(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년09월16일
(11) 등록번호 10-1438471
(24) 등록일자 2014년09월01일

(51) 국제특허분류(Int. Cl.)
H04N 19/00 (2014.01) *H04N 7/24* (2011.01)
(21) 출원번호 10-2013-7010079
(22) 출원일자(국제) 2011년12월31일
심사청구일자 2013년04월19일
(85) 번역문제출일자 2013년04월19일
(65) 공개번호 10-2013-0095279
(43) 공개일자 2013년08월27일
(86) 국제출원번호 PCT/CN2011/085170
(87) 국제공개번호 WO 2012/092841
국제공개일자 2012년07월12일
(30) 우선권주장
61/429,313 2011년01월03일 미국(US)
(뒷면에 계속)
(56) 선행기술조사문헌
KR1020100109459 A*
EP1841230 A
KR1020100039247 A
KR100668346 B1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
미디어텍 인크.
대만 신추 사이언스-베이스드 인더스트리얼 파크 더싱 로드 1 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 시앙샨 디스트릭트 종후아 로드 섹션 6 레인 647 앨리 31 넘버 115
(뒷면에 계속)
(74) 대리인
유미특허법인

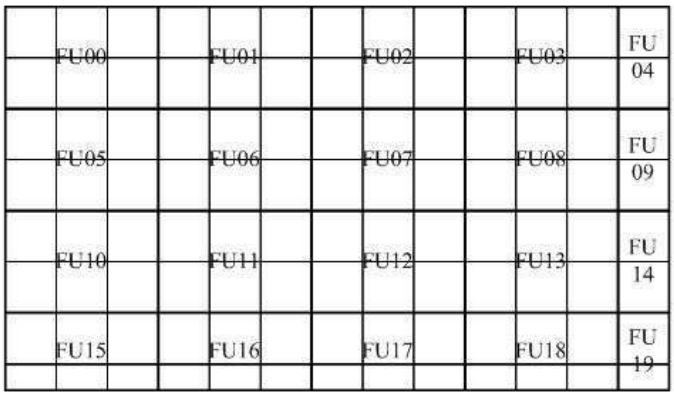
전체 청구항 수 : 총 22 항 심사관 : 조우연

(54) 발명의 명칭 **필터 유닛 기반 인-루프 필터링 방법**

(57) 요 약

비디오 디코더 및 인코더에서의 필터-유닛 기반 인-루프 필터링 방법이 개시된다. 일 실시예에서, 필터 파라미터가, 필터 인덱스를 기반으로 하는 각 필터에 대한 필터 파라미터 세트로부터 선택된다. 다른 실시예에서, 픽처가, 디폴트 사이즈와 다른 사이즈 사이에서 선택될 수 있는 필터 유닛 사이즈에 따라 필터 유닛으로 분할된다. 다른 사이즈가 선택되는 경우, 상기 필터 유닛 사이즈는 직접적인 사이즈 정보 또는 비율 정보를 이용하여 전달될 수 있다. 다른 실시예에서, 병합 플래그 및 병합 인덱스가 필터 유닛 병합 정보를 전달하기 위하여 사용된다. 컬러 비디오에 대한, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법이 개시된다. 일 실시예에서, 상기 방법은 상기 FU에 대한 컬러 성분 필터 신택스를 인터리빙(interleaving)하여, 필터 신택스를 상기 비디오 비트스트림에 포함시킨다.

대 표 도 - 도1a



(72) 발명자

챠이 시아-양

대만 뉴 타이페이 씨티 종헤 디스트릭트 민쉥 스트릿 레인 35 앨리 2 넘버 10 2층

황 유-웬

대만 타이페이 씨티 종샨 디스트릭트 롱지앙 로드 레인 298 넘버 23 8층

레이 쇼우-민

대만 흐신츄 카운티 쥬베이 씨티 리우지아 6쓰 스트릿 넘버 32 6층

(30) 우선권주장

61/498,949 2011년06월20일 미국(US)

61/503,870 2011년07월01일 미국(US)

61/549,931 2011년10월21일 미국(US)

특허청구의 범위

청구항 1

비디오 디코더에서의 필터-유닛(filter-unit) 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림(bitstream)을 수신하는 단계로서, FU(Filter Unit) 기반 인-루프 필터링(in-loop filtering)이 상기 압축된 비디오와 관련된 복원 루프에서 사용되는, 비디오 비트스트림 수신 단계;

상기 비디오 비트스트림으로부터 복원된 비디오를 추출하는 단계로서, 상기 복원된 비디오는 하나 또는 그 이상의 FU로 분할되는, 복원된 비디오 추출 단계;

상기 하나 또는 그 이상의 FU의 적어도 하나에 대하여 상기 비디오 비트스트림으로부터 필터 인덱스(filter index)를 결정하는 단계;

상기 하나 또는 그 이상의 FU의 상기 적어도 하나에 대한 상기 필터 인덱스에 따라 필터 파라미터 세트(filter parameter set)로부터 필터 파라미터(filter parameter)를 결정하는 단계; 및

상기 필터 파라미터를 기반으로 하여 상기 하나 또는 그 이상의 FU의 상기 적어도 하나에 인-루프 필터링을 적용하는 단계를 포함하고,

상기 필터 파라미터 세트와 관련된 정보가, 상기 하나 또는 그 이상의 FU로부터 분리된 상기 비디오 비트스트림의 더 높은 레벨에 포함되는,

비디오 디코더에서의 필터-유닛(filter-unit) 기반 인-루프 필터링 방법.

청구항 2

삭제

청구항 3

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림을 생성하는 단계로서, FU(Filter Unit) 기반 인-루프 필터링이 상기 압축된 비디오와 관련된 복원 루프에서 사용되고, 복원된 비디오가 FU 사이즈에 따라 FU로 분할되는, 비디오 비트스트림 생성 단계;

필터 파라미터 세트를 상기 비디오 비트스트림에 포함시키는 단계로서, 상기 필터 파라미터 세트는 상기 FU의 가능한 인-루프 필터링에 대하여 결정되는, 필터 파라미터 세트 포함 단계;

상기 FU의 적어도 하나에 대하여 필터 인덱스를 상기 비디오 비트스트림에 포함시키는 단계로서, 필터 파라미터가 상기 필터 인덱스에 따라 상기 FU의 상기 적어도 하나에 대한 상기 필터 파라미터 세트로부터 선택되는, 필터 인덱스 포함 단계; 및

상기 비디오 비트스트림을 제공하는 단계를 포함하고,

상기 필터 파라미터 세트와 관련된 정보가 상기 FU로부터 분리된 상기 비디오 비트스트림의 더 높은 레벨에 포함되는,

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 4

삭제

청구항 5

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림을 수신하는 단계로서, FU (Filter Unit) 기반 인-루프 필터링이 상기 압축된 비디오와 관련된 복원 루프에서 허용되는, 비디오 비트스트림 수신 단계;

상기 비디오 비트스트림으로부터 복워된 비디오를 추출하는 단계;

상기 비디오 비트스트림에서 FU 사이즈를 추출하는 단계;

상기 FU 사이즈에 따라 상기 복원된 비디오를 FU로 분할하는 단계; 및

상기 FU에 상기 인-루프 필터링을 적용하는 단계를 포함하는,

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 6

제5항에 있어서,

상기 FU사이즈를 포함시키기 위한 신택스 모드(syntax mode)를 나타내기 위하여, 제1 플래그(flag)가 상기 비디오 비트스트림에 포함되는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 7

제6항에 있어서,

상기 신택스 모드가 직접 모드(direct mode)이면, 상기 FU 사이즈가 상기 비디오 비트스트림 내의 직접적인 사이즈 정보에 따라 결정되는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 8

제7항에 있어서,

상기 직접적인 사이즈 정보는 FU 폭(width)과 FU 높이(height)를 포함하고, 상기 FU 폭과 상기 FU 높이는 LCU((Largest Coding Unit) 또는 블록(block)의 단위로 측정되는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 9

제6항에 있어서,

상기 신택스 모드가 비율 모드(ratio mode)이면, 상기 FU 사이즈가 상기 비디오 비트스트림 내의 FU 사이즈 비율과 최소 FU 사이즈에 따라 결정되는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 10

제6항에 있어서,

상기 제1 플래그는 제2 플래그에 의존하고, 상기 제2 플래그는 디폴트(default) FU 사이즈가 사용되는지 여부를 나타내고, 상기 제2 플래그가 나타내는 대로 상기 디폴트 FU 사이즈가 사용되면, 상기 제1 플래그가 상기 비디오 비트스트림에 포함되지 않는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 11

제10항에 있어서,

상기 신택스 모드가 비율 모드이면, 상기 FU 사이즈가 상기 비디오 비트스트림 내의 FU 사이즈 비율과 최소 FU 사이즈에 따라 결정되고, 상기 최소 FU 사이즈는 상기 디폴트 FU 사이즈와 동일한, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 12

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림을 생성하는 단계로서, FU((Filter Unit) 기반 인-루프 필터링이 상기 압축된 비디오와 관련된 복원 루프에서 사용되고, 복원된 비디오가 FU 사이즈에 따라 FU로 분할되는, 비디

오 비트스트림 생성 단계;

상기 비디오 비트스트림에 상기 FU 사이즈를 포함시키는 단계; 및

상기 비디오 비트스트림을 제공하는 단계를 포함하는,

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 13

제12항에 있어서,

상기 방법은, 제1 플래그를 상기 비디오 비트스트림에 포함시키는 단계를 더 포함하고,

상기 제1 플래그는 상기 FU 사이즈를 상기 비디오 비트스트림에 포함시키기 위한 신택스 모드를 나타내는, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 14

제13항에 있어서,

상기 방법은, 상기 신택스 모드가 직접 모드이면, 직접적인 사이즈 정보를 상기 비디오 비트스트림에 포함시키는 단계를 더 포함하고,

상기 직접적인 사이즈 정보는 상기 FU 사이즈를 나타내는, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 15

제14항에 있어서,

상기 직접적인 사이즈 정보는 FU 폭과 FU 높이를 포함하고, 상기 FU 폭과 상기 FU 높이는 LCU(Largest Coding Unit) 또는 블록 단위로 측정되는, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 16

제13항에 있어서,

상기 방법은, 상기 신택스 모드가 비율 모드이면, FU 사이즈 비율을 상기 비디오 비트스트림에 포함시키는 단계를 더 포함하고, 상기 FU 사이즈 비율은 상기 FU 사이즈와 최소 FU 사이즈의 비율인, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 17

제16항에 있어서,

상기 제1 플래그는 제2 플래그에 의존하고, 상기 제2 플래그는 디폴트 FU 사이즈가 사용되는지 여부를 나타내고, 상기 제2 플래그가 나타내는 대로 상기 디폴트 FU 사이즈가 사용되면, 상기 제1 플래그가 상기 비디오 비트스트림에 포함되지 않는, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 18

제17항에 있어서,

상기 신택스 모드가 비율 모드이면, 상기 FU 사이즈가 상기 비디오 비트스트림 내의 FU 사이즈 비율과 최소 FU 사이즈에 따라 결정되고, 상기 최소 FU 사이즈는 상기 디폴트 FU 사이즈와 동일한, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 19

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림을 수신하는 단계로서, FU (Filter Unit) 기반 인-루프 필터링이 상기 압축된 비디오와 관련된 복원 루프에서 허용되는, 비디오 비트스트림 수신 단계;

상기 비디오 비트스트림으로부터 복원된 비디오를 추출하는 단계로서, 복원된 비디오가 FU로 분할되는, 복원된 비디오 추출 단계;

상기 FU의 적어도 하나에 대하여 상기 비디오 비트스트림으로부터 병합(merge) 플래그를 추출하는 단계로서, 상기 병합 플래그가, 상기 FU의 상기 적어도 하나가 이웃하는 FU와 병합되는지 여부를 나타내는, 병합 플래그 추출 단계;

상기 병합 플래그가, 상기 FU의 상기 적어도 하나가 병합되고 복수의 병합 후보 FU가 존재하는 것을 나타내면, 상기 비디오 비트스트림으로부터 병합 인덱스를 수신하는 단계로서, 필터 파라미터가 상기 병합 인덱스에 의해 나타난 상기 이웃하는 FU의 필터 파라미터로부터 결정되는, 병합 인덱스 수신 단계;

상기 병합 플래그가 상기 FU의 상기 적어도 하나가 병합되지 않았음을 나타내면, 상기 비디오 비트스트림으로부터 상기 필터 파라미터를 수신하는 단계;

상기 필터 파라미터를 이용하여 상기 FU의 상기 적어도 하나에 상기 인-루프 필터링을 적용하는 단계; 및

상기 비디오 비트스트림으로부터 로우 반복 플래그(row-repeating flag)를 수신하는 단계

를 포함하고,

상기 로우 반복 플래그가 상기 FU의 로우가 동일한 필터 파라미터를 공유함을 나타내면, 상기 병합 플래그 추출 단계, 상기 병합 인덱스 수신 단계 및 상기 필터 파라미터를 수신하는 단계가 생략되는,

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 20

삭제

청구항 21

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 비디오에 대응하는 비디오 비트스트림을 생성하는 단계로서, FU(Filter Unit) 기반 인-루프 필터링이 상기 압축된 비디오와 관련된 복원 루프에서 사용되고, 복원된 비디오가 FU로 분할되는, 비디오 비트스트림 생성 단계;

상기 FU의 적어도 하나에 대해 병합 플래그를 상기 비디오 비트스트림에 포함시키는 단계로서, 상기 병합 플래그는 상기 FU의 상기 적어도 하나가 이웃하는 FU와 병합되는지 여부를 나타내는, 병합 플래그 포함 단계;

상기 병합 플래그가, 상기 FU의 상기 적어도 하나가 병합되고 복수의 병합 후보 FU가 존재하는 것을 나타내면, 병합 인덱스를 상기 비디오 비트스트림에 포함시키는 단계로서, 상기 FU의 상기 적어도 하나가 상기 병합 인덱스에 의해 나타난 상기 이웃하는 FU와 필터 파라미터를 공유하는, 병합 인덱스 포함 단계;

상기 병합 플래그가 상기 FU의 상기 적어도 하나가 병합되지 않았음을 나타내면, 상기 필터 파라미터를 상기 비디오 비트스트림에 포함시키는 단계;

상기 비디오 비트스트림을 제공하는 단계; 및

로우 반복 플래그를 상기 비디오 비트스트림에 포함시키는 단계

를 포함하고,

상기 로우 반복 플래그가 상기 FU의 로우가 동일한 필터 파라미터를 공유함을 나타내면, 상기 병합 플래그 포함 단계, 상기 병합 인덱스 포함 단계 및 상기 필터 파라미터를 상기 비디오 비트스트림에 포함시키는 단계가 생략되는,

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 22

삭제

청구항 23

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 컬러 비디오에 대응하는 비디오 비트스트림을 수신하는 단계로서, FU (Filter Unit) 기반 인-루프 필터링이 상기 압축된 컬러 비디오와 관련된 복원 루프에서 사용되고, 복원된 컬러 비디오가 FU로 분할되며, 각각의 상기 FU가 제1 컬러 성분과 제2 컬러 성분을 포함하는, 비디오 비트스트림 수신 단계;

상기 비디오 비트스트림 내의 상기 FU에 대하여 상기 제1 컬러 성분과 관련된 제1 필터 신택스 및 상기 제2 컬러 성분과 관련된 제2 필터 신택스를 수신하는 단계로서, 상기 FU에 대한 상기 제1 필터 신택스 및 제2 필터 신택스가 상기 비디오 비트스트림에서 인터리브되는(interleaved), 제1 필터 신택스 및 제2 필터 신택스 수신 단계;

상기 제1 필터 신택스에 따라 각각의 FU와 관련된 상기 제1 컬러 성분에 상기 인-루프 필터링을 적용하는 단계; 및

상기 제2 필터 신택스에 따라 각각의 FU와 관련된 상기 제2 컬러 성분에 상기 인-루프 필터링을 적용하는 단계를 포함하고,

상기 FU에 대한 상기 제1 필터 신택스 및 제2 필터 신택스가 하나의 FU 또는 복수의 FU를 기반으로 하여 인터리브되는,

비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 24

삭제

청구항 25

삭제

청구항 26

제23항에 있어서,

각각의 상기 FU가 제3 컬러 성분을 포함하고, 픽처-레벨 인-루프 필터링이 상기 제3 컬러 성분에 적용되며, 상기 제3 컬러 성분에 대한 제3 필터 신택스가 상기 비트스트림에 포함되는, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 27

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법으로서, 상기 방법은,

압축된 컬러 비디오에 대응하는 비디오 비트스트림을 생성하는 단계로서, FU (Filter Unit) 기반 인-루프 필터링이 상기 압축된 컬러 비디오와 관련된 복원 루프에 사용되고, 복원된 컬러 비디오가 FU로 분할되며, 각각의 상기 FU가 제1 컬러 성분 및 제2 컬러 성분을 포함하는, 비디오 비트스트림 생성 단계;

상기 비디오 비트스트림 내의 상기 FU에 대하여 상기 제1 컬러 성분과 관련된 제1 필터 신택스 및 상기 제2 컬러 성분과 관련된 제2 필터 신택스를 포함시키는 단계로서, 상기 FU에 대하여 상기 제1 필터 신택스 및 상기 제2 필터 신택스가 상기 비디오 비트스트림에서 인터리브되는, 제1 필터 신택스 및 제2 필터 신택스 포함 단계; 및

상기 비디오 비트스트림을 제공하는 단계를 포함하고,

상기 FU에 대하여 상기 제1 필터 신택스 및 제2 필터 신택스가 하나의 FU 또는 복수의 FU를 기반으로 하여 인터리브되는,

비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

청구항 28

삭제

청구항 29

삭제

청구항 30

제27항에 있어서,

각각의 상기 FU가 제3 컬러 성분을 포함하고, 픽처-레벨 인-루프 필터링이 상기 제3 컬러 성분에 적용되며, 상기 제3 컬러 성분에 대한 제3 필터 신택스가 상기 비트스트림에 포함되는, 비디오 인코더에서의 필터-유닛 기반 인-루프 필터링 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 2011년 1월 3일에 출원된 "미디어텍의 적응적 루프 필터(MediaTek's Adaptive Loop Filter)"라는 명칭의 미국 가특허 출원 제61/429,313호, 2011년 6월20일에 출원된 "샘플 적응적 오프셋을 위한 LCU 기반 신택스(LCU-based Syntax for Sample Adaptive Offset)"라는 명칭의 미국 가특허 출원 제61/498,949호, 2011년 7월 1일에 출원된, "샘플 적응적 오프셋을 위한 LCU 기반 신택스(LCU-based Syntax for Sample Adaptive Offset)"라는 명칭의 미국 가특허 출원 제61/503,870호, 2011년 10월 21일에 출원된 "적응적 루프 필터와 샘플 적응적 오프셋을 위한 필터 유닛 적응(Filter Unit Adaptation for Adaptive Loop Filter and Sample Adaptive Offset)"이라는 명칭의 미국 가특허 출원 제61/549,931호에 대한 우선권을 주장한다. 상기 미국 가특허 출원의 내용은 그 전체가 원용에 의해 본 명세서에 포함된다.

[0002] 본 발명은 비디오 프로세싱(video processing)에 관한 것이다. 특히, 본 발명은 샘플 적응적 오프셋(sample adaptive offset)과 적응적 루프 필터(adaptive loop filter)를 포함하는, 필터 유닛 기반 인-루프 필터링 장치 및 방법에 관한 것이다.

배 경 기 술

[0003] 비디오 코딩 시스템에서, 비디오 데이터는 예측(prediction), 변환(transform), 양자화(quantization) 및 디블로킹(deblocking)과 같은 다양한 프로세싱의 대상이다. 상기 비디오 코딩 시스템에서 프로세싱 경로를 따라, 노이즈가 유입될 수 있고, 비디오 데이터에 적용되는 오퍼레이션(operations)으로 인하여 프로세싱된 비디오 데이터의 특성(characteristics)이 오리지널(original) 비디오 데이터로부터 달라질 수 있다. 예를 들어, 처리된 비디오의 평균값(mean value)이 시프트(shift)될 수 있다. 강도 시프트(intensirty shift)는 시각적 결함(visual impairment) 또는 아티팩트(artifacts)를 야기할 수 있고, 이는 특히 강도 시프트가 픽처(picture)마다 가변할 경우 두드러지게 나타날 수 있다. 따라서, 픽셀(pixel) 강도 시프트는, 아티팩트를 감소시키기 위하여 신중하게 보상되거나 복원되어야 한다. 몇 가지 강도 오프셋 방식이 본 발명의 기술 분야에서 사용되어 왔다. 예를 들어, 샘플 적응적 오프셋(sample adaptive offset: SAO)으로 불리는 강도 오프셋 방식은, 선택된 컨텍스트(context)에 따라, 프로세싱된 비디오 데이터의 각 픽셀을 복수의 카테고리 중 어느 하나로 분류한다. SAO 방식은 일반적으로, 디코더가 적절하게 작동할 수 있도록, 픽처 또는 슬라이스(slice)를 블럭들로 분할하기 위한 파티션(partition) 정보와 각 블록의 SAO 오프셋 값과 같은, SAO 정보를 비디오 비트스트림(bitstream)에 포함시킬 것을 요구한다. SAO 외에, 적응적 루프 필터(adaptive loop filter: ALF)는, 비디오 품질을 향상시키기 위해 복원된 비디오에 흔히 적용되는 다른 종류의 인-루프 필터(in-loop filter)이다. 유사하게, 디코더가 적절하게 작동할 수 있도록, 파티션 정보와 필터 파라미터(parameters)와 같은 ALF 정보가 비디오 비트스트림에 포함되어야 한다.

[0004] 종래의 코딩 시스템에서, 픽처-레벨 인-루프 필터링(picture-level in-loop filtering)이 자주 적용되는데, 이 경우 동일한 필터 파라미터가 픽처의 모든 블럭에 의해 공유된다. 픽처 레벨 인-루프 필터링은 간단하지만, 픽처의 국부적인(local) 특성에 대한 적응성(adaptivity)이 부족하다. 픽처의 국부적인 특성에 적응할 수 있는 인-루프 필터링 방식을 개발하는 것이 요구된다. 국부적 적응성을 가지는 인-루프 필터링 방식은 필터 정보를 전달하기 위한 더 많은 대역폭을 필요로 할 수 있다. 따라서, 필터 정보를 효율적으로 및/또는 융통성 있게 전달할 수 있는 신택스(syntax)를 개발하는 것이 요구된다.

발명의 내용

[0005] 비디오 디코더 및 인코더에서의 필터-유닛 기반 인-루프 필터링 방법이 개시된다. 본 발명에 따른 일 실시예를 포함하는 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법은 압축된 비디오에 대응하는 비디오 비트스트림(bitstream)을 수신하는 단계로서, FU(Filter Unit) 기반 인-루프 필터링(in-loop filtering)이 상기 압축된 비디오와 관련된 복원 루프에 사용되는, 비디오 비트스트림 수신 단계; 상기 비디오 비트스트림으로부터 복원된 비디오를 추출하는 단계로서, 상기 복원된 비디오는 하나 또는 그 이상의 FU로 분할되는, 복원된 비디오 추출 단계; 각각의 상기 하나 또는 그 이상의 FU에 대하여 상기 비디오 비트스트림으로부터 필터 인덱스(filter index)를 결정하는 단계; 상기 각각의 상기 하나 또는 그 이상의 FU에 대한 상기 필터 인덱스에 따라 필터 파라미터 세트(filter parameter set)로부터 필터 파라미터(filter parameter)를 결정하는 단계; 및 상기 필터 파라미터를 기반으로 하여 상기 각각의 상기 하나 또는 그 이상의 FU에 인-루프 필터링을 적용하는 단계를 포함한다. 대응하는 인코더에서, 상기 방법은 각각의 신택스 정보를 상기 비디오 비트스트림에 포함시킨다.

[0006] 본 발명에 따른 다른 실시예에 있어서, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법은, 상기 비디오 비트스트림으로부터 FU 사이즈를 추출하는 단계; 상기 FU 사이즈에 따라 상기 복원된 비디오를 FU로 분할하는 단계; 및 상기 FU에 상기 인-루프 필터링을 적용하는 단계를 포함한다. FU 사이즈를 상기 비디오 비트스트림에 포함시키기 위한 신택스 모드(syntax mode)를 나타내기 위하여, 제1 플래그(flag)가 상기 비디오 비트스트림에 포함될 수 있다. 상기 신택스 모드가 직접 모드(direct mode)이면, 상기 FU 사이즈는 상기 비디오 비트스트림에 포함된 직접적인 사이즈 정보에 따라 결정된다. 상기 신택스 모드가 비율 모드(ratio mode)이면, 상기 FU 사이즈는 상기 비디오 비트스트림 내의 상기 FU 사이즈 비율과 최소 FU 사이즈에 따라 결정된다. 대응하는 인코더에서, 상기 방법은 각각의 신택스 정보를 상기 비디오 비트스트림에 포함시킨다.

[0007] 본 발명에 따른 또 다른 실시예에 있어서, 비디오 디코더에서의 필터-유닛 기반 인-루프 필터링 방법은, 각각의 상기 FU에 대하여 상기 비디오 비트스트림으로부터 병합(merge) 플래그를 추출하는 단계로서, 상기 병합 플래그가, 상기 각각의 상기 FU가 이웃하는 FU와 병합되는지 여부를 나타내는, 병합 플래그 추출 단계; 상기 병합 플래그가 상기 각각의 상기 FU가 병합되는 것을 나타내면, 상기 비디오 비트스트림으로부터 병합 인덱스를 수신하는 단계로서, 필터 파라미터가 상기 병합 인덱스가 나타내는 상기 이웃하는 FU의 필터 파라미터로부터 결정되는, 병합 인덱스 수신 단계; 상기 병합 플래그가 상기 각각의 상기 FU가 병합되지 않은 것을 나타내면, 상기 비디오 비트스트림으로부터 상기 필터 파라미터를 수신하는 단계; 및 상기 필터 파라미터를 이용하여 상기 각각의 상기 FU에 상기 인-루프 필터링을 적용하는 단계를 포함한다. 대응하는 인코더에서, 상기 방법은 상기 비디오 비트스트림에 각각의 신택스 정보를 포함시킨다.

[0008] 컬러 비디오에 대한 비디오 디코더 및 인코더에서의 필터-유닛 기반 인-루프 필터링 방법이 또한 개시된다. 본 발명의 일 실시예에 따른 필터-유닛 기반 인-루프 필터링 방법은, 상기 FU에 대한 컬러 성분 필터 신택스를 인터리빙(interleaving)하여, 필터 신택스를 상기 비디오 비트스트림에 포함시킨다.

도면의 간단한 설명

[0009] 도 1a는 픽쳐 사이즈가 832×480이고, 필터 유닛 사이즈가 192×128인, 예시적인 필터 유닛 분할을 도시한다.

도 1b는 픽쳐 사이즈가 832×480이고, 필터 유닛 사이즈가 384×256인, 예시적인 필터 유닛 분할을 도시한다.

도 1c는 전체 픽쳐가 하나의 필터 유닛으로서 처리되는, 예시적인 필터 유닛 분할을 도시한다.

도 2는 본 발명의 일 실시예에 따른 *alf_param()*을 위한 예시적인 신택스 디자인을 도시한다.

도 3은 본 발명의 일 실시예에 따른 *alf_fu_size_param()*을 위한 예시적인 신택스 디자인을 도시한다.

도 4는 본 발명의 일 실시예에 따른 *sao_alf_unit()*을 위한 예시적인 신택스 디자인을 도시한다.

도 5는 본 발명의 다른 실시예에 따른 *sao_alf_unit()*을 위한 예시적인 신택스 디자인을 도시한다.

도 6은 본 발명의 일 실시예에 따른, 컬러 비디오를 위한 예시적인 인터리브된 FU 신택스를 도시한다.

도 7은 본 발명의 일 실시예에 따른, 컬러 비디오에 대한 컬러 성분의 픽쳐 기반 인-루프 필터링과 함께 예시적인 인터리브된 FU 신택스를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0010] HEVC(High-Efficiency Video Coding)에서, 비디오 품질을 향상시키기 위해서 몇 가지 인-루프 필터링 툴(in-loop filtering tools)이 포함되어 왔다. 예를 들어, 적응적 오프셋(Adaptive Offset: AO)으로 불리는 기술이 복원된 비디오의 오프셋을 보상하기 위하여 도입되었고, AO가 복원 루프 내에 적용된다. 오프셋 보상을 위한 방법 및 시스템은 2011년 6월 12일에 출원된, "비디오 코딩을 위한 샘플 적응적 오프셋 장치 및 방법(Apparatus and Method of Sample Adaptive Offset for Video Coding)"라는 명칭의 미국 정규 특허 출원 제13/158,427호에 개시된다. 상기 방법은 각 픽셀을 복수의 카테고리 중 어느 하나로 분류하고, 강도 시프트 보상 또는 복원을 각 픽셀의 상기 카테고리에 기초하여 프로세싱된 비디오 데이터에 적용한다. 위의 AO 기술은 샘플 적응적 오프셋(Sample Adaptive Offset: SAO)으로 불린다. SAO 외에, 적응적 루프 필터(Adaptive Loop Filter: ALF) 또한, 비디오 품질을 향상시키기 위해 HEVC에 도입되어 왔다. ALF는 상기 복원 루프 내에서, 복원된 비디오에 공간 필터(spatial filter)를 적용한다. SAO와 ALF는 모두 본 명세서에서 인-루프 필터링으로서 고려된다.

[0011] 인-루프 필터링이 용이하게 이루어지도록, 인-루프 필터링 프로세스와 관련된 정보가 비트스트림 신택스(bitstream syntax)에 포함되어야 할 것이다. 인-루프 필터링과 관련된 신택스는, 관련된 신택스의 비트 레이트(bit rate)를 감소시키기 위하여 공유할 다수의 블록(blocks)에 대해, 시퀀스(sequence) 레벨 또는 픽처(picture) 레벨과 같은 더 높은 레벨에 주로 포함된다. 높은 레벨에서 상기 인-루프 필터링 정보를 포함하는 것이 상기 관련된 신택스의 비트 레이트를 감소시킬 수는 있으나, 그러한 방법은 픽처 내에서의 국부적인(local) 변화에 빠르게 적응하지 못할 수 있다. 몇 가지 개선된 방식은 어느 정도의 국부적 적응성을 제공한다고 알려져 있다. 예를 들어, SAO는 픽처 또는 픽처 영역을 더 작은 블록으로 적응적으로 분할하기 위해서 쿼드트리(quadtree)를 이용할 수 있고, SAO는 각각의 리프 블록(leaf block)에 적용될 수 있다. 반면에, ALF는 국부화(localization)를 제공하기 위하여 그룹 플래그(group flag)를 이용할 수 있다.

[0012] 인-루프 필터링 프로세스가 픽처의 국부적 특성에 기초하여 실시될 수 있도록 하기 위하여 보다 동적인 적응(dynamic adaptation)을 제공할 수 있는 기술을 개발하는 것이 바람직하다. 본 발명에 따른 실시예는 픽처 레벨(picture-level) 신택스 대신 신택스를 기반으로 하는 필터 유닛(filter unit)을 이용한다. 필터 유닛(FU)은 인-루프 필터링 프로세스에 사용되는 프로세싱 유닛으로서, HEVC에서 사용되는 코딩 유닛(CU)와는 상이할 수 있다. 2011년 4월 25일에 출원된, "적응적 루프 필터링 방법 및 장치(Method and Apparatus of Adaptive Loop Filtering)"라는 명칭의 미국 정규 특허 출원 제13/093,068호에서, 각각의 FU가 필터 세트 중 어느 하나를 후보로 선택할 수 있는, FU 기반의 ALF가 개시되어 있다. 일 예에서, 픽처는 동일한 크기를 가지는 필터 유닛들로 분할될 수 있다. 픽처 사이즈가 FU 사이즈에 따라 분할될 수 없는 경우, 픽처의 로우/컬럼(row/column)의 끝에서의 몇 가지 FU는 더 작을 수 있다. 도 1a 내지 도 1c는 세 가지의 상이한 FU 사이즈에 대한 FU 분할(partition)의 예를 나타낸다. 이러한 예에서 사용된 픽처 사이즈는 832×480이다. 도 1a 내지 도 1c에서 각각의 작은 정사각형은 64×64 LCU(Largest Coding Unit)를 나타내고, 가장 아래에 있는 블록들은 64×32의 크기를 가진다. 픽처 사이즈와 FU 사이즈가 알려져 있다면, FU 분할은 그에 따라 결정될 수 있다. 예를 들어, 192×128의 FU 사이즈는, 도 1a에 도시된 바와 같이 20개의 FU를 만들 수 있으며, FU04, FU09 및 FU14는 64×128의 사이즈를 가지고, FU15부터 FU18은 192×96의 사이즈를 가지며, FU19는 64×96의 사이즈를 가진다. 도 1b는 FU가 384×256의 사이즈를 가지는 경우를 나타낸다. 그것은 여섯 개의 FU를 만들고, FU2는 64×256의 사이즈를 가지고, FU3 및 FU4는 384×224의 사이즈를 가지며, FU5는 64×224의 사이즈를 가진다. 도 1c는 FU가 전체 픽처를 커버(cover)하는 예를 나타낸다. 도 1a 내지 도 1c에서, 의도된 FU는 LCU에 의해 측정되었으나, 다른 블록 사이즈 또한 사용될 수 있다.

[0013] 픽처가 FU로 분할된 후에, 상기 인-루프 필터링 프로세스에서 픽처의 국부적 특성에 적응하기 위하여, 복수의 후보 필터로 구성된 필터 세트(필터 파라미터 세트로 불리기도 함)가 결정될 수 있다. 도 2는 ALF의 인-루프 필터 파라미터를 수용하는 *alf_param()*의 예시적인 신택스 디자인을 나타낸다. 부호 210의 신택스 섹션(syntax section)은, FU 분할이 **alf_fu_size_flag**가 나타내는 바와 같이 사용되면, 필터 세트의 후보 필터 개수(*AlfFsNum*)가 *AlfFsNum* = **alf_fs_num_minus1** + 1에 따라 결정되는 것을 나타내고, 여기서 **alf_fs_num_minus1**은 필터 세트의 후보 필터 개수와 관련된 비트스트림의 신택스 요소(syntax element)이다. 부호 220의 신택스 섹션은 각각의 후보 필터의 필터 파라미터가 결정되는 것을 나타낸다. 픽처 폭(picture width)에서 FU의 개수(*AlfFuNumW*)와, 픽처 높이(picture height)에서 FU의 개수(*AlfFuNumH*)는 부호 230의 신택스 섹션에서 결정된다. 또한, FU 분할이 ***alf_fu_size_flag***가 나타내는 바와 같이 가능하면, 부호 240의 신택스 섹션에 도시된 바와 같이, FU 병합(merge)이 허용되는지와 후보 필터가 각각의 FU에 대해 선택되는지를 나타내기 위하여, 플래그, **alf_enable_fu_merge**가 포함된다. 도 2에서의 신택스 디자인은, 각각의 필터 유닛의 비트스트림에 개

별적으로 필터 파라미터를 포함하는 대신 각각의 필터 유닛에 대해 필터 인덱스(filter index)를 사용하는 예를 나타낸다. 필터 인덱스는 명시적인 필터 파라미터보다 비트 레이트 면에서 더 효율적인 필터 정보 표현이다. 본 발명이 속하는 기술 분야의 당업자라면 도 2에 도시된 상기 신택스 디자인을 수정함으로써 본 발명을 실시할 수 있다.

[0014] FU 사이즈 정보는, 도 3에 도시된 바와 같이, *alf_fu_size_param( )*를 이용하여 포함될 수 있다. **alf_fu_size_flag**가 1의 값을 가지면, 그것은 FU 분할이 사용되는 것을 나타낸다. FU 분할이 사용되면, 디폴트(default) FU 사이즈 또는 다른 FU 사이즈 중 어느 것이 사용되었는지와 관련하여, FU 사이즈가 FU 사이즈 변경 플래그, **alf_change_fu_size**에 의해 표시될 수 있다. 디폴트 사이즈가 아닌 FU 사이즈가 사용되는 경우(즉, **alf_change_fu_size** = 1), 상기 FU 사이즈 정보를 포함하는 **alf_fu_size_syntax_ mode**가 나타내는 바와 같이, 예시적인 신택스는 두 개의 대체 가능한 방법을 수용한다. **alf_fu_size_syntax_ mode**가 1의 값을 가지면, FU 사이즈 정보는 부호 310의 신택스 섹션에 나타난 바와 같이 직접 포함되며, **alf_fu_ width_in_lcu_minus1**는 LCU 단위의 FU 폭(FU width)과 관련되고, **alf_fu_ height_in_lcu_minus1**는 LCU 단위의 FU 높이(FU height)와 관련된다. **alf_fu_size_syntax_mode**가 0의 값을 가지면, FU 사이즈 정보는 FU 사이즈 비율(ratio)을 이용하여 표현된다. 이러한 경우에, FU 사이즈는, 부호 320의 신택스 섹션에 도시된 바와 같이, 최소 FU 사이즈와 곱셈 인자(multiplication factor)(**alf_fu_size_ratio_minus2**)의 곱과 같다. 도 3의 예는 최소 FU 사이즈로서 디폴트 FU 사이즈를 사용한다. 신택스 요소, **alf_default_fu_width_in_lcu_ minus1**은 LCU 단위의 디폴트 FU 폭과 관련되고, 신택스 요소, **alf_default_fu_height_in_ lcu_ minus1**은 LCU 단위의 최소 FU 높이와 관련된다. 신택스 요소, **alf_change_fu_size**가 0의 값을 가지면, 부호 330의 신택스 섹션에 도시된 바와 같이, 디폴트 FU 사이즈가 사용된다. 신택스 요소, **alf_fu_size_flag**가 0의 값을 가지면, 그것은, 부호 340의 신택스 섹션에 도시된 바와 같이, FU 분할이 없고 FU 사이즈가 픽쳐 사이즈임을 나타낸다. 도 3의 신택스 디자인은, 신택스 모드 플래그(즉, **alf_fu_size_syntax_mode**)에 따라 직접 모드(direct mode) 또는 비율 모드(ratio mode)를 사용하여, FU 사이즈 정보를 상기 비트스트림 신택스에 포함시키는 예를 나타낸다. 본 발명이 속하는 기술 분야의 당업자라면 도 3에 도시된 상기 신택스 디자인을 수정함으로써 본 발명을 실시할 수 있다.

[0015] FU 병합이 허용되면, 유사한 특성을 가지는 이웃하는 FU들은, 코딩 효율을 향상시키기 위하여, 동일한 후보 필터를 공유할 수 있다. FU 병합 정보를 전달하는 일 예는 병합된 연속 FU의 개수를 전달하는 것인데, 상기 병합된 연속 FU의 개수를 런(*run*)이라고 한다. 2011년 12월 6일에 출원된, "루마 및 크로마 성분에 대한 샘플 적응적 오프셋 장치 및 방법(Apparatur and Method of Sample Adaptive Offset for Luma and Chroma Components)"이라는 명칭의 미국 정규 특허 출원 제13/311,953호에서, 런을 기반으로 한 FU 병합 표현이 개시되어 있다. FU 구조에 대한 예시적인 신택스 디자인, *sao_alf_unit(rx, ry, c)*이 도 4에 도시되어 있으며, 인-루프 필터링은 SAO 및 ALF를 포함하고, *rx* 및 *ry*는 각각 FU 수평 인덱스 및 수직 인덱스이며, *c*는 컬러 성분(즉, Y, Cb or Cr)을 나타낸다. 코딩 효율을 더 향상시키기 위하여, FU의 로우는 동일한 후보 필터를 공유할 수 있다. 비트 레이트를 유지하기 위해서, *ry*에서 상기 FU의 로우가 동일한 필터 정보를 가지는지 아닌지를 나타내는데 신택스 요소, **repeat_row_flag** [*c*] [*ry*]가 사용될 수 있다. 좌측 FU와 병합된 FU의 개수를 나타내는데 신택스 요소, **run_diff** [*c*] [*ry*] [*rx*]가 사용될 수 있다.

[0016] 도 4의 예시적인 신택스 디자인이, FU 기반 인-루프 필터링을 위한 필터 공유의 런 기반 표현을 나타내지만, 상기 필터 공유를 나타내기 위하여 다른 방법 또한 사용될 수 있다. 병합 플래그를 기반으로 하는 대체 가능한 신택스 디자인이 도 5에 도시되어 있다. (*rx, ry*)에서 컬러 성분 *c*를 가지는 FU가 그것의 이웃하는 블록 중 어느 하나와 병합되는지를 나타내기 위하여 신택스 요소, **merge_flag** [*c*] [*ry*] [*rx*]가 사용될 수 있다. 부호 510의 신택스 섹션에 나타난 바와 같이, **repeat_row_flag** [*c*] [*ry*]가 FU의 로우가 반복되지 않았음을 나타내는 경우에만, 신택스 요소, **merge_flag** [*c*] [*ry*] [*rx*]가 포함된다. 또한, 부호 510의 신택스 섹션에 따라, *rx*가 0이고 *ry*가 0이면 **merge_flag** [*c*] [*ry*] [rx]가 0이 된다. **merge_flag** [*c*] [*ry*] [*rx*]가 FU가 그것의 이웃하는 블록 중 어느 하나와 병합되는 것을 나타내면, 부호 520의 신택스 섹션에 도시된 바와 같이, FU가 이웃하는 블록들 중 어느 블록과 병합되었는지를 나타내기 위하여 **merge_index** [*c*] [*ry*] [*rx*]가 포함된다. **merge_flag** [*c*] [*ry*] [*rx*]가 FU가 병합되지 않은 것을 나타내는 경우, 부호 530의 신택스 섹션에 도시된 바와 같이, 필터 정보를 제공하기 위하여 신택스 요소, *sao_alf_param* (*rx, ry, c*)가 포함될 것이다. 도 5의 신택스 디자인은 현재의 FU가 병합되었는지 여부와 현재의 FU가 이웃하는 FU들 중 어느 FU와 병합되었는지를 보여주는 병합 인덱스(merge index)를 나타내기 위하여, 병합 플래그를 포함시키는 예를 나타낸다. 본 발명이 속하는 기술 분야의 당업자라면 도 5에 도시된 상기 신택스 디자인을 수정함으로써 본 발명을 실시할 수 있다.

[0017] 상기 픽쳐 레벨 인-루프 필터링 신택스가 사용되면, 신택스는 언제나 상기 픽쳐 또는 더 높은 레벨에 포함된다. 컬러 비디오의 경우, 픽쳐가 각각의 컬러 성분에 대해 개별적인 인-루프 필터링 정보를 사용하도록 하기 위하여, 또는 색차(chrominance: chroma) 성분이 휘도(luminance: luma) 성분과 인-루프 필터링 정보를 공유하도록 하기 위하여, 픽쳐 레벨 신택스가 사용될 수 있다. 컬러 비디오에 대한 FU 기반 인-루프 필터링의 신택스는, 전체 픽쳐에 대한 루마 성분과 관련된 정보와 그 뒤에 이어지는 전체 픽쳐에 대한 크로마 성분과 관련된 정보를 포함할 수 있다. 따라서, 크로마 FU에 대한 상기 인-루프 필터링 프로세스는, 픽쳐의 모든 루마 FU에 대한 정보가 수신될 때까지 기다려야 할 수 있다. 대기 시간(latency)을 감소시키기 위하여, 본 발명에 따른 일 실시예는, FU 기반 인-루프 필터링을 사용하는 컬러 비디오에 대하여 인터리브된 신택스(interleaved syntax)를 활용한다. 도 6은 컬러 비디오의 FU 기반 인-루프 필터링에 대한 인터리브된 신택스의 예시적인 예를 나타낸다. FU 기반 필터 신택스는 FU 순서에 따라 포함된다. 각각의 FU에 대하여, 컬러 성분에 대한 신택스는 차례로 포함된다. 예를 들어, FU0의 Y성분(610)과 관련된 신택스 다음에 Cb 성분(612)과 Cr성분(614)이 이어진다. FU0의 모든 컬러 성분에 대한 필터 신택스가 포함된 후에, FU1의 Y성분(620)이 포함되고, Cb성분(622)과 Cr성분(624)이 이어진다. 나머지 FU에 대한 신택스가 동일한 방식으로 포함될 수 있다. 도 6의 인터리브된 필터 신택스는 본 발명에 따른 예시적인 실시예를 나타낸다. 필터 신택스는 도 6의 모든 FU에 대하여 인터리브된다. 당업자라면, 본 발명의 사상을 벗어나지 않는 다른 배열을 이용하여 상기 발명을 실시할 수 있다. 예를 들어, 세 가지 컬러 성분에 대한 YCbCr 신택스 순서는 CbCrY 또는 YCrCb와 같이 수정될 수 있다. 도 6이 하나 하나의 FU에 대한 신택스 인터리빙의 예를 나타내지만, 본 발명의 실시예는 복수의 FU에 대한 필터 신택스 인터리빙을 수행할 수 있다. 예를 들어, 신택스 인터리빙은 두 개의 FU마다, 즉, Y FU0, Y FU1, Cb FU0, Cb FU1, Cr FU0, Cr FU1, Y FU2, Y FU3, Cb FU2, Cb FU3, Cr FU2, Cr FU3 등과 같이 수행될 수 있다.

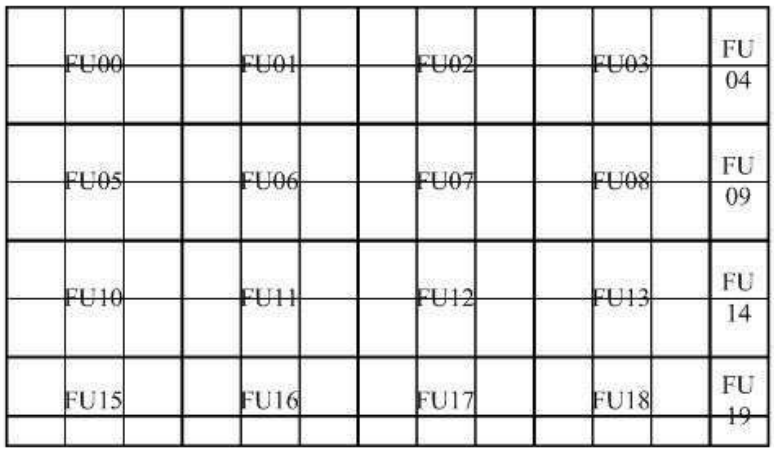
[0018] FU 기반 인-루프 필터링이 허용되면, 슬라이스 레벨(slice-level), 픽쳐 레벨 또는 시퀀스 레벨에서의 선택 플래그(selection flag)에 따라, FU 기반 또는 픽쳐 기반의 인-루프 필터링을 적응적으로 선택하기 위하여, 슬라이스 레벨, 픽쳐 레벨 또는 시퀀스 레벨에서의 플래그가 사용될 수 있다. 본 발명에 따른 일 실시예에 있어서, 선택 플래그는 컬러 비디오에 대한 필터 신택스 인터리빙과 함께 사용될 수 있다. 필터 신택스 인터리빙과 선택 플래그를 이용하는, 예시적인 구성이 도 7에 도시되어 있는데, Y 및 Cr 성분은 FU 기반 인-루프 필터링을 사용하지만, Cb 성분은 픽쳐 레벨 인-루프 필터링을 사용한다. 필터 신택스는, 도 6에서 언급된 경우와 유사하게 Y 및 Cr 성분에 대한 FU 순서에 따라 포함된다. Cb 픽쳐에 대한 필터 신택스는 Y 성분의 첫번째 FU에 대한 필터 신택스 다음에 삽입될 수 있다. 따라서, 도 7에 도시된 바와 같이, 필터 신택스 배열은 Y FU0(710), Cb 픽쳐(712), Cr FU0(714), Y FU1(720), Cr FU1(724), Y FU2(730) 및 Cr FU2(734)가 된다. 또는, Cb 픽쳐에 대한 필터 신택스는 Y FU0에 대한 필터 신택스 앞에 배치될 수 있다.

[0019] 본 발명에 따른 인-루프 필터링에 대한 신택스의 실시예는, 위에서 설명된 바와 같이, 다양한 하드웨어, 소프트웨어 코드, 또는 양자의 결합으로 구현될 수 있다. 예를 들어, 본 발명의 실시예는, 여기서 설명된 프로세싱을 수행하기 위하여, 비디오 압축 칩(video compression chip)에 통합된 회로이거나 또는 비디오 압축 소프트웨어(video compression software)에 통합된 프로그램 코드(program codes)일 수 있다. 본 발명의 실시예는 또한, 여기서 설명된 프로세싱을 수행하기 위하여, DSP(Digital Signal Processor) 상에서 실행되는 프로그램 코드일 수 있다. 본 발명은 또한, 컴퓨터 프로세서(computer processor), DSP, 마이크로프로세서(microprocessor), 또는 FPGA(field programmable gate array)에 의해서 수행되는 다수의 기능을 포함할 수 있다. 이러한 프로세서는, 본 발명에 의해 구현되는 상기 특정 방법을 규정하는 기계 판독 가능(machine readable) 소프트웨어 코드 또는 펌웨어(firmware) 코드를 실행함으로써, 본 발명에 따른 특정의 태스크(task)를 수행하도록 구현될 수 있다. 상기 소프트웨어 코드 또는 펌웨어 코드는 상이한 프로그래밍 언어 또는 상이한 포맷(format) 또는 형식(type)으로 개발될 수 있다. 상기 소프트웨어 코드는 또한, 상이한 타겟 플랫폼(target platform)에 대해 컴파일될 수 있다. 그러나, 본 발명에 따라 상기 태스크를 수행하기 위한, 소프트웨어 코드의 상이한 코드 포맷, 스타일 및 언어와 코드를 구성하는 다른 수단은 본 발명의 사상 및 범위를 벗어나지 않을 것이다.

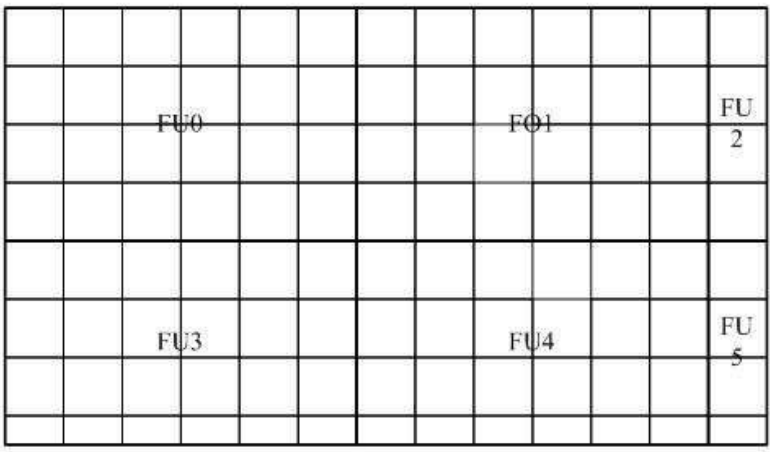
[0020] 본 발명은 그것의 사상 또는 본질적인 특성을 벗어나지 않는, 다른 특정 형태로 구현될 수 있다. 상기 설명된 예들은, 어디까지나 일 예로서 고려되어야 하고, 한정적인 것은 아니다. 상기 발명의 범위는, 따라서, 전술한 설명에 의해서가 아니라, 첨부된 청구항에 의해 정해진다. 상기 청구항의 의미 및 균등 범위 내에서 일어나는 모든 변화는 청구항의 범위 내에서 포함되어야 한다.

도면

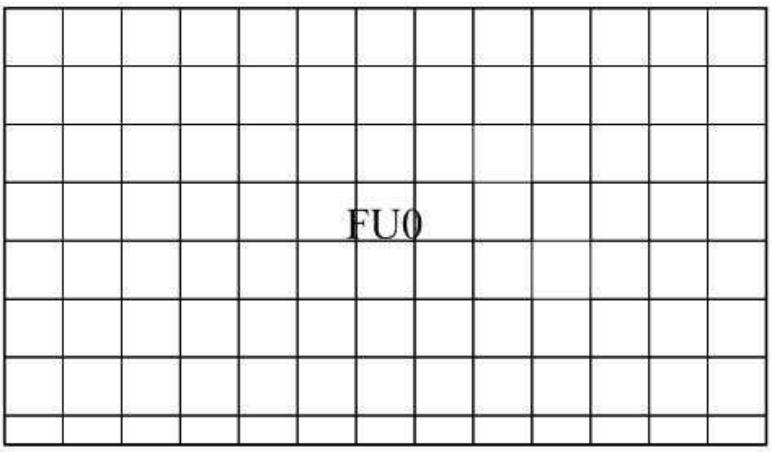
도면1a



도면1b



도면1c



도면2

alf_param( ) {	Descriptor	
alf_flag	u(1)|ae(v)	
if(alf_flag) {		
alf_fu_size_param( )		
if(alf_fu_size_flag)		210
AlfFsNum = **alf_fs_num_minus1** + 1	ue(v)|ae(v)	
else		
AlfFsNum = 1		
for(idx = 0; idx < AlfFsNum; idx ++)		220
alf_fs_param(idx)		
AlfFuNumW=(PicWidthInLcu+AlfFuWidthInLcu–1)/AlfFuWidthInLcu		230
AlfFuNumH=(PicHeightInLcu+AlfFuHeightInLcu–1)/AlfFuHeightInLcu		
if(alf_fu_size_flag) {		240
alf_enable_fu_merge	u(1)|ae(v)	
for(j = 0; j < AlfFuNumH; j++)		
for(i = 0; i < AlfFuNumW; i++)		
alf_fs_selection_param(j, i)		
}		
}		
}		

도면3

alf_fu_size_param( ) {	Descriptor	
if(**alf_fu_size_flag**) {	u (1)|ae(v)	
if(**alf_change_fu_size**) {	u(1)|ae(v)	
if(**alf_fu_size_syntax_mode**) {		
alf_fu_width_in_lcu_minus1	ue(v)|ae(v)	310
alf_fu_height_in_lcu_minus1	ue(v)|ae(v)	
AlfFuWidthInLcu = alf_fu_width_in_lcu_minus1 + 1		
AlfFuHeightInLcu = alf_fu_height_in_lcu_minus1 + 1		
} else {		
alf_fu_size_ratio_minus2	ue(v)|ae(v)	320
AlfRatio = alf_fu_size_ratio_minus2 + 2		
AlfFuWidthInLcu = (alf_default_fu_width_in_lcu_minus1+1*)AlfRatio		
AlfFuHeightInLcu = (alf_default_fu_height_in_lcu_minus1+1)*AlfRatio		
}		
} else {		
AlfFuWidthInLcu = alf_default_fu_width_in_lcu_minus1 + 1		330
AlfFuHeightInLcu = alf_default_fu_height_in_lcu_minus1 + 1		
}		
} else {		
AlfFuWidthInLcu = PicWidthInLcu		340
AlfFuHeightInLcu = PicHeightInLcu		
}		
}		

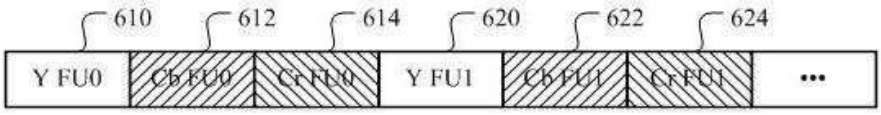
도면4

sao_alf_unit (rx, ry, c) {	Descriptor
if (rx == 0 && ry > 0) {	
repeat_row_flag [c] [ry]	u(1)
} else if (rx == 0 && ry == 0) {	
repeat_row_flag [c] [ry] = 0	
}	
if (!repeat_row_flag [c] [ry]) {	
if (rx == 0 || run [c] [ry] [rx] < 0) {	
if (ry == 0) {	
run [c] [ry] [rx] = 0 + **run_diff** [c] [ry] [rx]	
} else if (rx == NumLCUsInPicWidth - 1) {	
run [c] [ry] [rx] = 0	
} else {	
run [c] [ry] [rx] = run [c] [ry-1] [rx] + **run_diff** [c] [ry] [rx]	
}	
if (rx < NumLCUsInPicWidth - 1) {	
for (k = 0; k <= run [c] [ry] [rx]; k++) {	
run [c] [ry] [rx+k+1] = run [c] [ry] [rx+k] - 1	
}	
}	
if (ry > 0) {	
merge_up_flag [c] [ry] [rx]	u(1)
} else {	
merge_up_flag [c] [ry] [rx] = 0	
}	
if (!merge_up_flag [c] [ry] [rx]) {	
sao_alf_param (rx, ry, c)	
}	
}	
}	
}	

도면5

sao_alf_unit(rx, ry, c) {	Descriptor	
if (rx == 0 && ry > 0) {		
repeat_row_flag [c] [ry]	u(1)	
} else if (rx==0 && ry == 0) {		
repeat_row_flag [c] [ry] = 0		
}		
if (!repeat_row_flag [c] [ry]) {		510
if (rx > 0 || ry > 0) {		
merge_flag [c] [ry] [rx]	u(1)	
} else {		
merge_flag [c] [ry] [rx] = 0		
}		
if (merge_flag [c] [ry] [rx]) {		520
if (rx == 0) {		
merge_index [c] [ry] [rx] = 1 //upper		
} else if (ry == 0) {		
merge_index [c] [ry] [rx] = 0 //left		
} else if (upper_same_as_left ( )) {		
merge_index [c] [ry] [rx] = 0 //left		
} else {		
merge_index [c] [ry] [rx]	u(1)	
}		
else {		530
sao_alf_param (rx, ry, c)		
}		
}		
}		

도면6



도면7

