블록별로 적용된다. ALF가 성능의 향상(더 낮은 왜곡, 더 낮은 비트 레이트 또는 더 우수한 R-D 성능)에 도움을 준다면, ALF는 블록에 대해 턴온된다. ALF가 성능의 향상에 도움을 주지 않는다면, ALF는 블록에 대해 턴오프된다. ALF 방식(QC\_ALF로 지칭되는)은 Qualcomm(2010년 4월 15-23에 독일 Dresden에서 개최된 Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 1st Meeting에서 Karczewicz 등에 의해 발표된 "Video coding technology proposal by Qualcomm Inc.", Document: JCTVC-A121)에 의해 제안되어 있다. QC\_ALF에 따라, ALF는 디블로킹된 비디오 데이터에 픽셀별로(on a pixel by pixel basis) 적용된다. 각각의 블록에 대해, 블록 내의 각각의 픽셀  $(i, j)$ 의 변형 라플라시안 합 측정(SLM)  $SLM(i, j)$ 이 계산된다:

$$SLM(i, j) = \sum_{k=-K}^K \sum_{l=-L}^L [2R(i+k, j+l) - R(i+k-1, j+l) - R(i+k+1, j+l) + 2R(i+k, j+l) - R(i+k, j+l-1) - R(i+k, j+l+1)]$$

[0015]

[0016]

여기서,  $R(i, j)$ 는 디블로킹된 비디오 데이터이다. SLM은  $(i, j)$ 에 중심을 두고 있는  $(2K+1) \times (2L+1)$  인접 범위(neighboring window)에 기초하여 계산된다. 인접 범위 크기는  $9 \times 9$ ,  $7 \times 7$ ,  $5 \times 5$ , 또는  $3 \times 3$ 이 될 수 있다. 복잡도를 감소시키기 위해,  $3 \times 3$  인접 범위가 이용되고 있다. 블록의 각각의 픽셀에 대해 계산된 SLM 값은 픽셀을 M개의 그룹 중의 하나로 분류하기 위해 이용된다. 도 2는 각각의 정사각형이 픽셀을 나타내고 있는 SLM 분류의 예를 도시한다. 픽셀은 도 2에 도시된 바와 같이 SLM 값에 따라 3개의 그룹으로 분류된다. QC\_ALF 방식은 그룹과 관련된 픽셀을 필터링하기 위해 각각의 그룹에 대응하는 필터를 선택한다. ALF를 위해 이용된 필터는 계산 복잡도를 감소시키기 위해 수평 및 수직으로 대칭을 이루는 경우가 있다. 복잡도를 추가로 감소시키기 위해, 특히 대형 크기 필터의 경우에 다이아몬드 형상 필터가 이용될 수 있다. 예컨대,  $9 \times 9$ ,  $7 \times 7$  또는  $5 \times 5$  다이아몬드 형상 필터, 또는  $5 \times 5$  또는  $3 \times 3$  정사각형 필터가 이용될 수 있다. SLM 기반 ALF가 픽셀별로 적용되며, 픽셀은 상이한 필터를 이용할 수 있다. 따라서, QC\_ALF는 또한 픽셀-적응(pixel-adaptive 또는 pixel-adaptation)(PA) ALF로서 지칭된다. 전술한 SF ALF는 단지 하나의 그룹이 이용되는 PA ALF의 특수한 경우로서 간주될 수 있다.

[0017]

PA ALF가 필터를 픽셀별로 적용 방식으로 선택할 수 있지만, 이것은 요구된 보조 정보가 부호화되면 이 보조 정보가 상당한 것이 될 것이기 때문에 디코더측에서 각각의 픽셀에 대한 SLM에 기초하여 그룹 정보를 얻는 것을 요구한다. 따라서, 더 우수한 성능 및/또는 감소된 복잡도를 달성할 수 있는 ALF 방식을 개발하는 것이 바람직하다. 이에 따라, 본 명세서에서는 영역 기반의 ALF 방식이 개시된다. 본 발명에 따른 ALF 방식은 ALF를 디블로킹된 비디오 데이터에 영역별로(on a area by area basis) 적용한다. 영역은 화상 또는 화상 영역을 고정된 블록 또는 고정된 세트의 블록으로 분할함으로써 형성될 수 있다. 이와 달리, 영역은 화상 또는 화상 영역을

재귀적으로(recursively) 파티셔닝함으로써 형성될 수 있다. 예컨대, 재귀적 영역 파티셔닝을 위해 쿼드트리 이용될 수 있다. 영역 기반 ALF가 적용되는지 아니면 비영역 기반 ALF 방법이 적용되는지의 여부를 나타내기 위해 ALF 정보 세트에 대한 선택에서의 플래그가 이용된다. 예컨대, 영역 기반 ALF와 블록 기반 ALF 간에 선택하기 위해 플래그가 이용될 수 있다. 블록을 위해 선택된 필터는 픽셀 기반 ALF에서 이용된 유사 방법으로부터 얻어질 수 있다. 예컨대, Chong 등은 각각의 4×4 블록이 하나의 라플라시안 활성화도 값(Laplacian activity value)을 이용할 수 있도록 4×4 블록의 라플라시안 기반 활동도 측정지표(Laplacian based activity metrics)의 평균이 구해지는 블록 기반 ALF를 설명하였다(2011년 3월 16-23일에 스위치 제노바에서 개최된 Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 5th Meeting에서 Chong 등이 발표한 "CE8 Subtest 2: Block based adaptive loop filter(ALF)", Document: JCTVC-E323). 이 방법은 4×4 블록의 각각의 픽셀에 대한 라플라시안 활성화도 값을 계산하는 것을 필요로 하며, 종래의 픽셀 기반 ALF 이상으로 계산을 줄이지는 못한다. 그러나, 블록 기반 ALF는 픽셀 기반 ALF에 비하여 필터 스위칭 활성화도의 빈도를 감소시킬 수 있다. 라플라시안 활성화도를 분류기(classifier)로 이용하는 대신, 샘플 적응 오프셋(SAO)을 위해 이용되는 밴드 오프셋(BO) 또는 에지 오프셋(EO) 분류기와 같은 기타 측정이 이용되어, Karczewicz 등에 의해 이용되는 SLM 계산을 대체할 수 있다. BO 및 EO 양자는 SLM 측정에 비하여 훨씬 더 적은 계산을 요구한다.

[0018]

블록의 세트를 이용하는 영역 기반 ALF의 예로서, 화상이 16개(즉, 4×4)의 대략 동일 크기의 영역으로 분할될 수 있다. 예컨대, 최우측이 아닌 영역의 영역 폭은 (*PicWidth*/4)가 될 수 있으며, 여기서 *PicWidth*은 화상 폭을 의미한다. 최우측 영역에 대해서는, 영역 폭은 (*PicWidth*-(*PicWidth*/4)\*3)이 될 수 있다. 바닥이 아닌 영역의 영역 높이는 (*PicHeight*/4)이 될 수 있으며, 여기서 *PicHeight*는 화상 높이를 의미한다. 바닥 영역에 대해, 영역 높이는 (*PicHeight*-(*PicHeight*/4)\*3)이 될 수 있다. 이 예에서, 최우측 영역 및 바닥 영역은 다른 영역보다 크게 될 수도 있다. 또 다른 예는 화상을 16개(즉, 4×4)의 대략 동일 크기의 LCU 정렬 영역으로 파티션할 것이며, 여기서 영역 경계는 또한 도 3에 도시된 바와 같이 가장 큰 부호화 단위(LCU : largest coding unit) 경계가 될 수 있다. 화상 크기는 416×240이고, 7×4 LCU를 포함하며, 여기서 각각의 LCU는 64×64 픽셀을 갖는다. 최우측이 아닌 영역의 영역 폭은 (((*PicWidthInLCUs*+1)/4)\*64)가 될 수 있으며, 여기서 *PicWidthInLCUs*는 화상 폭의 LCU의 개수를 의미한다. 바닥이 아닌 영역의 영역 높이는 (((*PicHeightInLCUs*+1)/4)\*64)일 수 있으며, 여기서 *PicHeightInLCUs*는 화상 높이의 LCU의 개수를 의미한다. 최우측 및 바닥 영역의 크기는 *PicWidth*, *PicHeight*, 및 최우측이 아니면서 바닥도 아닌 영역의 크기로부터 얻어질 수 있다. 4×4 영역의 각각에 인덱스가 할당된다. 4×4 영역이 예로서 이용되지만, 본 발명은 특정한 4×4 영역으로 한정되지 않는다는 것을 이해할 것이다. 실제로, 본 발명을 실시하기 위해 M×N 영역이 이용될 수 있으며, 여기서 M과 N은 정수이다. 수평으로는 *PicWidthInSamples* 픽셀을 갖고 수직으로는 *PicHeightInSamples* 픽셀을 갖는 화상에 대해, 상단 좌측 모서리 좌표 (x,y)를 갖는 영역에 대한 영역 인덱스는 다음과 같이 얻어질 수 있다. (1<<Log2MaxCUSize)는 최대 부호화 단위 크기이며, *xMaxIndex*와 *yMaxIndex*는 각각 수평 방향과 수직 방향에서의 최대 영역 인덱스이다. 영역의 수평 간격 *x\_interval*과 수직 간격 *y\_interval*은 다음과 같이 얻어질 수 있다:

[0019]

$$x\_interval = (((PicWidthInSamples + (1 \ll \text{Log2MaxCUSize}) - 1) \gg \text{Log2MaxCUSize}) + x\_round) / (xMaxIndex + 1), \text{ 및} \quad (1)$$

[0020]

$$y\_interval = (((PicHeightInSamples + (1 \ll \text{Log2MaxCUSize}) - 1) \gg \text{Log2MaxCUSize}) + y\_round) / (yMaxIndex + 1), \quad (2)$$

[0021]

여기서,  $x\_round = \max(0, (xMaxIndex + 1) / 2 - 1)$ 이고,  $y\_round = \max(0, (yMaxIndex + 1) / 2 - 1)$ 이다. 수평 인덱스 *index x\_idx* 및 수직 인덱스 *index y\_idx*는 다음과 같이 얻어질 수 있다:

[0022]

$$x\_idx = \min(xMaxIndex, x / (x\_interval \ll \text{Log2MaxCUSize})), \text{ 및} \quad (3)$$

[0023]

$$y\_idx = \min(yMaxIndex, y / (y\_interval \ll \text{Log2MaxCUSize})) \quad (4)$$

[0024]

영역 인덱스 *region\_idx*는 다음에 따라 결정된다:

[0025]

$$region\_idx = y\_idx * (xMaxIndex + 1) + x\_idx \quad (5)$$

[0026]

영역 기반 ALF에 대해 4×4 영역이 이용될 때, 영역 인덱스는 다음과 같이 얻어질 수 있다. 영역의 수평 간격 *x\_interval*과 수직 간격 *y\_interval*은 다음과 같이 얻어질 수 있다:

[0027]  $x\_interval = (((PicWidthInSamples+(1<<Log2MaxCUSize)-1)>>Log2MaxCUSize)+ 1)>>2)$ , 및 (6)

[0028]  $y\_interval = (((PicHeightInSamples+(1<<Log2MaxCUSize)-1)>>Log2MaxCUSize)+ 1)>>2)$  (7)

[0029] 수평 인덱스  $x\_idx$ 와 수직 인덱스  $y\_idx$ 는 다음과 같이 얻어질 수 있다:

[0030]  $x\_idx = \min(3, x/(x\_interval<<Log2MaxCUSize))$ , 및 (8)

[0031]  $y\_idx = \min(3, y/(y\_interval<<Log2MaxCUSize))$  (9)

[0032] 영역 인덱스  $region\_idx$ 는 다음에 따라 결정된다:

[0033]  $region\_idx = (y\_idx<<2) + x\_idx$  (10)

[0034] 필터 인덱스  $filter\_idx(x,y)$ 는 다음에 따라 결정된다:

[0035]  $filter\_idx(x, y) = region\_tab [region\_idx]$  (11)

[0036] 여기서,  $region\_tab [16] = \{0, 1, 4, 5, 15, 2, 3, 6, 14, 1, 10, 7, 13, 12, 9, 8\}$ .

[0037] 2개의 인접 영역, 즉 연속적인 인덱스를 갖는 영역이 병합될 수 있다. 영역 병합 후, 각각의 병합된 영역에 하나의 필터가 적용된다. 화상은 그 후 픽셀 기반 ALF, 블록 기반 ALF, 또는 영역 기반 ALF를 이용함으로써 처리될 수 있다. 픽셀 기반 ALF 및 블록 기반 ALF에 비하여, 영역 기반 ALF는 각각 픽셀 적용의 픽셀 기반 분류(pixel-based classification of pixel adaptation) 및 블록 적용의 블록 기반 분류(block-based classification of block adaptation)에 관련된 계산을 줄일 수 있어서, 영역 기반 ALF가 통합되는 때에는 평균 디코더 복잡도가 감소될 수 있다. 더욱이, 영역 기반 ALF는 또한 화상에서의 필터 스위칭의 수를 현저하게 감소시키며, 그 결과 스위칭 파워를 더 낮게 한다. 영역 기반 ALF와 픽셀 기반 ALF 간 또는 영역 기반 ALF와 블록 기반 ALF 간의 선택을 지원하기 위한 신택스의 예가 도 4에 도시되어 있다. 유일한 신택스 변경(syntax change)은 픽셀 기반 ALF와 영역 기반 ALF 간 또는 블록 기반 ALF와 영역 기반 ALF 간에 선택하기 위해 슬라이스 헤더 또는 화상 파라미터 세트(PPS)의 ALF 파라미터 세트  $alf\_param()$ 에 하나의 플래그 **Region\_adaptation\_flag**를 추가하는 것이다.

[0038] 영역 병합에 의해 부호화 효율이 추가로 향상될 수 있다. 인접 영역은 유사한 특성을 가질 수 있으며, ALF 필터를 나타내기 위해 요구되는 정보를 감소시키기 위해 동일한 필터를 공유할 수 있다. 영역 병합을 수행하기 위한 한 가지 방법은 2-D 영역을 1-D 영역으로 정리(order)하는 것이다. 예컨대,  $4 \times 4$  영역은 0 내지 15의 그룹 인덱스를 갖는 1-D 영역으로 전환될 수 있다. 2-D에서 1-D로의 전환은 구체적인 스캐닝 패턴을 따라 수행될 수 있다. 변형 힐버트 곡선(deformed Hilbert curve)(도 5), 수평 스네이크 스캔(도 6), 수직 스네이크 스캔(도 7), 지그-재그 스캔(도 8), 나선형 스캔(도 9), 쿼드-트리 스캔(도 10), 및 래스터 스캔(도 11)과 같은, 2-D 영역을 1-D 영역으로 변환하기 위해 이용될 수 있는 다수의 공지된 스캔 패턴이 있다. 2-D 영역을 1-D 영역으로 변환 시에, 인접 영역, 즉 연속적인 그룹 인덱스를 갖는 영역이 병합 신택스에 의해 지시된 것과 동일한 필터를 공유하도록 병합될 수 있다. 예컨대, 그룹 인덱스  $n$ 을 갖는 영역이 그룹 인덱스  $(n-1)$ 를 갖는 영역과 병합되는지의 여부를 나타내기 위해 하나의 병합 플래그가 이용될 수 있다.

[0039] 영역 병합은 또한 영역이 주변 영역과 병합될 수도 있는 2-D 병합 신택스를 이용하여 병합이 수행될 수도 있다. 더욱이, 이 영역이 최초로 병합되는지의 여부를 나타내기 위해 제1 플래그가 이용될 수 있다. 제1 플래그가 이 영역이 병합되는 것으로 나타내면, 제1 플래그는 하나의 병합 후보 플래그가 후속된다. 어떠한 환경에서, 주변 영역 중의 일부는 병합 후보로 이용 가능하지 않을 수도 있다. 예컨대, 화상의 경계 상의 또는  $4 \times 4$  영역의 경계 상의 영역은 특정한 병합 후보를 갖지 않을 것이다. 이에 따라, 병합 후보 플래그는 인접 영역에 기초하여 간략화될 수 있다. 예컨대, 병합 후보는 좌측 또는 상위 영역일 수 있고, 병합 후보 플래그로서 1-비트 플래그가 이용될 수 있다. 그러나, 좌측 및 상위 영역이 동시에 존재하지 않으면, 병합 후보 플래그가 절감될 수 있다. 이에 따라, 병합 후보 플래그의 표시는 인접 영역의 이용 가능성에 맞추어진다. 인접 영역의 일부가 이용 가능하지 않을 때에는, 병합 후보 플래그는 더 적은 비트로 표현될 수 있다. 영역 병합은  $4 \times 4$  영역의 각각의 세트에 적용 방식으로 적용될 수 있다.  $4 \times 4$  영역에 대해 영역 병합이 허용되는지를 나타내기 위해 영역 병합 인에이블 플래그가 이용될 수 있다.

[0040] ALF 필터링을 위한 영역은 또한 필터 단위(FU : filter unit)가 될 수 있다. 화상 또는 화상 영역은, 영역 나뭇이 더 낮은 레이트-왜곡(R-D) 비용(lower rate-distortion cost)과 같은 더 우수한 성능을 발생할 것이라면, 더 작은 FU로 재귀적으로 파티션될 수 있다. 하나의 필터링될 영역(이 경우에는 개개의 FU)과 영역 나눔으로

비롯되는 영역에 대하여 R-D 비용이 계산될 수 있다. 영역 나눔이 더 낮은 R-D 비용을 발생시킬 것이라면, 필터링될 영역은 나누어질 것이며, 그렇지 않은 경우에는, 필터링될 영역은 나누어지지 않는다. 이와 달리, 화상 또는 화상 영역이 먼저 더 작은 영역으로 분할될 수도 있다. 인접 영역은 이들의 병합이 더 낮은 R-D 비용과 같은 더 우수한 성능을 발생시킬 것이라면 병합될 수 있다. 병합된 영역에 대한 비용이 개별 영역들에 대한 비용보다 작다면, 이 영역들은 병합될 것이며, 그렇지 않은 경우에는, 이 영역들은 병합되지 않을 것이다. 본 발명을 구현하는 방법에 따라, 필터 정보에 연관된 비트 레이트를 감소시키기 위해 여러 개의 필터링될 영역이 하나의 FU로 병합될 수 있다.

[0041] 도 12에는 영역(필터 단위 FU 1210)을 5개의 영역(FU 1220), 즉  $F_0, F_1, F_2, F_3$  및 노-필터 영역으로 나누는 영역 나눔의 예가 도시되어 있다. 영역 나눔은 더 낮은 R-D 비용에 따라 이루어질 수 있다. 도 13은 본 발명의 일실시예에 따른 영역 병합의 예를 도시하는 도면이다. 필터링될 영역(1310)은 R-D 비용에 따라 영역 병합이 고려된다. 개개의 인접 영역에 대한 R-D 비용이 이들 개개의 인접 영역에 대응하는 병합된 영역의 R-D 비용과 비교된다. 개개의 인접 영역은 이들의 병합이 더 낮은 R-D 비용을 발생하면 병합된다. 도 13의 예는 5개의 필터링될 영역(1310)을 3개의 영역(1320)  $F_0', F_1'$  및 노-필터 FU로 병합하는 경우를 도시하는 도면이다. 영역  $F_0$  과  $F_2$ 는  $F_0'$ 으로 병합되고, 영역  $F_1$ 과  $F_3$ 는  $F_1'$ 으로 병합된다.

[0042] 도 14에는 2-D 영역 병합을 지원하기 위한 일례의 선택스 설계가 도시되어 있다. *alf\_fs\_selection\_param*(*r, c*) 함수는 영역 병합을 위한 필터 공유를 처리한다. ALF 병합 플래그 **alf\_merge\_flag**는 현재 영역(즉, 이 예에서는 FU)이 또 다른 영역과 병합되는지를 지시하며, 여기서 값 1은 현재 영역이 병합된다는 것을 나타내고, 값 0는 현재 FU가 병합되지 않는다는 것을 나타낸다. 현재 영역의 필터 세트 인덱스 **alf\_fu\_filter\_set\_idx**는 현재 영역이 병합되지 않을 때에는 통합된다. 화상의 최초의 영역은 항상 0로 설정된 필터 세트 인덱스를 가지며, 인덱스를 보낼 필요가 없다. 선택스 요소 **alf\_merge\_up\_flag**는 영역이 상단층의 영역과 병합되는지를 지시하며, 여기서 값 0는 현재 영역이 좌측 FU와 병합된다는 것을 나타내고, 값 1은 현재 영역이 상위 영역과 병합된다는 것을 나타낸다.

[0043] 본 발명의 또 다른 특징은 ALF 필터 설계에 관련된다. 레이트-왜곡 최적화(RDO)는 우수한 부호화 효율을 획득하기 위해 비디오 인코딩에 이용되는 널리 알려진 기술이다. RDO는 통상적으로 매크로블록 또는 서브-매크로블록 또는 부호화 단위의 모드 설계, 인트라 예측, 움직임 추정, 적응 루프 필터(ALF) 결정(필터 크기, 필터 형상, 온/오프 스위치 등에 대해), 및 샘플 적응 오프셋(SAO) 결정(상이한 픽셀 분류 방법에 대해) 동안 적용된다. 최상의 RDO 결정은 레이트-왜곡 비용 함수  $J=D+\lambda R$ 를 최소화하는 결정이며, 여기서 D는 원본 픽셀과 재구성된(또는 예측된) 픽셀 간의 추정된 왜곡이고, R은 보조 정보를 보내기 위해 요구되는 추정된 레이트이고,  $\lambda$ 는 라그랑주 승수(Lagrange multiplier)이다.

[0044] 레이트-왜곡-복잡도 최적화(RDCO) 또한 부호화 효율과 부호화 복잡도 간의 우수한 균형(trade-off)을 획득하기 위해 비디오 인코딩에서 폭 넓게 이용되는 널리 공지된 기술이다. RDCO는 통상적으로 매크로블록, 서브-매크로블록 또는 부호화 단위의 모드 결정, 인트라 예측 및 움직임 추정 동안 적용된다. 최상의 RDCO 결정은 레이트-왜곡-복잡도 비용 함수  $J=D+\lambda_1 R+\lambda_2 C$ 를 최소화하는 결정이며, 여기서 D는 원본 픽셀과 재구성된(또는 예측된) 픽셀 간의 추정된 왜곡이고, C는 인코딩 또는 디코딩을 위해 요구된 추정된 복잡도(클록 사이클, 메모리 액세스, 또는 기타 복잡도 측정치)이며,  $\lambda_1$  및  $\lambda_2$ 는 라그랑주 승수이다.

[0045] 본 발명에 따른 실시예는 ALF/SAO 설계를 위해 RDCO를 통합하며, 최상의 후보는 RDCO에 따라 결정된 설계 특징(design feature)에 연관된다. 설계 특징은 모드 결정 또는 영역 파티션일 수 있다. ALF 상이 모드(ALF different mode) 중에서의 모드 결정은 I/B/P-슬라이스와 같은 슬라이스 타입 또는 인트라/인트라 부호화 단위(CU)의 백분율에 좌우될 수 있다. 본 발명에 따른 또 다른 실시예는 ALF 필터 선택을 위해 RDCO를 통합한다. AFL 필터 선택을 위해 RDCO가 이용될 때, RDCO의 복잡도는 필터 계수(filter coefficient)의 개수에 연관될 수 있다. 이와 달리, RDCO는 ALF 필터 선택을 위해 이용될 수 있으며, 여기서 RDCO의 복잡도 C는  $C=A*B$ 와 같이 필터 계수의 개수 A 및 필터링될 픽셀의 개수 B에 연관될 수 있다. 본 발명에 따른 또 다른 실시예에서, RDCO는 ALF 필터 선택을 위해 이용될 수 있으며, 여기서 RDCO의 복잡도 C는  $C=A*B*TableOfNumOperations[Mode]$ 와 같이 필터 계수의 개수 A, 필터링될 픽셀의 개수 B, 및 한 가지 모드에서의 요구된 연산의 개수 *TableOfNumOperations[Mode]*에 연관될 수 있다.

[0046] "Apparatus and Method of Sample Adaptive Offset for Video Coding"을 발명의 명칭으로 하여 2011년 6월 12일자로 출원된 미국 특허 출원 번호 13/158,427호에서는, 샘플 적응 오프셋(SAO) 방식이 개시되어 있으며, 여기

서 각각의 픽셀이 복수의 픽셀 분류 타입을 이용하여 분류되고, 각각의 분류 타입이 픽셀을 복수의 범주로 분류할 수 있다. 픽셀 분류 타입 및 픽셀의 범주에 따라, 오프셋을 보상하기 위해 오프셋 값이 결정된다. SAO 방식은 픽셀을 범주로 분류하기 위해 밴드 오프셋(BO) 컨텍스트(band offset context) 또는 에지 오프셋(E0) 컨텍스트(edge offset context) 중의 하나를 이용한다. 복수의 픽셀 분류 타입에 연관된 복잡도는 상이할 수도 있다. 따라서, RDCO 기술은 최상의 RDCO 성능을 달성하도록 픽셀 분류 타입을 선택하기 위한 SAO 결정을 위해 이용될 수 있다. 미국 특허 출원 번호 13/158,427호에 개시된 발명의 또 다른 특징은 영역 파티션을 어드레스한다. RDCO 기술은 또한 영역 파티션을 결정하기 위한 SAO 결정을 위해 이용될 수 있다. RDCO를 위한 비용은, 처리된 픽셀의 개수와, 모드(즉, 고려되는 픽셀 분류 타입)를 위한 요구된 연산의 개수 TableOfNumOperations[Mode]에 연관될 수 있다.

[0047]

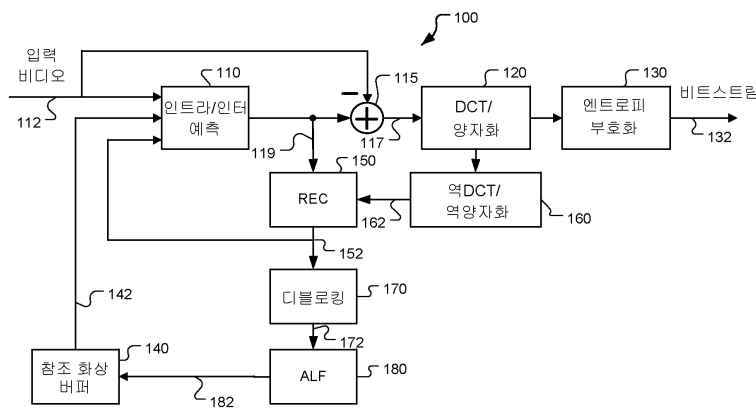
전술한 적응 루프 필터 방식은 비디오 인코더에서뿐만 아니라 비디오 디코더에서도 이용될 수 있다. 비디오 인코더 및 비디오 디코더 둘 모두에서, 재구성된 비디오 데이터는 부호화 결합에 놓이게 된다. 전술한 적응 루프 필터 방식은 재구성된 비디오의 시각적 화질을 향상시키는데 도움을 줄 수 있다. 전술한 바와 같은 본 발명에 따른 영역 기반 ALF를 갖는 인코딩 시스템의 실시예는 다양한 하드웨어, 소프트웨어 코드, 또는 이들 양자의 조합으로 실시될 수 있다. 예컨대, 본 발명의 실시예는 본 명세서에서 설명한 처리를 수행하기 위해 비디오 압축 칩에 통합된 회로 또는 비디오 압축 소프트웨어에 통합된 프로그램 코드일 수 있다. 본 발명의 실시예는 또한 본 명세서에서 설명한 처리를 수행하기 위해 디지털 신호 프로세서(DSP) 상에서 실행될 프로그램 코드일 수 있다. 본 발명은 또한 컴퓨터 프로세서, 디지털 신호 프로세서, 마이크로프로세서, 또는 필드 프로그래머블 게이트 어레이(FPGA)에 의해 수행될 다수의 함수를 수반할 수 있다. 이들 프로세서는 본 발명에 의해 구현된 특정 방법을 정의하는 기기 관독 가능한 소프트웨어 코드 또는 펌웨어 코드를 실행함으로써 본 발명에 따라 특정 태스크를 수행하도록 구성될 수 있다. 소프트웨어 코드 또는 펌웨어 코드는 상이한 프로그래밍 언어 및 상이한 포맷 또는 스타일로 개발될 수 있다. 소프트웨어 코드는 또한 상이한 타겟 플랫폼을 위해 컴파일될 수 있다. 그러나, 본 발명에 따라 태스크를 수행하기 위한 소프트웨어 코드의 상이한 코드 포맷, 스타일 및 언어와 코드를 구성하는 기타 수단은 본 발명의 사상 및 범위에서 벗어나지 않을 것이다.

[0048]

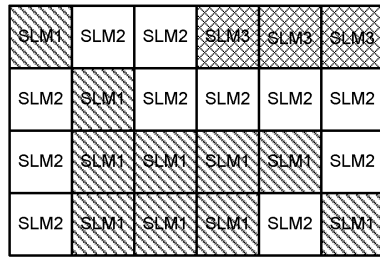
본 발명은 본 발명의 사상 및 기본 특성으로부터 벗어나지 않고서도 기타의 특정한 형태로 구현될 수 있다. 위에서 설명한 예는 그 모든 특징에 있어서 단지 예시를 위한 것으로 간주될 것이며, 본 발명을 제한하려는 것이 아니다. 따라서, 본 발명의 범위는 전술한 설명에 의해서가 아니라 첨부된 청구범위에 의해 정해져야 한다. 청구범위의 등가물의 의미 및 범위 내에서 이루어지는 모든 변경 또한 본 발명의 범위 내에 포함될 것이다.

**도면**

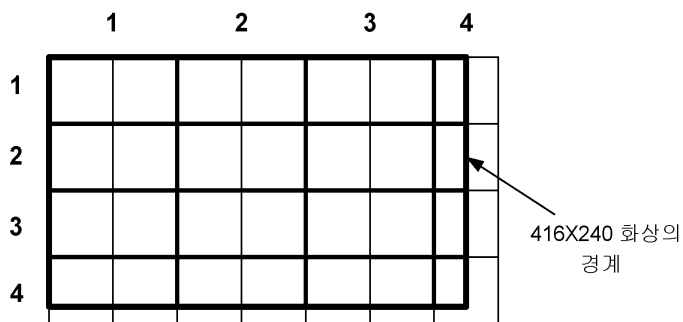
**도면1**



도면2



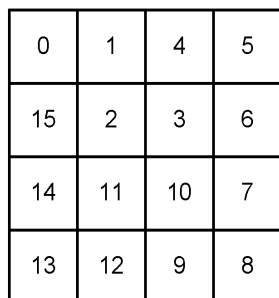
도면3



도면4

alf_param() {	C	디스크립터
alf_flag	2	u(1)ae(v)
if( alf_flag ) {		
<b>region_adaptation_flag</b>	2	u(1)ae(v)
... // original ALF parameters for luma		
alf_chroma_idc	2	u(2)ae(v)
... // original ALF parameters for chroma and luma CU on/off control		
}		
}		

도면5



도면6

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12

도면7

0	7	8	15
1	6	9	14
2	5	10	13
3	4	11	12

도면8

0	1	5	6
2	4	7	12
3	8	11	13
9	10	14	15

도면9

0	1	2	3
11	12	13	4
10	15	14	5
9	8	7	6

도면10

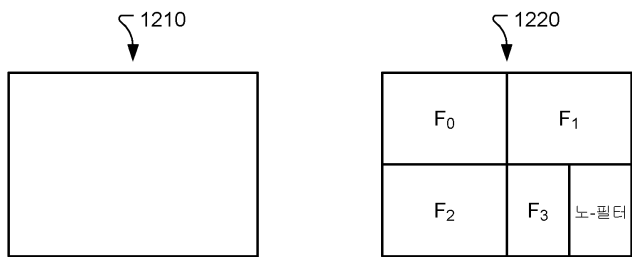
0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

도면11

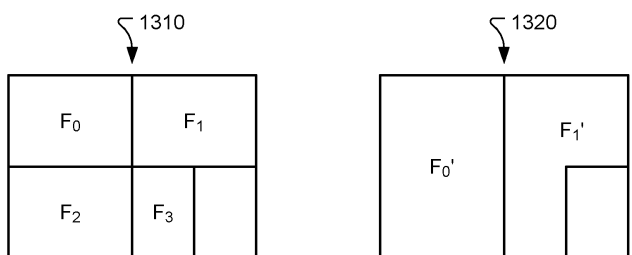
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



도면12



도면13



도면14

alf_fs_selection_param( r, c ) {	C	디스크립터
if( c == 0 && r == 0 ) {		
AlfFuFilterSetIdx[ r ][ c ] = 0		
} else {		
if( alf_enable_fu_merge ) {		
if( c == 0 ) {		
if( alf_merge_flag )	2	u(1)ac(v)
AlfFuFilterSetIdx[ r ][ c ] = AlfFuFilterSetIdx[ r - 1 ][ c ]		
else if ( AlfFsNum > 1 )		
AlfFuFilterSetIdx[ r ][ c ] = alf_fu_filter_set_idx	2	u(v)ac(v)
else		
AlfFuFilterSetIdx [ r ][ c ] = 0		
} else if ( r == 0 ) {		
if( alf_merge_flag )	2	u(1)ac(v)
AlfFuFilterSetIdx[ r ][ c ] = AlfFuFilterSetIdx[ r ][ c - 1 ]		
else if ( AlfFsNum > 1 )		
AlfFuFilterSetIdx[ r ][ c ] = alf_fu_filter_set_idx	2	u(v)ac(v)
else		
AlfFuFilterSetIdx [ r ][ c ] = 0		
} else {		
if( alf_merge_flag ) {	2	u(1)ac(v)
if( AlfFuFilterSetIdx[ r-1 ][ c ] == AlfFuFilterSetIdx[ r ][ c-1 ] ){		
AlfFuFilterSetIdx[ r ][ c ] = AlfFuFilterSetIdx[ r ][ c - 1 ]		
} else {		
if( alf_merge_up_flag )	2	u(1)ac(v)
AlfFuFilterSetIdx[ r ][ c ] = AlfFuFilterSetIdx[ r - 1 ][ c ]		
else		
AlfFuFilterSetIdx[ r ][ c ] = AlfFuFilterSetIdx[ r ][ c - 1 ]		
}		
} else {		
if ( AlfFsNum > 1 )		
AlfFuFilterSetIdx[ r ][ c ] = alf_fu_filter_set_idx	2	u(v)ac(v)
else		
AlfFuFilterSetIdx[ r ][ c ] = 0		
}		
}		
}		