



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0049603
(43) 공개일자 2013년05월14일

(51) 국제특허분류(Int. Cl.)
H04N 7/26 (2006.01) H04N 7/32 (2006.01)
(21) 출원번호 10-2011-0114713
(22) 출원일자 2011년11월04일
심사청구일자 없음

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50, 연세대학교 (신촌동)
(72) 발명자
유제윤
서울특별시 강남구 강남대로128길 27, 306호 (논현동)
김영조
경기도 부천시 소사구 경인로535번길 42, 미림홈타운 302호 (괴안동)
(74) 대리인
권혁수, 송윤희, 오세준

전체 청구항 수 : 총 1 항

(54) 발명의 명칭 HEVC 더블록킹 필터 병렬 처리 알고리즘 및 HEVC 더블록킹 필터에서 병렬 처리가 가능하게 만든 방법

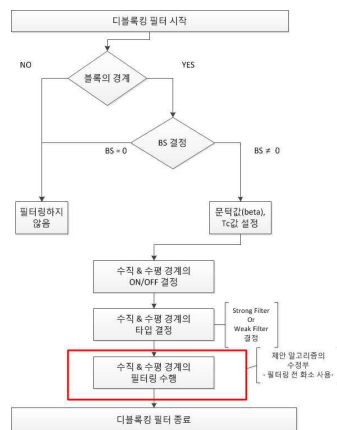
(57) 요약

본 발명은 비디오 코덱의 더블록킹 필터 성능 향상 및 속도 향상 기술에 관한 것이다.

본 발명의 일 실시예에 따르면 HEVC 더블록킹 필터에서 병렬 처리가 가능하게 만든 방법이 제공된다.

따라서, 본 발명의 구성에 따르면 연산시간을 병렬 처리로 인하여 최대 50%까지 줄일 수 있는 이점이 있다. 또한, 본 발명을 사용하면 더블록킹 필터의 필터링 과정에서 조금 더 나은 성능 이득을 가져온다. 병렬 처리가 불가능했던 구조에서 병렬 처리에 대한 확장성을 가져 다양한 방식으로 이용이 가능하다. 또한, 본 발명의 구성에 따르면 병렬 처리에 대한 확장성을 가져 다양한 방식으로 이용이 가능하다.

대표도 - 도7



(72) 발명자

변주원

서울특별시 성북구 종암2동 sk아파트 104동 1108호

김재석

서울특별시 구로구 경인로65길 16-15, 1104동 502호 (신도림동, 신도림4차e-편한세상)

이 발명을 지원한 국가연구개발사업

과제고유번호 1415107372

부처명 지식경제부

연구사업명 고해상도 영상정보 수집용 Gbps급 초고속 초저전력 무선통신 SoC 연구

연구과제명 산업원천기술개발

주관기관 연세대학교 산학협력단

연구기간 2011.03.01 ~ 2012.02.29

특허청구의 범위

청구항 1

HEVC 디블록킹 필터에서 병렬 처리가 가능하게 만든 방법.

명세서

기술분야

- [0001] 본 발명은 비디오 코덱의 디블록킹 필터 성능 향상 및 속도 향상 기술에 관한 것이다.
- [0002] 본 발명은 비디오 코덱의 디블록킹 필터 방식에 관한 것이다. 이 방식은 고화질 영상의 처리시 더 빠른 속도를 요구하는 코덱에 적용될 수 있다.

배경기술

- [0003] 최근 HD(High Definition) 영상 및 UHD(Ultra High Definition) 영상과 같은 고해상도, 고품질을 받으실 영상에 대한 수요가 다양한 응용 분야에서 증가하고 있다. 영상 데이터가 고해상도, 고품질이 될수록 기존의 영상 데이터에 비해 상대적으로 데이터량이 증가하기 때문에 기존의 유무선 광대역 회선과 같은 매체를 이용하여 영상 데이터를 전송하거나 기존의 저장 매체를 이용해 저장하는 경우, 전송 비용과 저장 비용이 증가하게 된다. 영상 데이터가 고해상도, 고품질화 됨에 따라 발생하는 이러한 문제들을 해결하기 위해서는 고효율의 영상 압축 기술들이 활용될 수 있다.
- [0004] 영상 압축 기술로 현재 픽처의 이전 또는 이후 픽처로부터 현재 픽처에 포함된 화소값을 예측하는 화면 간 예측 기술, 현재 픽처 내의 화소 정보를 이용하여 현재 픽처에 포함된 화소값을 예측하는 화면 내 예측 기술, 출현 빈도가 높은 값에 짧은 부호를 할당하고 출현 빈도가 낮은 값에 긴 부호를 할당하는 엔트로피 부호화 기술 등 다양한 기술이 존재하고 이러한 영상 압축 기술을 이용해 영상 데이터를 효과적으로 압축하여 전송 또는 저장할 수 있다.

발명의 내용

해결하려는 과제

- [0005] 본 발명은 진술한 종래 기술의 문제점을 해결하기 위하여, 데이터 의존도를 제거하여 더 빠른 execution time을 위한 병렬 처리 방법을 제공하는데 그 목적이 있다.
- [0006] 세로 경계 필터링의 결과를 기다린 이후 가로 경계 필터링을 수행하게 되면 병렬 처리가 되지 않는다. 따라서, 쓰는 데이터와 읽는 데이터를 분리하여 데이터 의존도를 제거하여 병렬 처리가 가능할 수 있게 하는 것을 목적으로 한다.
- [0007] 본 발명은 또한 필터링 처리된 픽셀을 이용하지 않고 필터링 전 픽셀을 이용하여 경계의 특성을 더 잘 반영하여 화질 향상을 가져오는 것을 목적으로 한다.

과제의 해결 수단

- [0008] 본 발명의 일 실시예에 따르면 HEVC 디블록킹 필터에서 병렬 처리가 가능하게 만든 방법이 제공된다.

발명의 효과

- [0009] 본 발명을 사용하면 디블록킹 필터의 필터링 과정에서 조금 더 나은 성능 이득을 가져온다. 병렬 처리가 불가능

했던 구조에서 병렬 처리에 대한 확장성을 가져 다양한 방식으로 이용이 가능하다.

[0010] 본 발명의 구성에 따르면 병렬 처리에 대한 확장성을 가져 다양한 방식으로 이용이 가능하다.

도면의 간단한 설명

[0011] 도 1은 본 발명의 일 실시예에 따른 영상 부호화 장치를 나타내는 블록도이다.

도 2는 본 발명의 또 다른 실시예에 따른 영상 복호화 장치의 구성을 나타내는 블록도이다.

도 3은 8x8 블록 내의 데이터 의존도를 설명하기 위한 도면이다.

도 4는 8x8 블록 간의 데이터 의존도를 설명하기 위한 도면이다.

도 5는 디블록킹 필터의 동작을 설명하기 위한 도면이다.

도 6은 이웃한 블록 간 필터링의 동작 순서를 설명하기 위한 도면이다.

도 7은 제안하는 디블록킹 필터의 알고리즘을 나타낸 흐름도이다.

도 8은 제안하는 알고리즘의 동작 순서를 설명하기 위한 도면이다.

도 9는 두 개의 블록을 동시에 처리하는 동작을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하, 도면을 참조하여 본 발명의 실시 형태에 대하여 구체적으로 설명한다. 본 명세서의 실시예를 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 명세서의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0013] 어떤 구성 요소가 다른 구성 요소에 “연결되어” 있다거나 “접속되어” 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있으나, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 아울러, 본 발명에서 특정 구성을 “포함” 한다고 기술하는 내용은 해당 구성 이외의 구성을 배제하는 것이 아니며, 추가적인 구성이 본 발명의 실시 또는 본 발명의 기술적 사상의 범위에 포함될 수 있음을 의미한다.

[0014] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.

[0015] 또한 본 발명의 실시예에 나타나는 구성부들은 서로 다른 특징적인 기능들을 나타내기 위해 독립적으로 도시되는 것으로, 각 구성부들이 분리된 하드웨어나 하나의 소프트웨어 구성단위로 이루어짐을 의미하지 않는다. 즉, 각 구성부는 설명의 편의상 각각의 구성부로 나열하여 포함한 것으로 각 구성부 중 적어도 두 개의 구성부가 합쳐져 하나의 구성부로 이루어지거나, 하나의 구성부가 복수 개의 구성부로 나뉘어져 기능을 수행할 수 있고 이러한 각 구성부의 통합된 실시예 및 분리된 실시예도 본 발명의 본질에서 벗어나지 않는 한 본 발명의 권리범위에 포함된다.

[0016] 또한, 일부의 구성 요소는 본 발명에서 본질적인 기능을 수행하는 필수적인 구성 요소는 아니고 단지 성능을 향상시키기 위한 선택적 구성 요소일 수 있다. 본 발명은 단지 성능 향상을 위해 사용되는 구성 요소를 제외한 본 발명의 본질을 구현하는데 필수적인 구성부만을 포함하여 구현될 수 있고, 단지 성능 향상을 위해 사용되는 선택적 구성 요소를 제외한 필수 구성 요소만을 포함한 구조도 본 발명의 권리범위에 포함된다.

[0017] 도 1은 본 발명의 일 실시예에 따른 영상 부호화 장치를 나타내는 블록도이다.

[0018] 도 1을 참조하면, 상기 영상 부호화 장치(100)는 움직임 예측부(111), 움직임 보상부(112), 인트라 예측부(120), 스위치(115), 감산기(125), 변환부(130), 양자화부(140), 엔트로피 부호화부(150), 역양자화부(160), 역변환부(170), 가산기(175), 필터부(180) 및 참조영상 버퍼(190)를 포함한다.

[0019] 영상 부호화 장치(100)는 입력 영상에 대해 인트라(intra) 모드 또는 인터(inter) 모드로 부호화를 수행하고 비

트스트림을 출력한다. 이하 본 발명의 실시예에서는 인트라 예측은 화면 내 예측, 인터 예측은 화면 간 예측과 동일한 의미로 사용될 수 있다. 예측 단위에 대한 최적의 예측 방법을 결정하기 위해 예측 단위에 대해 화면 내 예측 방법 및 화면 간 예측 방법이 선택적으로 사용될 수 있다. 영상 부호화 장치(100)는 입력 영상의 원본 블록에 대한 예측 블록을 생성한 후, 원본 블록과 예측 블록의 차분을 부호화한다.

[0020] 화면 내 예측 모드인 경우, 인트라 예측부(120)(또는 화면 내 예측부도 동일한 의미를 가지는 용어로 사용될 수 있다.)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성한다.

[0021] 인트라 예측부(120)는 소정의 예측 단위에 대해 화면 내 예측 모드에 대한 SATD를 산출하고 산출된 제1 SATD값과 제2 SATD값 그리고 제1 SATD값에 해당하는 제1 화면 내 예측 모드를 산출하여 후보 화면 내 예측 모드 리스트에 저장하고 제1 SATD값과 상기 제2 SATD값의 차가 소정의 임계값 이하인지 여부를 판단하여 상기 소정의 예측 단위의 예측 블록을 생성할 수 있다. 또한, 제1 SATD값과 제2 SATD값의 차가 소정의 임계값보다 큰 경우, 제1 SATD값에 해당하는 화면 내 예측 모드를 현재 예측 단위의 최종 화면 내 예측 모드로 결정하고, 제1 SATD값과 제2 SATD값의 차가 소정의 임계값 이하인 경우, 추가의 화면 내 예측 모드를 후보 화면 내 예측 모드 리스트에 포함하여 후보 화면 내 예측 모드 리스트에 저장된 화면 내 예측 모드를 기초로 산출된 RD 비용을 이용하여 최종 화면 내 예측 모드를 결정할 수 있다.

[0022] 또한, 인트라 예측부(120)는 소정의 부호화 단위에 대한 SATD값이 임계 SATD값을 넘는지 여부를 판단하고 소정의 부호화 단위에 대한 SATD값이 임계 SATD값보다 작은 경우 부호화 단위를 추가로 분할하지 않고 소정의 부호화 단위에 대한 SATD값이 임계 SATD 값 이상인 경우 추가로 부호화 단위를 분할할 수 있다.

[0023] 인트라 예측부(120)의 동작에 대해서는 이하의 본 발명의 실시예 및 도 3내지 도 9에서 구체적으로 설명한다.

[0024] 화면 간 예측 모드인 경우, 움직임 예측부(111)는, 움직임 예측 과정에서 참조 영상 버퍼(190)에 저장되어 있는 참조 영상에서 입력 블록과 가장 매치가 잘 되는 영역을 찾아 움직임 벡터를 구한다. 움직임 보상부(112)는 움직임 벡터를 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성한다.

[0025] 감산기(125)는 입력 블록과 생성된 예측 블록의 차분에 의해 잔여 블록(residual block)을 생성한다. 변환부(130)는 잔여 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 출력한다. 그리고 양자화부(140)는 입력된 변환 계수를 양자화 파라미터에 따라 양자화하여 양자화된 계수(quantized coefficient)를 출력한다. 엔트로피 부호화부(150)는 입력된 양자화된 계수를 확률 분포에 따라 엔트로피 부호화하여 비트스트림(bit stream)을 출력한다.

[0026] HEVC는 인터 예측 부호화, 즉 화면 간 예측 부호화를 수행하므로, 현재 부호화된 영상은 참조 영상으로 사용되기 위해 복호화되어 저장될 필요가 있다. 따라서 양자화된 계수는 역양자화부(160)에서 역양자화되고 역변환부(170)에서 역변환된다. 역양자화, 역변환된 계수는 가산기(175)를 통해 예측 블록과 더해지고 복원 블록이 생성된다.

[0027] 복원 블록은 필터부(180)를 거치고, 필터부(180)는 디블록킹 필터(deblocking filter), SAO(Sample Adaptive Offset), ALF(Adaptive Loop Filter) 중 적어도 하나 이상을 복원 블록 또는 복원 픽처에 적용할 수 있다. 필터부(180)는 적응적 인루프(in-loop) 필터로 불릴 수도 있다. 디블록킹 필터는 블록 간의 경계에 생긴 블록 왜곡을 제거할 수 있다. SAO는 코딩 에러를 보상하기 위해 화소값에 적정 오프셋(offset) 값을 더해줄 수 있다. ALF는 복원된 영상과 원래의 영상을 비교한 값을 기초로 필터링을 수행할 수 있으며, 고효율이 적용되는 경우에만 수행될 수도 있다. 필터부(180)를 거친 복원 블록은 참조 영상 버퍼(190)에 저장된다.

[0028] 도 2는 본 발명의 또 다른 실시예에 따른 영상 복호화 장치의 구성을 나타내는 블록도이다.

[0029] 도 2를 참조하면, 상기 영상 복호화 장치(200)는 엔트로피 복호화부(210), 역양자화부(220), 역변환부(230), 인트라 예측부(240), 움직임 보상부(250), 필터부(260) 및 참조 영상 버퍼(270)를 포함한다.

[0030] 영상 복호화 장치(200)는 부호화기에서 출력된 비트스트림을 입력 받아 인트라 모드 또는 인터 모드로 복호화를 수행하고 재구성된 영상, 즉 복원 영상을 출력한다. 인트라 모드인 경우 화면 내 예측 모드를 사용하여 예측 블록을 생성하고 인터 모드인 경우 화면 간 예측 방법을 사용하여 예측 블록을 생성한다. 영상 복호화 장치(200)는 입력 받은 비트스트림으로부터 잔여 블록(residual block)을 얻고 예측 블록을 생성한 후 잔여 블록과 예측 블록을 더하여 재구성된 블록, 즉 복원 블록을 생성한다.

[0031] 엔트로피 복호화부(210)는 입력된 비트스트림을 확률 분포에 따라 엔트로피 복호화하여 양자화된 계수

(quantized coefficient)를 출력한다. 양자화된 계수는 역양자화부(220)에서 역양자화되고 역변환부(230)에서 역변환되며, 양자화된 계수가 역양자화/역변환 된 결과, 잔여 블록(residual block)이 생성된다.

[0032] 화면 내 예측 모드인 경우, 인트라 예측부(240)(또는 화면 간 예측부)는 현재 블록 주변의 이미 부호화된 블록의 화소값을 이용하여 공간적 예측을 수행하여 예측 블록을 생성한다.

[0033] 화면 간 예측 모드인 경우, 움직임 보상부(250)는 움직임 벡터 및 참조 영상 버퍼(270)에 저장되어 있는 참조 영상을 이용하여 움직임 보상을 수행함으로써 예측 블록을 생성한다.

[0034] 잔여 블록과 예측 블록은 가산기(255)를 통해 더해지고, 더해진 블록은 필터부(260)를 거친다. 필터부(260)는 디블록킹 필터, SAO, ALF 중 적어도 하나 이상을 복원 블록 또는 복원 픽처에 적용할 수 있다. 필터부(260)는 재구성된 영상, 즉 복원 영상을 출력한다. 복원 영상은 참조 영상 버퍼(270)에 저장되어 화면 간 예측에 사용될 수 있다.

[0035] 부호화/복호화 장치의 예측 성능을 향상시키기 위한 방법에는 보간(interpolation) 영상의 정확도를 높이는 방법과 차신호를 예측하는 방법이 있다. 여기서 차신호란 원본 영상과 예측 영상과의 차이를 나타내는 신호이다. 본 발명에서 “차신호”는 문맥에 따라 “차분 신호”, “잔여 블록” 또는 “차분 블록”으로 대체되어 사용될 수 있으며, 해당 기술분야에서 통상의 지식을 가진 자는 발명의 사상, 본질에 영향을 주지 않는 범위 내에서 이를 구분할 수 있을 것이다.

[0036] 전술한 바와 같이 이하, 본 발명의 실시예에서는 설명의 편의상 코딩 유닛(Coding Unit)을 부호화 단위라는 용어로 사용하지만, 부호화뿐만 아니라 복호화를 수행하는 단위가 될 수도 있다. 이하, 본 발명의 실시예에 따르면 도 3내지 도 9에서 설명하는 두개의 후보 인트라 예측 모드를 이용한 화면 내 예측 모드의 부/복호화 방법은 도 1 및 도 2에서 전술한 각 모듈의 기능에서 맞게 구현될 수 있고 이러한 부호화기 및 복호화기는 본 발명의 권리범위에 포함된다. 즉, 본 발명의 실시예에서 후술할 영상 부호화 방법 및 영상 복호화 방법은 도 1 및 도 2에서 전술한 영상 부호화기 및 영상 복호화기에 포함된 각 구성부에서 수행될 수 있다. 구성부의 의미는 하드웨어적인 의미뿐만 아니라 알고리즘을 통해 수행될 수 있는 소프트웨어적인 처리 단위도 포함할 수 있다.

[0037] 또한, 이하, 본 발명의 실시예에서 후술할 영상 부호화 방법 및 영상 복호화 방법은 도 1 및 도 2에서 전술한 영상 부호화기 및 영상 복호화기에 포함된 각 구성부에서 수행될 수 있다. 구성부의 의미는 하드웨어적인 의미뿐만 아니라 알고리즘을 통해 수행될 수 있는 소프트웨어적인 처리 단위도 포함할 수 있다.

[0038] 표준화가 진행되고 있는 HEVC의 디블록킹 필터는 병렬화가 이슈이다. 현재 Panasonic's Deblocking Filter는 세로 경계 필터링 이후 가로 경계 필터링의 방식을 취하고 있다. 도 3의 표시 부분에 데이터 의존도가 존재(세로 경계 필터링 결과를 기다렸다가 가로 경계를 수행)하므로 병렬화되지 않는 문제점이 있다. 즉, Panasonic's Deblocking Filter는 Data Dependency의 문제로 인하여 병렬화가 되지 않는 단점이 존재한다.

[0039] 본 발명의 구성은 필터링 전 픽셀로 이용하는 모든 input을 대체한다.

[0040] 본 발명에서 제안하는 병렬 디블록킹 필터 알고리즘은 다음과 같다.

[0041] 1. 8x8 블록에서의 데이터 의존도

[0042] 병렬화가 되지 않는 이유는 여러 과정들 사이에 존재하는 데이터 의존도이다. 도 4를 살펴보면 이전 8x8 블록의 수평 경계의 값이 필터링 이후에 쓰여져야 다음 8x8 블록의 수직 경계 필터링을 수행할 수 있다. 이 두 경계 사이에 데이터 의존도가 존재하는 것이다. Panasonic 및 Tandberg, Nokia, ericsson의 디블록킹 필터는 하나의 8x8 블록 내부에서도 수직 경계와 수평 경계 사이에 데이터 의존도가 존재한다. 데이터 의존도는 동시 필터링 수행을 할 수 없게 만들기 때문에 병렬화에 장애물이 된다. 도 4에서 보이듯 #n 8x8 블록과 #n 8x8 블록이 동시에 진행될 수 없게 되는 것이다.

[0043] 2. 필터링 되지 않은 화소의 재이용

[0044] Panasonic의 디블록킹 필터 알고리즘을 살펴보면 필터링 되지 않은 화소를 결정과정에서 이용함으로써 데이터 의존도를 제거하여 병렬화가 가능하다. 이를 필터링 과정에 이용하게 되면 필터링 과정에서도 데이터 의존도가 사라지게 된다. 데이터 의존도는 읽어야만 하는 데이터가 아직 쓰여지지 않았을 때 발생하게 된다. 새로운 데이터를 읽지 않고 이전의 데이터를 읽게 되면 원하는 결과가 나오지 않을 가능성이 있기 때문이다. Panasonic의 디블록킹 필터는 결정과정 (ON/OFF, 타입)에서는 필터링 전 화소를 이용하지만 필터링 과정에서는 이전 블록의 필터링 값의 결과가 쓰여진 이후에 그 결과값을 새로운 인풋으로 받아 다음 과정을 전개한다. 그리하여 데이터

의존도가 존재하게 되고, 이는 병렬 처리를 불가능하게 만드는 요인이 된다. 이를 해결하기 위하여 필터링 되지 않은 화소를 결정과정 이외에 필터링 과정에서도 이용한다.

[0045] 도 5는 필터링 과정에서 필터링 전 화소를 이용했을 때 데이터 처리 순서이다. 도 5의 (1)은 데이터를 읽어 들여 ON/OFF 결정을 하고, (2)는 데이터를 읽어 들여 타입 결정을 한다. Panasonic 알고리즘에서 개선된 부분이기도 5의 (3), (4) 인데 두 과정 모두 필터링 전 화소를 이용하게 된다. 필터링 전 화소는 라인 버퍼에 저장되어 있으며 저장되어 있는 화소를 읽어 필터링을 수행한다. 필터링을 수행한 이후 참조 프레임에 데이터가 쓰여지게 된다. (3)과 (4)는 한클럭의 차이만 두게 되면 서로 충돌없이 데이터가 덮어 쓰여지게 된다. 이전 알고리즘에서는 (3)의 결과를 이용하여 (4)를 처리하여 필터링 수행시간만큼의 딜레이가 발생하지만 필터링 전 화소를 사용하게 되면 한클럭 차이로 딜레이를 줄일 수 있게 된다.

[0046] 또한, 이웃한 블록 사이의 데이터 의존도 역시 사라지게 된다. 도 6에서 보이듯 데이터가 쓰여지는 타이밍만 차이나게 조절해 주게 되면 $\#n$ 블록과 $\#n+1$ 블록을 동시에 수행할 수 있게 된다. 도 6에서 짚게 표시된 화소들은 주변의 화소들을 4개 ~ 6개씩 이용하여 새로운 값을 만들고 그 값을 다시 메모리에 쓰면 필터링이 완료가 된다. 메모리에 쓰는 타이밍을 겹치지 않게 설계하면 도 6의 동작 (1)과 동작 (2)는 한 클럭의 시간 차이로 필터링 수행 후 메모리에 쓰는 동작까지 완료가 가능하다.

[0047] 3. 새로운 디블록킹 필터

[0048] 필터링 전 화소를 결정과정 및 필터링 수행 과정에 이용하게 되면 블록간의 데이터 의존도를 제거함으로써 병렬 처리가 가능하다. 다음, 도 7는 제안하는 디블록킹 필터의 알고리즘이다.

[0049] 새로운 디블록킹 필터는 도 7의 과정과 같이 진행이 된다. Panasonic 알고리즘은 ON/OFF 결정과 타입 결정은 두 경계를 동시에 진행하지만 필터링 수행 과정에서는 전술한 데이터 의존도의 문제로 인하여 수직 경계와 수평 경계를 순차적으로 수행해야 한다. 하지만 제안하는 알고리즘은 위에서 설명한 것과 같이 필터링 전 화소를 사용하여 데이터 의존도를 제거함으로써 수직 경계와 수평 경계를 동시에 필터링 수행할 수 있다.

[0050] 도 8은 한 블록 내에서 각각의 과정을 처리할 때 이용하고 적용되는 영역을 표시한 것이다. 도 8의 (1)은 ON/OFF 결정을 하기 위해서 3, 6번째 라인에서의 화소를 이용하여 판단하고, (2)는 전 라인의 화소를 이용하여 각각의 라인의 필터링 타입 - strong or weak -를 결정한다. 각각의 필터링이 결정이 되면 도 8의 (3)의 과정이 시작되고 바로 (4)의 과정이 시작된다. "2. 필터링 되지 않은 화소의 재이용"에서 설명하였듯 필터링 전 화소를 사용하면 데이터 의존도가 사라져 결과를 기다리지 않고 다음 과정을 진행할 수 있게 된다.

[0051] 도 9은 두 개의 블록을 동시에 처리하는 동작을 표시한 도면이다. 도 9의 첫 단계는 $\#n$ 블록과 이웃하는 블록의 수직, 수평 경계의 ON/OFF 결정을 하게 된다. 두 번째 단계는 $\#n$ 블록과 이웃하는 블록은 동시에 타입 결정을 하게 되고, $\#n+2$ 블록이 새롭게 ON/OFF 결정단계가 시작된다. 마지막 단계는 $\#n$ 블록은 필터링을 수행하고, $\#n+2$ 블록은 타입 결정을 하게 된다. 데이터간의 의존도가 없기 때문에 도 9과 같이 연속적으로 디블록킹 필터링이 가능하다.

[0052] Panasonic의 알고리즘과는 다르게 제안하는 알고리즘의 가장 큰 특성은 필터링 전 화소를 모든 단계에서 이용한다는 것이다. 표 1는 기존의 알고리즘에서 이용하는 화소와 제안된 알고리즘에서 이용하는 화소의 차이를 보여준다.

[0053] [표 1] 알고리즘의 input 비교

	Tandberg, Nokia, Ericsson 알고리즘	Panasonic 알고리즘	제안 알고리즘
ON/OFF 결정	필터링 후 화소	필터링 전 화소	필터링 전 화소
타입 결정	필터링 후 화소	필터링 전 화소	필터링 전 화소
필터링 수행	필터링 후 화소	필터링 후 화소	필터링 전 화소

[0054]

[0055] 표 1을 보면, 레퍼런스 알고리즘인 Panasonic 알고리즘과 비교하였을 때 제안 알고리즘은 필터링 수행 단계에서 필터링 전 화소를 사용하는 것이 차이점이다. 또한 각각의 단계에서 사용하는 수식식은 다음과 같다.

$p3_0$	$p2_0$	$p1_0$	$p0_0$	$q0_0$	$q1_0$	$q2_0$	$q3_0$
$p3_1$	$p2_1$	$p1_1$	$p0_1$	$q0_1$	$q1_1$	$q2_1$	$q3_1$
$p3_2$	$p2_2$	$p1_2$	$p0_2$	$q0_2$	$q1_2$	$q2_2$	$q3_2$
$p3_3$	$p2_3$	$p1_3$	$p0_3$	$q0_3$	$q1_3$	$q2_3$	$q3_3$
$p3_4$	$p2_4$	$p1_4$	$p0_4$	$q0_4$	$q1_4$	$q2_4$	$q3_4$
$p3_5$	$p2_5$	$p1_5$	$p0_5$	$q0_5$	$q1_5$	$q2_5$	$q3_5$
$p3_6$	$p2_6$	$p1_6$	$p0_6$	$q0_6$	$q1_6$	$q2_6$	$q3_6$
$p3_7$	$p2_7$	$p1_7$	$p0_7$	$q0_7$	$q1_7$	$q2_7$	$q3_7$

[0056]

[0057] <8x8 블록의 경계와 화소>

수학식 1

$$d = |p2_2 - 2 * p1_2 + p0_2| + |q2_2 - 2 * q1_2 + q0_2| + |p2_5 - 2 * p1_5 + p0_5| + |q2_5 - 2 * q1_5 + q0_5|$$

[0058]

수학식 2

$$d < \beta$$

[0059]

[0060] 수학식 1과 수학식 2는 ON/OFF 결정 단계이다. 수학식 2를 만족시키면 다음 단계로 진행된다.

수학식 3

$$d < (\beta \gg 2)$$

[0061]

수학식 4

$$|p3_i - p0_i| + |q0_i - q3_i| < (\beta \ll 3)$$

[0062]

수학식 5

$$|p0_i - q0_i| < ((5 * t_c + 1) \gg 1)$$

[0063]

[0064] 수학식 3 내지 수학식 5를 모두 만족시킬 경우 strong filter를, 하나라도 만족시키지 못할 경우 weak filter를 수행한다. 타입 결정단계에서 strong filter 결정 시 다음의 수학식 6 내지 수학식 11을 이용하여 필터링을 수행한다.

수학식 6

$$p0'_i = Clip((p2_i + 2 * p1_i + 2 * p0_i + 2 * q0_i + q1_i + 4) \gg 3)$$

[0065]

수학식 7

$$[0066] \quad q0_i' = Clip((p1_i + 2 * p0_i + 2 * q0_i + 2 * q1_i + q2_i + 4) \gg 3)$$

수학식 8

$$[0067] \quad p1_i' = Clip((p2_i + p1_i + p0_i + q0_i + 2) \gg 2)$$

수학식 9

$$[0068] \quad q1_i' = Clip((p0_i + q0_i + q1_i + q2_i + 2) \gg 2)$$

수학식 10

$$[0069] \quad p2_i' = Clip((2 * p3_i + 3p2_i + p1_i + p0_i + q0_i + 4) \gg 3)$$

수학식 11

$$[0070] \quad q2_i' = Clip((p0_i + q0_i + q1_i + q2_i + 2 * q3_i + 4) \gg 3)$$

[0071] 수학식 6 내지 수학식 11은 strong filter의 동작을 표현한 것이다. 각각의 계산 이후 클리핑을 거친 이후에 경계로부터 양쪽 3개의 화소를 바꾸게 된다. 타입 결정에서 weak filter로 결정 시 다음의 수학식 12 내지 수학식 16을 이용하여 필터링을 수행한다.

수학식 12

$$[0072] \quad \Delta = Clip(-t_c, t_c, (13 * (q0_i - p0_i) + 4 * (q1_i - p1_i) - 5 * (q1_i - p2_i) + 16) \gg 5)$$

수학식 13

$$[0073] \quad p0_i' = Clip(p0_i + \Delta)$$

수학식 14

$$[0074] \quad q0_i' = Clip(q0_i + \Delta)$$

수학식 15

[0075]
$$p1_i' = Clip(p1_i + \Delta/2)$$

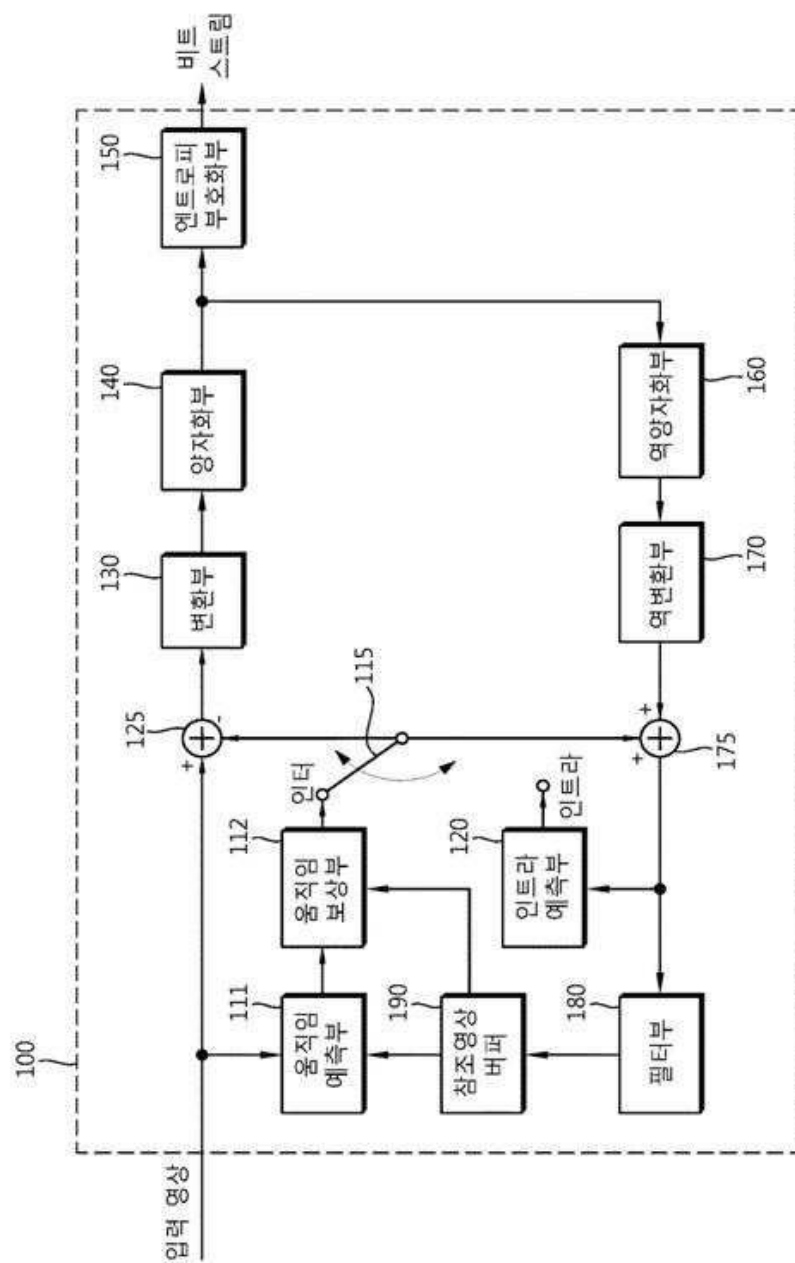
수학식 16

[0076]
$$q1_i' = Clip(q1_i + \Delta/2)$$

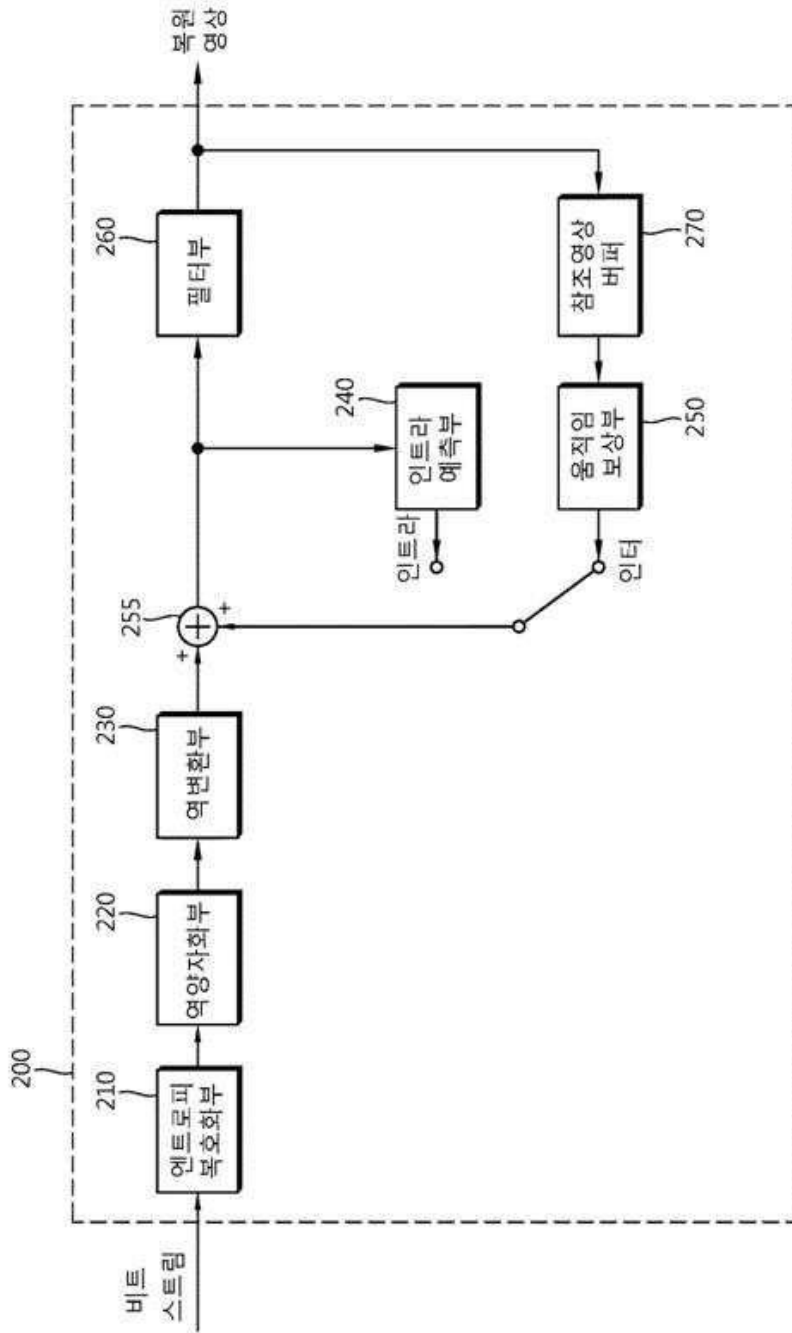
[0077] 수학식 12 내지 수학식 16은 weak filter의 동작을 표현한 것이다. 경계로부터 양쪽 두 개의 화소를 대체하게 된다.

도면

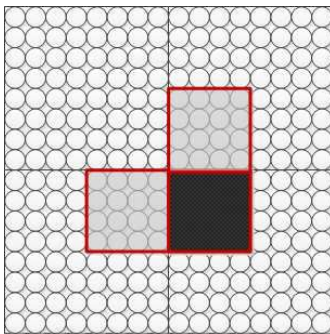
도면1



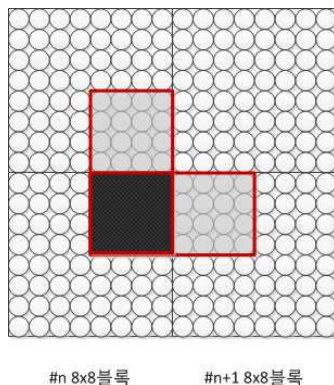
도면2



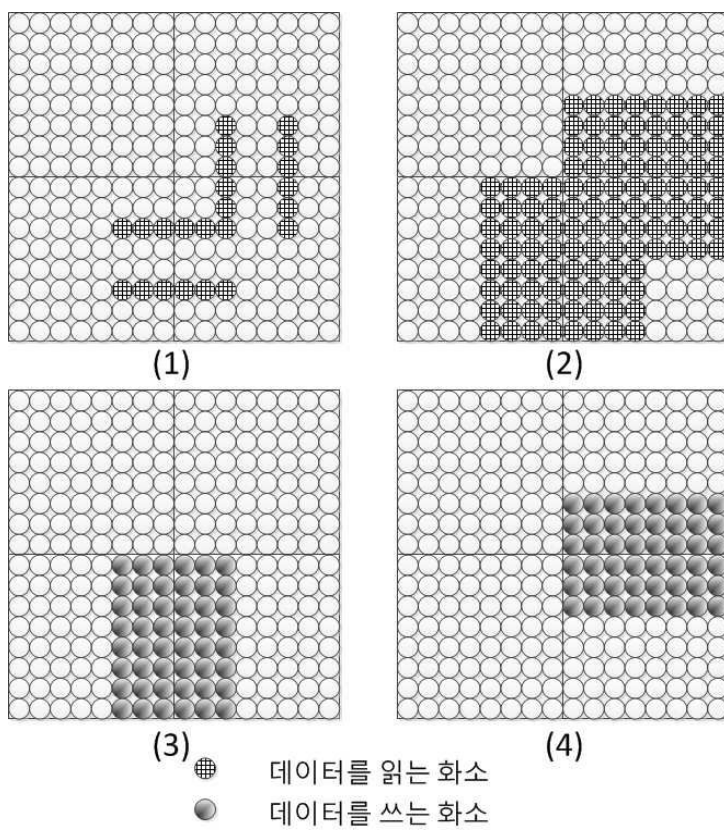
도면3



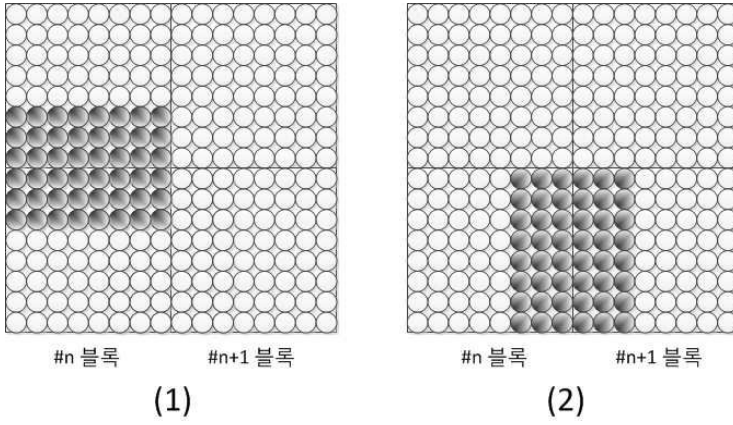
도면4



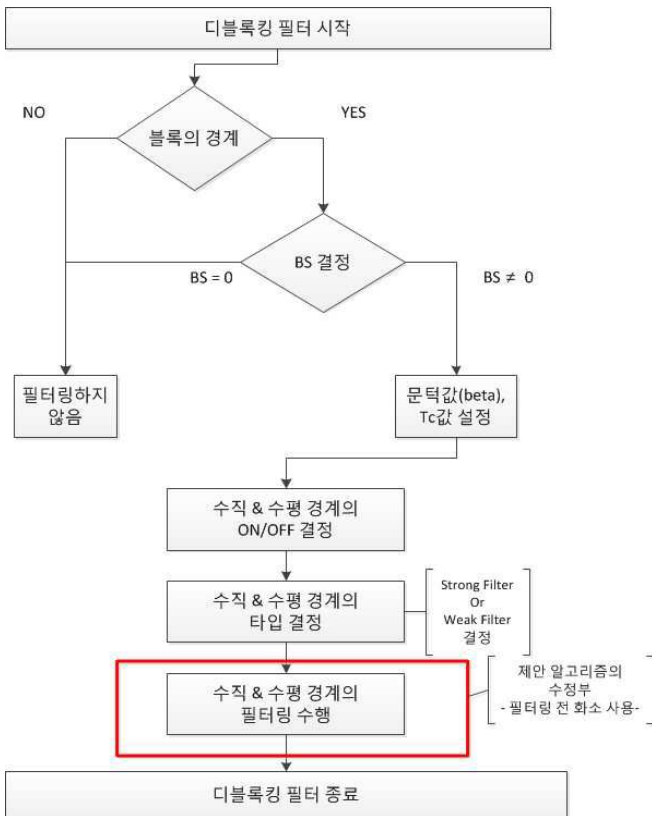
도면5



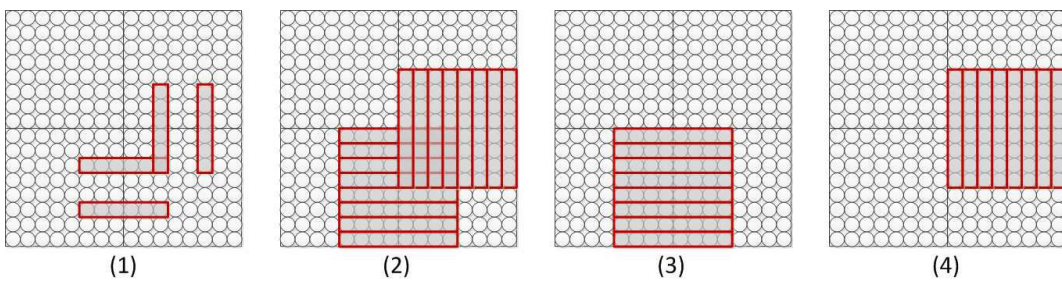
도면6



도면7



도면8



도면9

