



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년07월31일
(11) 등록번호 10-2140271
(24) 등록일자 2020년07월27일

(51) 국제특허분류(Int. Cl.)
H04N 19/102 (2014.01) H04N 19/593 (2014.01)
(21) 출원번호 10-2013-0154410
(22) 출원일자 2013년12월12일
심사청구일자 2018년08월08일
(65) 공개번호 10-2014-0076652
(43) 공개일자 2014년06월20일
(30) 우선권주장
1020120144078 2012년12월12일 대한민국(KR)
(56) 선행기술조사문헌
KR100773761 B1*
이호동 외, "HEVC 고속 부호화를 위한 주변 CU의 깊이정보 유사성에 기반한 빠른 CU 깊이 결정 방법", 대한전자공학회 학술대회, 2012.6, 1171-4.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
서울미디어대학원대학교 산학협력단
서울 강서구 화곡로61길 99, (등촌동)
(72) 발명자
최윤식
경기 고양시 일산서구 강선로 142, 1704동 501호 (일산동, 후곡마을17단지아파트)
김용구
서울 서초구 신반포로 9, 97동 507호 (반포동, 주공아파트)
김중호
서울 마포구 동교로41길 11, 204호 (연남동)
(74) 대리인
특허법인우인

전체 청구항 수 : 총 12 항

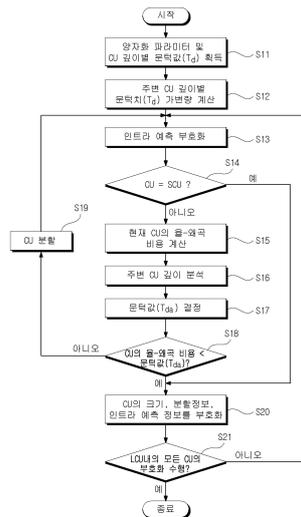
심사관 : 김영태

(54) 발명의 명칭 문턱값 기반 부호화 단위 분할을 이용한 고속 인트라 부호화 방법 및 장치

(57) 요약

문턱값 기반 부호화 단위 분할을 이용한 고속 인트라 부호화 방법 및 장치를 공개한다. 본 발명은 영상 부호화 방법에 있어서, 부호화하고자 하는 코딩유닛에 인접하는 주변 블록들을 이용하여 코딩유닛을 인트라 예측 부호화하는 단계, 코딩 유닛의 깊이와 양자화 파라미터에 따라 부호화를 수행 중인 현재 코딩 유닛의 문턱값을 설정하는 단계, 현재 코딩 유닛의 문턱값을 이전 부호화된 주변 코딩유닛들의 깊이에 따라 문턱값에 따라 가변하는 단계, 코딩유닛의 울-왜곡 비용과 문턱값을 비교하는 단계 및 비교단계의 결과에 따라 코딩유닛의 분할여부를 결정하는 단계를 포함한다.

대표도 - 도7



이 발명을 지원한 국가연구개발사업
과제고유번호 1415123227
부처명 지식경제부
연구관리전문기관 정보통신산업진흥원
연구사업명 정보통신기술인력양성
연구과제명 디지털 방송장비 연구
기 여 율 1/1
주관기관 연세대학교 산학협력단
연구기간 2012.01.01 ~ 2012.12.31

명세서

청구범위

청구항 1

영상 부호화 방법에 있어서,

부호화하고자 하는 코딩유닛에 인접하는 주변 블록들을 이용하여 상기 코딩유닛을 인트라 예측 부호화하는 단계;

상기 코딩 유닛의 깊이와 양자화 파라미터에 따라 부호화를 수행 중인 현재 코딩 유닛의 문턱값을 설정하는 단계;

상기 현재 코딩 유닛의 문턱값을 이전 부호화된 주변 코딩유닛들의 깊이에 따라 상기 문턱값을 가변하는 단계;

상기 코딩유닛의 울-왜곡 비용과 상기 문턱값을 비교하는 단계; 및

상기 비교 결과에 따라 상기 코딩유닛의 분할여부를 결정하는 단계;를 포함하며,

상기 코딩유닛의 분할여부를 결정하는 단계는 (i) 상기 코딩유닛의 울-왜곡 비용이 상기 문턱값보다 작으면 상기 코딩유닛의 분할을 수행하지 않고 상기 코딩유닛의 크기, 분할 정보 및 인트라 예측 정보를 부호화하고, (ii) 상기 코딩유닛의 울-왜곡 비용이 상기 문턱값보다 크거나 동일하면 상기 코딩유닛의 분할을 수행하며,

상기 문턱값은 상기 문턱값의 설정으로 인해 분할되지 않은 코딩유닛 중에 분할되어야 최적 부호 효율을 제공할 수 있는 코딩유닛이 전체 코딩유닛에서 차지하는 비율을 의미하는 매개 변수를 기준으로 설정되는 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 2

삭제

청구항 3

삭제

청구항 4

제1 항에 있어서, 상기 고속 인트라 부호화 방법은

상기 코딩유닛이 SCU(Smallest Coding Unit)인지 체크하는 단계를 더 포함하는 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 5

제1 항에 있어서, 상기 고속 인트라 부호화 방법은

상기 코딩유닛의 분할을 수행한 후, 분할된 코딩유닛들을 각각 인트라 예측 부호화하는 단계를 더 포함하는 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 6

제1 항에 있어서,

상기 주변 코딩유닛들은 상기 현재 코딩 유닛의 상하좌우에 인접하여 배치되는 4개의 코딩 유닛 중 적어도 하나인 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 7

제1 항에 있어서, 상기 고속 인트라 부호화 방법은

LCU (Largest Coding Unit)내의 모든 코딩유닛의 부호화가 수행되었는지 체크하는 단계를 더 포함하는 것을 특

징으로 하는 고속 인트라 부호화 방법.

청구항 8

제7 항에 있어서, 상기 고속 인트라 부호화 방법은

상기 LCU내의 모든 코딩유닛의 부호화가 수행되었으면 상기 LCU의 부호화를 종료하는 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 9

제7 항에 있어서, 상기 고속 인트라 부호화 방법은

상기 LCU내의 모든 코딩유닛의 부호화가 수행되지 않았으면, 부호화가 수행되지 않는 코딩유닛에 대해 인트라 예측 부호화하는 것을 특징으로 하는 고속 인트라 부호화 방법.

청구항 10

프로세서(processor); 및

상기 프로세서와 연결되며 상기 프로세서를 구동하기 위한 정보를 저장하는 메모리를 포함하되,

상기 프로세서는, 부호화하고자 하는 코딩유닛에 인접하는 주변 블록들을 이용하여 상기 코딩유닛을 인트라 예측 부호화하고 상기 코딩 유닛의 깊이와 양자화 파라미터에 따라 부호화를 수행 중인 현재 코딩 유닛의 문턱값을 설정하며, 상기 현재 코딩 유닛의 문턱값을 이전 부호화된 주변 코딩유닛들의 깊이에 따라 상기 문턱값을 가변하고, 상기 코딩유닛의 울-왜곡 비용을 산출한 후 상기 문턱값과 비교하고, 상기 비교 결과에 따라 상기 코딩 유닛의 분할여부를 결정하며,

상기 프로세서는, (i) 상기 코딩유닛의 울-왜곡 비용이 상기 문턱값보다 작으면 상기 코딩유닛의 분할을 수행하지 않고 상기 코딩유닛의 크기, 분할 정보 및 인트라 예측 정보를 부호화하고, (ii) 상기 코딩유닛의 울-왜곡 비용이 상기 문턱값보다 크거나 동일하면 상기 코딩유닛의 분할을 수행하며,

상기 문턱값은 상기 문턱값의 설정으로 인해 분할되지 않은 코딩유닛 중에 분할되어야 최적 부호 효율을 제공할 수 있는 코딩유닛이 전체 코딩유닛에서 차지하는 비율을 의미하는 매개 변수를 기준으로 설정되는 것을 특징으로 하는 고속 인트라 부호화 장치.

청구항 11

삭제

청구항 12

제10 항에 있어서, 상기 프로세서는

상기 코딩유닛이 SCU(Smallest Coding Unit)인지 체크하는 확인하고, 상기 코딩유닛이 SCU이면, 상기 코딩유닛의 분할을 수행하지 않는 것을 특징으로 하는 고속 인트라 부호화 장치.

청구항 13

제10 항에 있어서, 상기 프로세서는

상기 코딩유닛의 분할을 수행한 후, 분할된 코딩유닛들을 각각 인트라 예측 부호화를 다시 수행하는 것을 특징으로 하는 고속 인트라 부호화 장치.

청구항 14

제10 항에 있어서, 상기 프로세서는

상기 코딩유닛을 4개의 하위 코딩유닛으로 분할하는 것을 특징으로 하는 고속 인트라 부호화 장치.

청구항 15

제10 항에 있어서, 상기 프로세서는

LCU(Largest Coding Unit)내의 모든 코딩유닛의 부호화가 수행되었는지 체크하고, 상기 LCU내의 모든 코딩유닛의 부호화가 수행되었으면 상기 LCU의 부호화를 종료하고, 상기 LCU내의 모든 코딩유닛의 부호화가 수행되지 않았으면, 부호화가 수행되지 않는 코딩유닛에 대해 인트라 예측 부호화하는 것을 특징으로 하는 고속 인트라 부호화 장치.

발명의 설명

기술 분야

[0001] 본 발명은 고속 인트라 부호화 방법 및 장치에 관한 것으로, 특히 문턱값 기반으로 부호화 단위의 분할 여부를 판별하고, 판별된 분할 여부에 따라 부호화 단위를 분할하여 인트라 부호화를 수행하는 고속 인트라 부호화 방법 및 장치에 관한 것이다.

배경 기술

[0002] 초고화질 (UHD - Ultra High Definition) TV 방송의 상용화에 어려움 중 한 가지는 높은 공간, 시간 및 색상 해상도에 따른 막대한 데이터 량이다. 이는 단순히 영상 전송을 위한 압축의 효율성 문제만을 의미하는 것이 아니라, 제작 단계에서의 데이터 처리나 캠코더에서의 압축 문제를 모두 포함한다. 실제로 Full HD (High Definition - 1,920x1,080) 급 방송 카메라에서 현재 사용하고 있는 압축 기술은 MPEG-2 비디오 압축 표준의 Long-GOP (Group of Pictures) 모드나 H.264/AVC 비디오 압축 표준의 Intra Profile 모드로, 각기 50Mbps와 100Mbps 급 압축 원본을 캠코더 내에 저장한다. 4K급 UHD 콘텐츠의 해상도가 Full HD 콘텐츠의 최소 4배인 점을 감안하면, 기본적으로 모든 인터페이스에서 데이터 처리에 기본적으로 최소 4배 이상의 시간이 소요되고, 최소 4배 이상의 저장 용량을 필요로 한다. 하지만, 현재 최신 압축 표준인 H.264/AVC 기술이 HD급 이상의 초고해상도에서 그 압축 성능에 많은 개선의 여지를 가지고 있다는 사실이 이미 오래전부터 알려져 왔다.

[0003] 이에 최근 MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)은 H.264/AVC의 압축 효율을 향상시키는 것을 목표로 하는 영상 압축 기술인 HEVC(High Efficiency Video Coding)에 대한 표준화를 진행하였다.

[0004] HEVC는 기본적으로 H.264/AVC와 마찬가지로 하이브리드 방식의 기술을 채용하고 있다. 그러나 높은 해상도에서 보다 높은 압축 성능을 얻기 위해, 기존 압축 부호화 표준에서 사용하던 16x16 크기의 매크로블록을 64x64 크기의 부호화 단위(Coding Unit : CU)로 대체하였다. 그리고 기존 표준의 최대 8 방향 인트라 예측 방향을 최대 35 방향으로 확장하였으며, 움직임 벡터의 추정을 위한 보다 개선된 PU (Prediction Unit) 구조의 채택, 소수 화소 단위의 움직임 벡터 추정을 위한 필터의 개선, 움직임 벡터의 부호화의 효율 향상 및 변환 단위(TU ?? Transform Unit)의 확장과 새로운 변환의 설계 등 폭넓은 비디오 부호화 기술 세부 단위의 개선을 이루었다. 그 결과 기존 H.264/AVC 압축 표준의 HP (High Profile)를 기준으로 평균 35.4%의 압축 성능 향상을 이룩하였다.

[0005] HEVC의 기본 부호화 단위인 CU는 높은 해상도 영상이 가진, 일반적으로 넓은 공간 범위에서의 높은 화소 유사도를 효과적으로 이용하기 위해, H.264/AVC에서 16x16 의 고정된 크기를 갖는 매크로 블록을 확장하여 최대 64x64 크기에서 부호화 효율을 고려하여 8x8 크기까지 선택적으로 사용할 수 있다. CU 크기의 선택은 해당 CU의 부호화가 어떤 크기에서 가장 효율적으로 수행될 수 있는지를 부호기가 판단하여 결정하도록 되어있는데, HEVC 부호기 참조 모델을 보면 최대 크기 CU가 재귀적으로 분할되면서 (Quadtree 분할) 분할된 하위 CU들의 최적 부호화 결과를 분할되기 전 상위 CU의 최적 부호화 결과와 비교함으로써 수행되고 있다. 이러한 비교는 일반적으로 윌-왜곡 비용 (Rate-Disortion Cost)을 기준으로 사용하는데, 이러한 윌-왜곡 비용의 생성을 위해서는 실제 부호화를 따라서 비트 생성에 이르기까지 모두 수행해야하므로 각 CU 블록을 반복해서 압축해야 한다. 이러한 반복적 압축은 가장 효율적인 모드를 부호기에서 선택할 수 있도록 해주지만, 그로인한 계산 복잡도는 반복 회수 및 한 번 부호화를 수행할 때 고려해야 할 예측의 종류 및 그 방법에 따라 매우 크게 증가할 수 있다.

[0006] 도1 은 HEVC의 CU의 계층적인 구조를 나타낸다.

[0007] HEVC에서 가장 큰 크기의 CU를 LCU(Largest Coding Unit)라고 하고, 가장 작은 크기의 CU를 SCU(Smallest Coding Unit)라고 하는데, 이러한 CU의 크기는 부호화기에서 정의 될 수 있다. 도1 에서 (a)는 CU가 LCU로부터 SCU로 계층적으로 분할될 수 있음을 도시하고 있으며, (b)는 각 CU의 표시 방법을 나타낸다.

[0008] 도1 에서 LCU는 64x64로, SCU는 8x8로 정의되어 있다. 각 CU는 (a)에 도시된 바와 같이, 가로 및 세로의 크기가

절반인 4개의 하위 CU로 분할될 수 있는데, 이러한 분할 과정은 LCU에서 SCU까지 재귀적인 쿼드트리 방식으로 이루어진다. 여기서 각 CU의 크기는 깊이(depth) 정보를 이용해서 표현되는데, LCU의 깊이를 0으로 정의하고 각 CU가 한 번씩 분할될 때마다 깊이 정보가 1씩 증가하는 방식을 사용한다. HEVC는 이러한 다양한 크기를 가지는 CU 단위로 인트라 부호화가 수행되는데, 하나의 CU를 해당 크기 그대로 인트라 부호화할 경우의 윌-왜곡 비용(Rate Distortion cost : RD cost)과 CU를 4개의 하위 CU로 분할하여 각각을 따로 인트라 부호화할 경우의 윌-왜곡 비용의 합을 서로 비교하여 더 작은 윌-왜곡 비용을 도출하는 경우를 선택함으로써 CU의 분할 여부를 결정한다. 여기서 CU가 분할되지 않을 경우 split flag가 0으로, CU가 분할될 경우 split flag가 1로 부호화되어 CU의 분할 정보를 복호화기로 전달하게 된다.

[0009] 이처럼 CU의 분할 여부를 결정할 때 각 CU의 윌-왜곡 비용을 계산하기 위해서는 실제 부호화 과정이 모두 수행되어야 하기 때문에 부호화기에 많은 연산량이 요구된다. 특히 HEVC의 인트라 부호화의 경우 하나의 블록 당 35가지의 후보 예측 모드를 가질 수 있기 때문에 하나의 CU에 대한 연산량도 많고, 이러한 부호화 과정을 다양한 크기의 CU에 대해 모두 수행한 후 윌-왜곡 비용을 비교하기 때문에 부호화기의 복잡도가 매우 높은 문제점이 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명의 목적은 HEVC의 인트라 부호화 과정에서 CU를 분할하지 않고, 상위 CU의 윌-왜곡 비용만을 고려하여 해당 CU의 재귀적 분할 여부를 결정하도록 하여, CU 구조를 조기에 결정할 수 있도록 함으로써, HEVC의 인트라 부호화 과정의 복잡도를 줄일 수 있는 고속 인트라 부호화 방법을 제공하는 데 있다.

과제의 해결 수단

[0011] 상기 목적을 달성하기 위한 본 발명의 일 예에 따른 고속 인트라 부호화 방법은 영상 부호화 방법에 있어서, 부호화하고자 하는 코딩유닛에 인접하는 주변 블록들을 이용하여 상기 코딩유닛을 인트라 예측 부호화하는 단계; 상기 코딩 유닛의 깊이와 양자화 파라미터에 따라 부호화를 수행 중인 현재 코딩 유닛의 문턱값을 설정하는 단계; 상기 현재 코딩 유닛의 문턱값을 이전 부호화된 주변 코딩유닛들의 깊이에 따라 상기 문턱값을 가변하는 단계; 상기 코딩유닛의 윌-왜곡 비용과 상기 문턱값을 비교하는 단계; 및 상기 비교단계의 결과에 따라 상기 코딩 유닛의 분할여부를 결정하는 단계;를 포함한다.

[0012] 상기 목적을 달성하기 위한 본 발명의 일 예에 따른 고속 인트라 부호화 장치는 프로세서(processor); 및 상기 프로세서와 연결되며 상기 프로세서를 구동하기 위한 정보를 저장하는 메모리를 포함하되, 상기 프로세서는, 부호화하고자 하는 코딩유닛에 인접하는 주변 블록들을 이용하여 상기 코딩유닛을 인트라 예측 부호화하고 상기 코딩 유닛의 깊이와 양자화 파라미터에 따라 부호화를 수행 중인 현재 코딩 유닛의 문턱값을 설정하며, 상기 현재 코딩 유닛의 문턱값을 이전 부호화된 주변 코딩유닛들의 깊이에 따라 문턱값에 따라 가변하고, 상기 코딩유닛의 윌-왜곡 비용을 산출한 후 상기 문턱값과 비교하고, 상기 비교 결과에 따라 상기 코딩유닛의 분할여부를 결정하도록 구성된다.

발명의 효과

[0013] 따라서, 본 발명의 고속 인트라 부호화 방법 및 장치는 HEVC의 인트라 부호화 시 최적의 CU의 구조를 결정하는 과정에서 임의의 CU에 깊이 및 양자화 파라미터에 따라 미리 설정되는 문턱값을 주변 CU의 깊이에 대응하여 가변하고, 가변된 문턱값과 현재 CU에 대한 윌-왜곡 비용을 비교하여 CU의 분할 여부를 결정한다. 이를 통해 분할할 필요가 없는 CU를 조기에 선택하여 불필요한 CU의 분할 과정을 생략함으로써 부호화기의 연산 복잡도를 줄일 수 있다. 또한 문턱값을 현재 CU와 주변 CU의 부호화 정보를 이용하여 적응적으로 결정함으로써 보다 CU 분할 여부 판단의 적중도를 높임으로써 부호화 효율 감소를 최소화할 수 있다.

도면의 간단한 설명

- [0014] 도1 은 HEVC의 CU의 계층적인 구조를 나타낸다.
- 도2 는 CU에 따른 윌-왜곡 비용 히스토그램의 예를 나타낸다.
- 도3 은 FTR와 FSR의 개념을 설명하기 위한 도면이다.

도4 는 FTR의 변화에 따른 율-왜곡과 복잡도의 변화를 나타낸다.

도5 는 각각의 CU 깊이에 대한 문턱값과 양자와 파라미터의 상관관계를 나타낸다.

도6 은 본 발명의 영상 부호화 장치를 개략적으로 나타낸 블록도이다.

도7 은 본 발명의 일 실시예에 따른 고속 인트라 부호화 방법을 나타낸다.

발명을 실시하기 위한 구체적인 내용

- [0015] 본 발명과 본 발명의 동작상의 이점 및 본 발명의 실시에 의하여 달성되는 목적을 충분히 이해하기 위해서는 본 발명의 바람직한 실시예를 예시하는 첨부 도면 및 첨부 도면에 기재된 내용을 참조하여야만 한다.
- [0016] 이하, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예를 설명함으로써, 본 발명을 상세히 설명한다. 그러나, 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 설명하는 실시예에 한정되는 것이 아니다. 그리고, 본 발명을 명확하게 설명하기 위하여 설명과 관계없는 부분은 생략되며, 도면의 동일한 참조부호는 동일한 부재임을 나타낸다.
- [0017] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.
- [0018] 상기한 바와 같이, 기존의 HEVC의 참조 모델은 압축 효율 면에서 최적의 CU크기를 결정하기 위해서 각 CU 분할에 따른 율-왜곡 비용을 평가하면서, 재귀적으로 CU 분할을 수행하고 있다. 이러한 모든 CU 모드에 대한 탐색은 최적의 압축 효율을 제공하지만, 복잡도가 매우 높아진다.
- [0019] 본 발명의 최적 CU 크기 설정에 대한 아이디어는, 큰 CU는 화소의 변화가 크지 않은 평평한 배경 부에 많이 존재하고, 작은 크기의 CU는 최대 CU 내에 윤곽선 등의 존재로 복잡한 화소 밝기 값의 변화가 존재하는 부분에서 선택될 것이라는 사실이다. 따라서 큰 CU가 최적인 경우에는 그 CU의 율-왜곡 비용이 다소 낮을 것을 기대할 수 있고, 반면 최적 CU가 작은 경우에는 큰 CU에서 상대적으로 높은 율-왜곡 값을 가지게 될 것을 기대할 수 있다. 이를 검증하기 위해, 다양한 특성의 테스트 비디오에 대해 여러개의 양자화 파라미터(quantization parameter : QP)를 설정한 후 HEVC의 참조 모델(HM : HEVC Test Model) 소프트웨어에서 실험을 수행하였다. 이 실험은 각각의 CU 크기에서 그 크기가 최적으로 선택되는 경우와 더 작은 크기에서 최적의 율-왜곡이 제공되는 두 가지 경우에 있어 각 CU가 어떤 율-왜곡 비용 값을 가지게 되는지를 율-왜곡 비용 히스토그램으로 데이터 수집을 수행하는 것이다.
- [0020] 도2 는 CU에 따른 율-왜곡 비용 히스토그램의 예를 나타낸다.
- [0021] 도2 는 Basketball Drive라는 시퀀스(sequence)를 양자화 파라미터(QP) 27 및 37로 각각 부호화하는 경우를 도시한 것으로, (a) ~ (d)는 QP = 27로 부호화한 경우를 나타내며, (e) ~ (h)는 QP = 37로 부호화한 경우를 나타낸다. 그리고 (a)및 (e)는 CU의 크기가 64x64로, (b) 및 (f)는 CU의 크기가 32x32로, (c) 및 (g)는 CU의 크기가 16x16로, 그리고 (d) 및 (h)는 CU의 크기가 8x8로 부호화된 경우의 율-왜곡 비용 히스토그램을 나타낸다.
- [0022] 즉 도2 는 서로 다른 양자와 파라미터와 서로 다른 크기의 CU에서 각각 최적의 율-왜곡을 제공하는 CU들과 이보다 작은 크기에서 최적의 율-왜곡을 제공하는 CU들을 따로 모은 후, 각 CU가 어떤 율-왜곡 값을 나타내는지 조사한 결과이다. 도2 에서 그리고 각각의 도면에서 P(J|Non-SPLIT) 및 P(J|SPLIT)는 하위 CU로 분할되지 않은 경우와 분할된 경우 각각에 따른 조건부 확률 밀도 함수로서, 파란색 라인인 P(J|Non-SPLIT)는 현재 CU의 크기에서 최적 율-왜곡 비용 함수가 발생하는 경우이고, 빨간색 라인인 P(J|SPLIT)는 하위 CU로 분할되는 경우 최적의 율-왜곡 비용 함수가 나오는 경우이다. 도 2에서 x-축은 율-왜곡 비용 값을, y-축은 해당 CU의 수를 시퀀스의 전체 CU 수에 대한 비(%)로 나타낸 것이다.
- [0023] 도2 에 도시된 바와 같이, 현재 CU의 크기가 최적의 CU일 경우 (즉, 더 이상 분할이 필요하지 않은 경우) 특정 율-왜곡 비용을 가지는 CU들의 분포는 율-왜곡 비용이 낮은 값을 가지는 쪽에 많이 분포하고 있고, 반대로 추가 분할을 수행해야 하는 경우의 CU들은 전반적으로 넓은 율-왜곡 비용 범위에 분포하고 있다. 이러한 율-왜곡 비용의 특성은, 부호화 과정에서 특정 크기의 CU에 대한 율-왜곡 비용이 산출되었을 때, 이 값을 특정 문턱값과 비교하여 추가 분할 여부를 결정함으로써 추가 분할에 기인한 재귀적 부호화의 계산 부담을 효율적으로 제거할 수 있다는 사실을 나타낸다.

[0024] 이에 본 발명에서는 적절한 문턱값을 설정하여, 각 CU 별 재귀적 분할 과정에 있어, 분할된 CU의 율-왜곡 비용을 계산하기 전에 현재 CU 크기에서의 율-왜곡 비용을 설정된 문턱값과 비교함으로써 충분히 작은 율-왜곡 비용을 가지는 CU에 대해 더 이상의 CU 분할을 수행하지 않도록 하여 고속 부호화를 수행할 수 있도록 한다. 즉 본 발명에서 부호화기는 수학식 1과 같이 현재 CU의 크기에서 율-왜곡 비용이 설정된 문턱값보다 낮으면, CU 분할을 수행하지 않고 그대로 부호화를 수행하도록 하고, 현재 CU의 크기에서 율-왜곡 비용이 설정된 문턱값보다 크거나 같으면, CU 분할을 수행하도록 한다.

수학식 1

$$Decision = \begin{cases} Non-SPLIT, & \text{if } F_d < T_d \\ SPLIT, & \text{otherwise} \end{cases}$$

[0025]

[0026] (여기서, F_d 는 깊이 d에서 현재 CU의 율-왜곡 비용이고, T_d 는 깊이 d의 현재 CU에 대한 문턱값이다.)

[0027] 문턱값(T_d)은 현재 CU의 크기에 따라 다르게 설정될 수 있다. 문턱값(T_d)은 부호화의 복잡도 감소와 율-왜곡 비용 증대 사이의 균형을 조절할 수 있는 파라미터로, 본 발명에서는 적절한 문턱값 설정을 위해 FTR(False Termination Rate)과 FSR(False Split Rate)를 정의한다.

[0028] 도3 은 FTR와 FSR의 개념을 설명하기 위한 도면이다.

[0029] 도3 은 도2 와 같이 율-왜곡 비용 히스토그램을 나타내며, 도3 을 분석하면, FTR은 문턱값 설정으로 인해 분할되지 않은 CU 중에 CU 분할되어야 최적 부호 효율을 제공할 수 있는 CU가 전체 CU의 몇 %에 해당하는지를 의미하는 매개 변수를 나타내며, FSR은 분할되지 않아야 최적 부호 효율을 제공할 수 있는 CU가 전체 CU의 몇 %에 해당하는지를 의미하는 매개 변수를 나타낸다.

[0030] FTR은 수학식 2와 같이 계산될 수 있다.

수학식 2

$$FTR_d = \int_0^{T_d} P(J|SPLIT)dJ$$

[0031]

[0032] 그리고 FSR은 수학식 3과 같이 계산될 수 있다.

수학식 3

$$FSR_d = \int_{T_d}^{\infty} P(J|Non-SPLIT)dJ$$

[0033]

[0034] 여기서 FTR_d 와 FSR_d 는 깊이 d에서 문턱값(T_d)에 의해 분할과 비분할로 구분되어 최적 부호 효율을 제공하는데 실패한 CU들의 백분율을 나타낸다.

[0035] 도3 에서 FSR은 율-왜곡 손실과 부호화 시간 절감에 크게 연관되는 반면, FSR은 부호화 시간 절감에 연관되고 율-왜곡 손실에는 연관되지 않는다. FSR의 경우, 비 분할(Non-SPLIT) 분류에서 CU들에 대한 결정이 틀렸을지라도, 통상의 부호화 과정은 어떤 율-왜곡 손실 없이 원본처럼 진행될 수 있다. 그러나 FTR의 경우, 요구되는 추가 분할 과정이 스킵되므로, 분할(SPLIT) 분류에서 CU들에 대해 결정이 틀리면, 부호화 시간 절감과 더불어 율-왜곡 손실이 유발된다. 따라서 복잡도 감소와 율-왜곡 비용 감소 사이의 트레이드 오프(Trade-off)를 판단하기 위한 파라미터로는 FTR이 FSR보다 더 적합하다. 이에 문턱값(T_d)은 특정한 FTR 값을 대상으로 결정될 수 있다.

[0036] FTR에 따른 복잡도 감소와 율-왜곡 감소 사이의 트레이드 오프 성능을 서로 다른 해상도의 여러 테스트 시퀀스 (테스트 시퀀스로 잘 알려진Traffic, ParkScene, BQMall, BasketballPass 및 Johnn)들을 이용하여 실험하였다.

[0037] 도4 는 FTR의 변화에 따른 율-왜곡과 복잡도의 변화를 나타낸다.

[0038] 도4 에서 (a)는 FTR의 0% ~ 10% 까지의 변화에 따른 율-왜곡 손실을 측정하기 위한 BDBR(Bjonteggard Delta BitRate) 그래프를 나타내며, (b)는 부호화 시간 절감에 기초한 복잡도 감소 비율을 나타낸다. 부호화 시간 절감(Encoing Time Saving : EST)는 수학적 4와 같이 계산될 수 있다.

수학적 4

$$ETS(\%) = \frac{ET_{anchor} - ET_{proposed}}{ET_{anchor}} \times 100$$

[0039]

[0040] 여기서 ET_{anchor} 은 기존의 부호화 방법에 따른 부호화 시간을 나타내며, $ET_{proposed}$ 는 본 발명의 부호화 방법에 따른 부호화 시간을 나타낸다.

[0041] 도4 를 참조하면 FTR이 증가함에 따라 율-왜곡 손실이 증가하는 반면 부호화 시간 절감율은 점차적으로 포화된다. FTR이 크면, 상대적으로 큰 율-왜곡 손실에 비해 복잡도 감소에 작은 이득을 준다. 도4 의 결과로부터 FTR이 6% 로 문턱값(T_d)이 설정되면, 대략 1%의 BDBR 손실과 30% 부호화 시간 절감을 발생한다. 즉 율-왜곡의 큰 손실 없이 복잡성을 줄일 수 있다. 그러므로 본 발명에서는 일례로 문턱값(T_d)으로서 FTR을 6% 인 것으로 설정할 수 있다.

[0042] 상기한 바와 같이 율-왜곡 비용은 CU의 크기뿐만 아니라 양자화 파라미터(QP)에도 의존한다. 따라서 문턱값(T_d)은 CU의 크기와 양자화 파라미터(QP)를 모두 고려하여 결정되어야 한다.

[0043] 도5 는 각각의 CU 깊이에 대한 문턱값과 양자와 파라미터의 상관관계를 나타낸다.

[0044] 도5 에 도시된 바와 같이, CU 깊이에 대한 문턱값과 양자와 파라미터의 상관관계는 지수형 곡선으로 나타나게 된다. 그리고 이 지수형 곡선을 모델링하면, 각각의 CU 깊이와 양자화 파라미터(QP)에 따른 문턱값(T_d)을 수학적 5와 같이 획득할 수 있다.

수학적 5

$$T_0 = 1133e^{0.126 \times QP}$$

$$T_1 = 182e^{0.148 \times QP}$$

$$T_2 = 20.63e^{0.188 \times QP}$$

$$T_3 = 1.64e^{0.247 \times QP}$$

[0045]

[0046] 수학적 5 에 따라 현재 CU에 대한 문턱값(T_d)을 획득할 수 있으며, 획득된 문턱값(T_d)에 따라 CU의 분할 여부를 결정할 수 있다.

[0047] 그러나 상기한 바와 같이, 큰 CU는 화소의 변화가 크지 않은 평평한 배경부에 많이 존재하고, 작은 크기의 CU는

최대 CU 내에 윤곽선 등의 존재로 복잡한 화소 밝기 값의 변화가 존재하는 부분에서 선택될 것이라는 사실이다. 이는 시퀀스, 즉 이미지에서 작은 크기의 CU가 주로 배경과 물체의 경계선 등에 사용됨을 의미하고, 배경과 물체의 경계선은 대부분 서로 인접하여 배치되는 경우가 많다. 따라서, 미리 부호화된 주변 CU의 깊이를 수학적 5의 문턱값(T_d)에 추가로 반영할 수 있다.

[0048] 즉, 현재 CU의 문턱값(T_d)를 주변 CU의 깊이에 따라 증가 또는 감소시킬 수 있다. 이렇게 현재 CU의 문턱값(T_d)를 주변 CU의 깊이에 따라 증가 또는 감소시키게 되면, 현재 CU의 문턱값(T_d)에 따라 추가로 분할하거나 분할하지 않는 것으로 판단되는 CU의 크기가 주변 CU의 깊이에 따라 가변된 문턱값(T_{da})에 따라 올-왜곡을 더 줄이거나, CU를 추가로 분할하지 않도록 할 수 있다. 예를 들어, 현재 CU의 주위 CU가 배경부의 CU로서 CU의 크기가 크게 설정되어 있다면, 현재 CU의 문턱값(T_d)을 주변 CU의 깊이에 대응하는 값만큼 낮게 설정하여 추가 분할될 가능성을 낮출 수 있다. 또한 현재 CU의 주위 CU가 경계부의 CU로서 CU의 크기가 작게 설정되어 있다면, 현재 CU의 문턱값(T_d)을 주변 CU의 깊이에 대응하는 값만큼 크게 설정하여 현재 CU가 추가 분할될 가능성을 높일 수 있다.

[0049] 여기서 주변 CU는 현재 CU의 상하좌우에 배치되는 4개의 CU 중 적어도 하나일 수 있으며, 주변 CU에 의해 가변되는 문턱값(T_{da})의 크기는 주변 CU의 깊이에 따라 다양하게 설정될 수 있다.

[0050] 도6은 본 발명의 영상 부호화 장치를 개략적으로 나타낸 블록도이다.

[0051] 도6을 참조하면, 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.

[0052] 영상 부호화 장치(100)는 입력 영상에 대해 인트라(intra) 모드 또는 인터(inter) 모드로 부호화를 수행하고 비트스트림을 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.

[0053] 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.

[0054] 이때 인트라 예측부(120)는 CU의 깊이를 지정하기 위한 문턱값(T_{da})이 미리 설정되어 있으며, 설정된 문턱값(T_{da})에 따라 CU들의 분할 여부를 결정한다. 문턱값(T_{da})은 우선 수학적 5에 나타난 바와 같이, 양자화부(140)의 양자화 파라미터(QP)와 현재 CU의 깊이에 따라 현재 CU의 문턱값(T_d)이 결정되고, 현재 CU의 주변 CU들의 깊이를 현재 CU의 문턱값(T_d)에 반영하여 결정된다. 즉 본 발명에서 문턱값(T_{da})은 현재 CU의 깊이 및 양자화 파라미터 뿐만 아니라 주변 CU의 깊이를 함께 고려하여 설정된다.

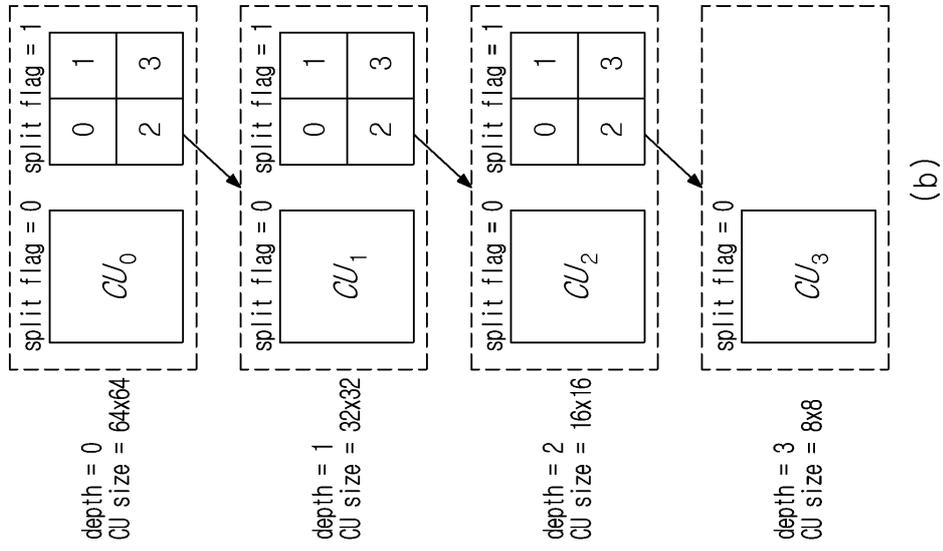
[0055] 인트라 예측부(120)는 현재 CU의 깊이d에서의 CU의 올-왜곡 비용(F_d)을 계산하고, 계산된 올-왜곡 비용(F_d)이 문턱값(T_{da})보다 작으면, 현재 깊이(d)에서의 CU의 크기를 최적의 크기로 판단하고, 다음 CU를 처리하기 위한 부호화 과정을 수행한다. 그러나, 계산된 올-왜곡 비용(F_d)이 문턱값(T_{da})보다 작지 않으면, 현재 CU가 SCU인지 판별하고, SCU가 아니면, 현재 CU를 도1에 도시된 바와 같이 4개로 분할하고, 깊이(d)를 1증가시킨다. 즉 현재 CU의 올-왜곡 비용(F_d)과 문턱값(T_{da})을 비교하여 현재 CU를 추가로 분할할지 여부를 미리 결정하므로, CU를 SCU로부터 SCU까지 모두 분할하여 올-왜곡 비용을 계산하지 않고서도, 빠르게 CU의 크기를 결정하여 부호화를 수행할 수 있다. 그러므로 계산량을 줄여 고속 인트라 부호화가 가능하도록 한다. 또한 주변 CU의 깊이를 함께 고려하여 문턱값(T_{da})을 설정함으로써, CU의 분할 또는 비분할 필요성을 더욱 정확하게 판별할 수 있도록 한다.

[0056] 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.

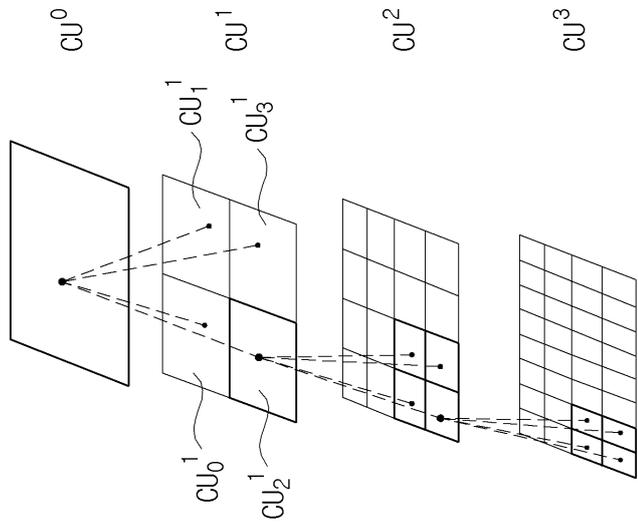
- [0057] 감산기(125)는 입력 블록과 생성된 예측 블록의 차분에 의해 잔여 블록(residual block)을 생성한다. 변환부(130)는 잔여 블록에 대해 변환(transform)을 수행하여 변환 계수(transform coefficient)를 출력한다. 그리고 양자화부(140)는 입력된 변환 계수를 양자화 파라미터(QP)에 따라 양자화하여 양자화된 계수(quantized coefficient)를 출력한다. 엔트로피 부호화부(150)는 입력된 양자화된 계수를 확률 분포에 따라 엔트로피 부호화하여 비트스트림(bit stream)을 출력한다.
- [0058] HEVC는 인트라 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.
- [0059] 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 경우에 만 수행될 수도 있다. 필터부(180)를 거친 복원 블록은 참조 영상 버퍼(190)에 저장된다.
- [0060] 도7 은 본 발명의 일 실시예에 따른 고속 인트라 부호화 방법을 나타낸다.
- [0061] 도7 은 도6 의 영상 부호화 장치 중 인트라 예측부(120)의 동작을 기초로 설명한다.
- [0062] 도7 에서 인트라 예측부(120)는 우선, 양자화 파라미터(QP) 및 CU의 깊이별 문턱값을 수학적 식 5와 같이 획득한다(S11). 그리고 획득된 문턱값(T_d)들 각각에 대해 주변 CU의 깊이에 따라 문턱값(T_d)을 가변할 가변량을 기설정된 방식으로 계산한다(S12). 여기서 주변 CU는 현재 인트라 부호화를 수행할 CU의 상하좌우에 인접하여 배치되고 미리 부호화된 4개의 CU 중 적어도 하나일 수 있으며, 경우에 따라서는 현재 CU의 주변에 배치되는 8개의 CU 중 적어도 하나일 수 있다.
- [0063] 그리고 영상이 입력되고 인트라 예측 모드이면, 인트라 예측 부호화를 시작한다(S13). 인트라 예측부(120)는 우선 현재 CU가 SUC인지 판별한다(S14). 이는 SCU가 분할할 수 있는 가장 작은 단위이므로 CU 분할을 더 이상 수행할 수 없기 때문에 이를 확인하기 위한 것이다.
- [0064] 현재 CU가 SCU가 아니면 CU의 윗-왜곡 비용을 계산한다(S15). 그리고 현재 CU의 주변 CU들의 깊이를 분석한다(S16). 주변 CU들의 깊이가 분석되면, 미리 설정된 CU 깊이별 문턱값(T_d)들 중 현재 CU의 깊이에 대응하는 문턱값(T_d)을 획득하고, 주변 CU 깊이별 가변량을 획득된 문턱값(T_d)에 반영하여 문턱값(T_{da})을 결정한다(S17).
- [0065] 그리고 계산된 현재 CU의 윗-왜곡 비용이 문턱값(T_{da})보다 작는지 판별한다(S18). 만일 현재 CU의 윗-왜곡 비용이 문턱값(T_{da})보다 작지 않으면, 현재 CU를 4개로 분할한다(S19). 그러나 현재 CU의 윗-왜곡 비용이 문턱값(T_{da})보다 작으면, CU의 크기, 분할 정보, 인트라 예측 정보를 비트스트림으로 부호화한다(S20).
- [0066] 이후 LCU내의 모든 CU의 부호화가 진행되었는지 판별한다(S21). LCU내의 모든 CU의 부호화가 수행되지 않았으면, 부호화되지 않은 CU에 대해 인트라 예측 부호화를 시작하고(S13), 모든 CU의 부호화가 완료되면 해당 LCU의 영상부호화가 종료된다.
- [0067] 본 발명에 따른 방법은 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- [0068] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다.
- [0069] 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

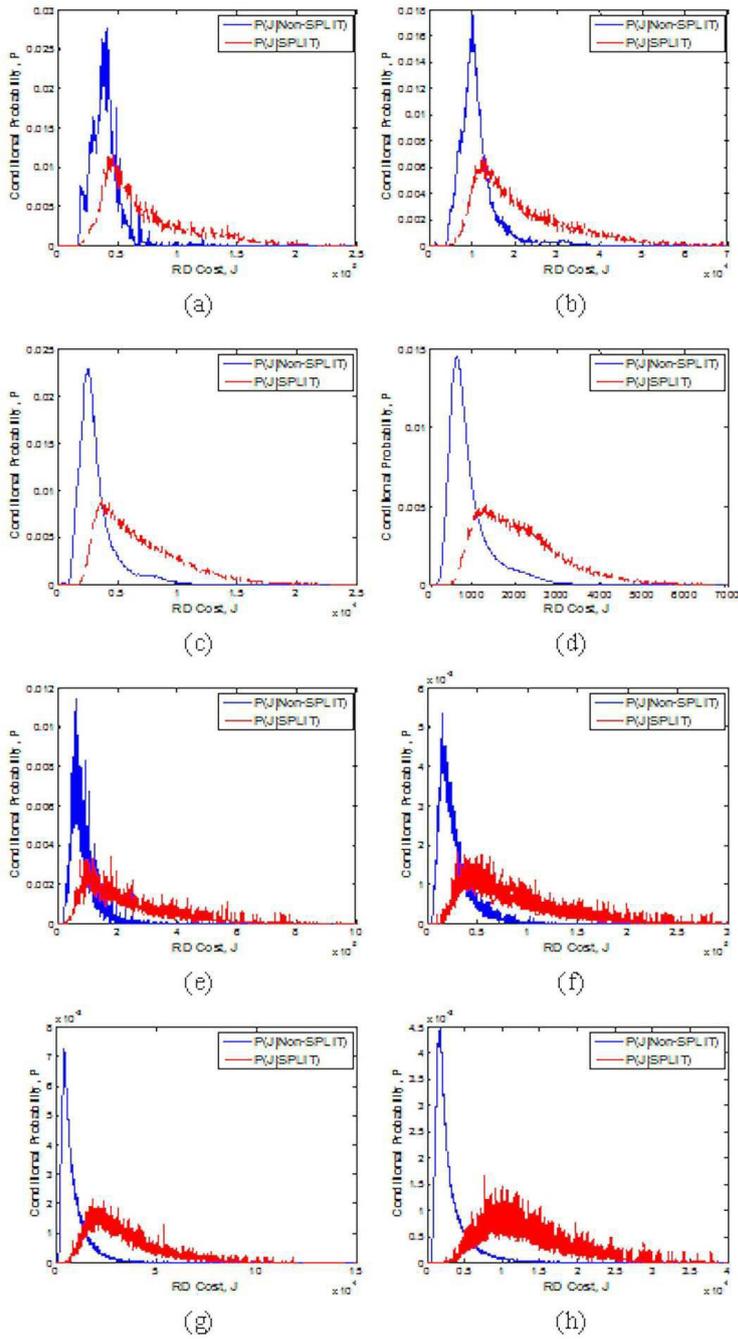
도면1



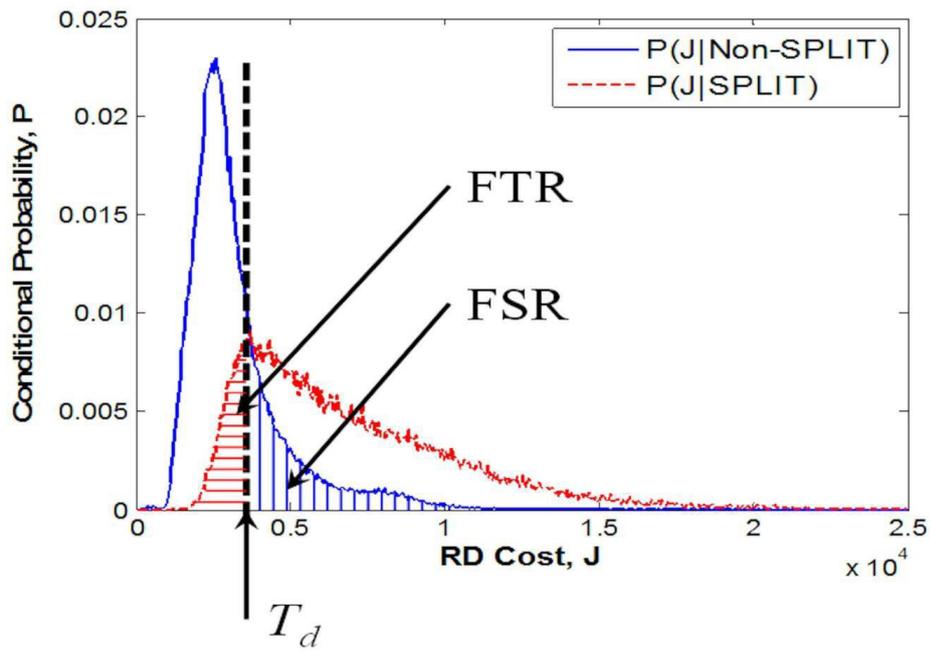
(a)



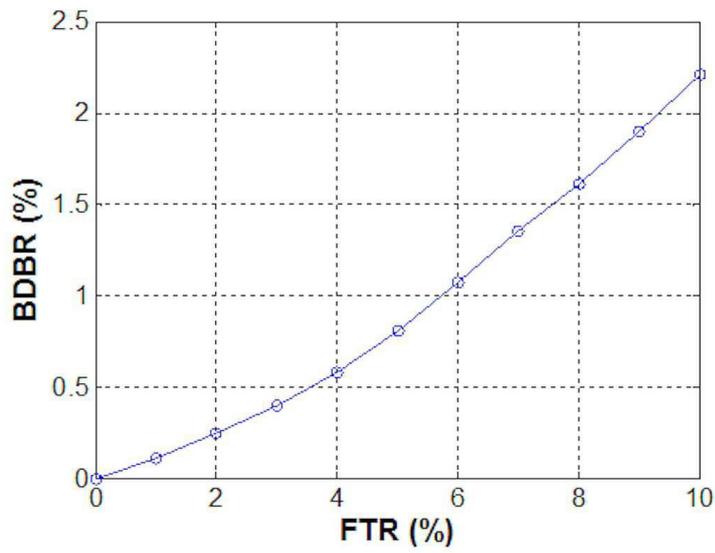
도면2



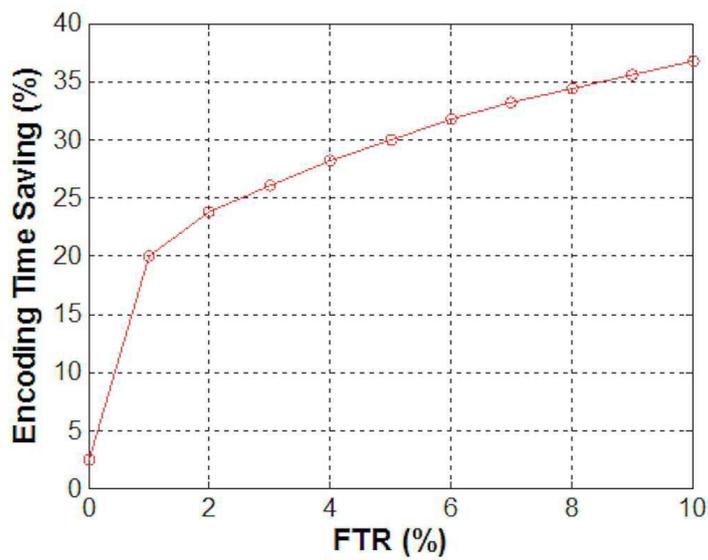
도면3



도면4

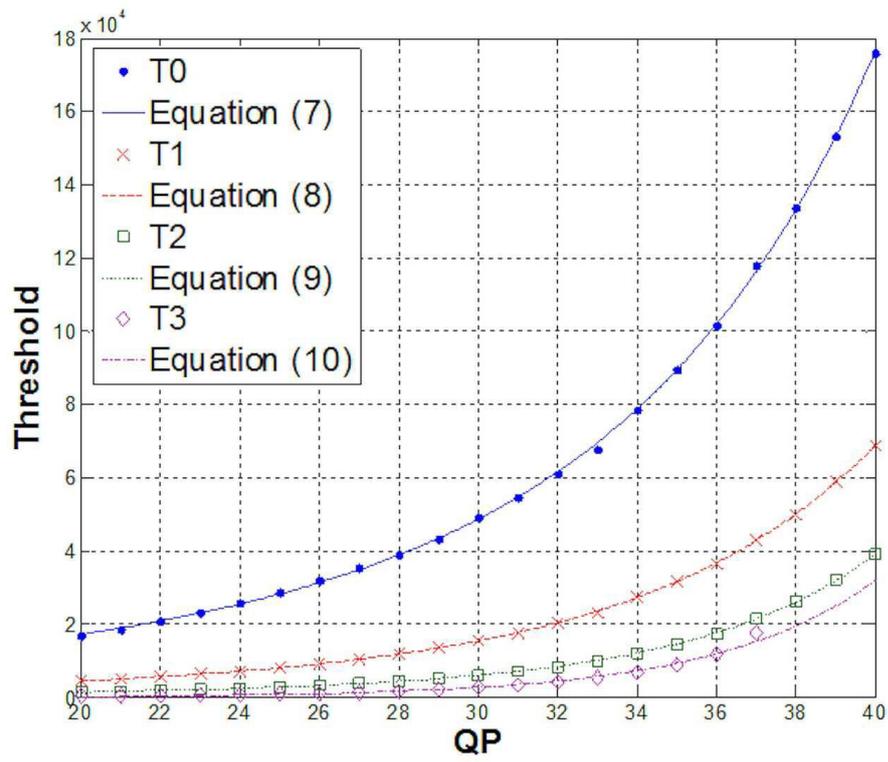


(a)

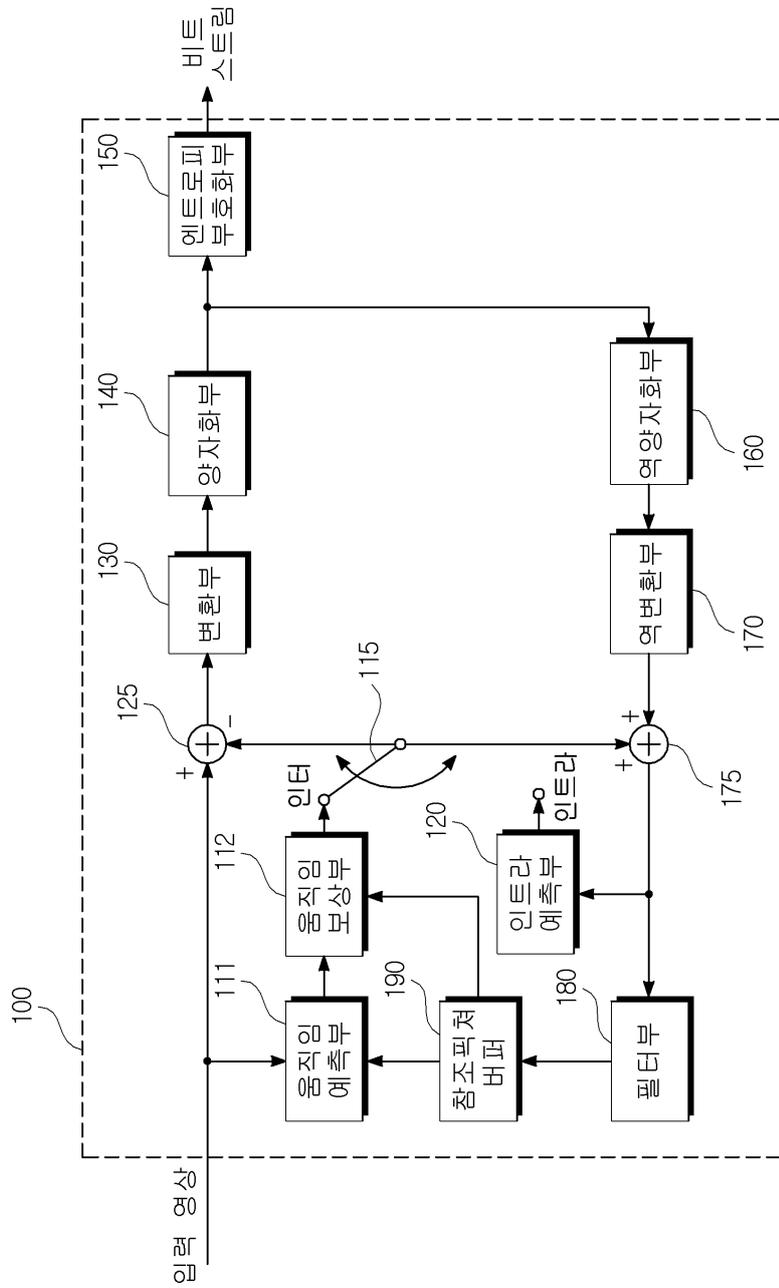


(b)

도면5



도면6



도면7

