



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0027142

(43) 공개일자 2020년03월12일

(51) 국제특허분류(Int. Cl.)

H04N 19/433 (2014.01) H04N 19/17 (2014.01)

(52) CPC특허분류

H04N 19/433 (2015.01)

H04N 1/413 (2013.01)

(21) 출원번호 10-2018-0105071

(22) 출원일자 2018년09월04일

심사청구일자 2018년09월04일

(71) 출원인

세종대학교산학협력단

서울특별시 광진구 능동로 209 (군자동, 세종대학교)

(72) 발명자

박우찬

서울특별시 송파구 올림픽로 135, 262동 1301호(잠실동, 리센즈아파트)

황임재

서울특별시 광진구 천호대로110길 111, 102호(능동, 파크빌라)

이재신

서울특별시 송파구 오금로62길 10, 101동 1201호(거여동, 쌍용스윗닷홈 거여역1차)

(74) 대리인

정부연

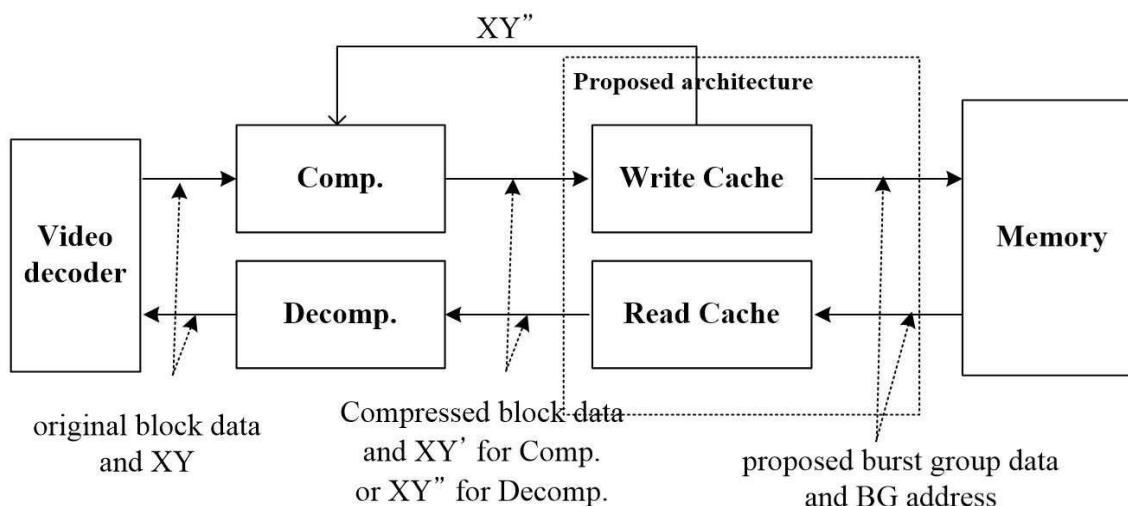
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 이미지 처리 장치 및 방법

(57) 요약

본 발명은 이미지 처리 장치 및 방법에 관한 것으로, 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 상기 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성하는 이미지 코딩부 및 상기 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹을 생성하여 상기 메모리의 제2 주소공간으로 변환하는 메모리 연산부를 포함한다. 따라서, 본 발명은 효과적인 버스트 액세스 및 랜덤 액세스를 지원하는 무손실 이미지 데이터 압축이 가능할 수 있다.

대표도 - 도7



(52) CPC특허분류

H04N 19/17 (2015.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 1711065196

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 ICT융합산업원천기술개발사업

연구과제명 극사실적인 실시간 가상현실을 위한 모바일 GPU 하드웨어 개발

기 여 율 1/1

주관기관 세종대학교 산학협력단

연구기간 2017.01.01 ~ 2021.12.31

명세서

청구범위

청구항 1

이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 상기 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성하는 이미지 코딩부; 및

상기 복수의 이미지 압축 절편들의 패키징을 통해 가변적인 버스트 그룹(Burst Group, BG)을 생성하여 상기 메모리의 제2 주소공간으로 변환하는 메모리 연산부를 포함하는 이미지 처리 장치.

청구항 2

제1항에 있어서, 상기 이미지 코딩부는

상기 복수의 이미지 절편들 각각에 대하여 무손실 압축 알고리즘을 적용하여 상기 복수의 이미지 압축 절편들을 생성하는 것을 특징으로 하는 이미지 처리 장치.

청구항 3

제2항에 있어서, 상기 이미지 코딩부는

상기 복수의 이미지 절편들 각각의 첫 번째 이미지 픽셀 주소를 상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간과 매칭시켜 제1 매핑 정보를 생성한 후 주소 테이블에 저장하는 것을 특징으로 하는 이미지 처리 장치.

청구항 4

제3항에 있어서, 상기 이미지 코딩부는

상기 메모리의 상기 제2 주소공간으로부터 상기 이미지를 불러오는 과정에서, 상기 복수의 이미지 압축 절편들을 복원하여 상기 복수의 이미지 절편들을 생성하는 것을 특징으로 하는 이미지 처리 장치.

청구항 5

제1항에 있어서, 상기 메모리 연산부는

상기 복수의 이미지 압축 절편들을 차례대로 정렬한 후 상기 복수의 이미지 절편들에 대한 평균 압축 길이에 맞춰 그룹화함으로써 상기 버스트 그룹을 생성하는 것을 특징으로 하는 이미지 처리 장치.

청구항 6

제5항에 있어서, 상기 메모리 연산부는

상기 복수의 이미지 압축 절편들 중 특정 버스트 그룹 경계와 중첩되는 이미지 압축 절편의 경우 해당 버스트 그룹 경계 다음에 위치하는 버스트 그룹의 첫 번째 바이트에 정렬시키는 것을 특징으로 하는 이미지 처리 장치.

청구항 7

제5항에 있어서, 상기 메모리 연산부는

단일 버스트 그룹 내부에 존재하는 상기 복수의 이미지 압축 절편들 중 특정 바이트 경계와 중첩되는 이미지 압축 절편의 경우 해당 바이트 경계 다음에 위치하는 바이트에 정렬시키는 것을 특징으로 하는 이미지 처리 장치.

청구항 8

제1항에 있어서, 상기 메모리 연산부는

상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간을 상기 제2 주소공간과 매칭시켜 제2 매핑 정보를 생성한 후 주소 테이블에 저장하는 것을 특징으로 하는 이미지 처리 장치.

청구항 9

제1항에 있어서, 상기 메모리 연산부는

특정 버스트 그룹에 대한 메모리 주소에 해당하는 버스트 그룹 주소와 해당 버스트 그룹 내에서 각 이미지 압축 절편의 위치에 해당하는 이미지 압축 절편 주소를 이용하여 상기 제2 주소공간으로 변환하는 것을 특징으로 하는 이미지 처리 장치.

청구항 10

제9항에 있어서, 상기 메모리 연산부는

상기 버스트 그룹 주소를 기초로 상기 메모리로부터 특정 버스트 그룹을 읽고 상기 이미지 압축 절편 주소를 기초로 상기 버스트 그룹에 포함된 복수의 이미지 압축 절편들을 추출하는 것을 특징으로 하는 이미지 처리 장치.

청구항 11

이미지 처리 장치에서 수행되는 이미지 처리 방법에 있어서,

(a) 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 상기 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성하는 단계; 및

(b) 상기 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹을 생성하여 상기 메모리의 제2 주소공간으로 변환하는 단계를 포함하는 이미지 처리 방법.

청구항 12

제11항에 있어서, 상기 (a) 단계는

상기 복수의 이미지 절편들 각각의 첫 번째 이미지 픽셀 주소를 상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간과 매칭시켜 제1 매핑 정보를 생성하는 단계인 것을 특징으로 하는 이미지 처리 방법.

청구항 13

제12항에 있어서, 상기 (a) 단계는

상기 메모리의 상기 제2 주소공간으로부터 상기 이미지를 불러오는 과정에서, 상기 복수의 이미지 압축 절편들을 복원하여 상기 복수의 이미지 절편들을 생성하는 단계인 것을 특징으로 하는 이미지 처리 방법.

청구항 14

제11항에 있어서, 상기 (b) 단계는

상기 복수의 이미지 압축 절편들을 차례대로 정렬한 후 상기 복수의 이미지 절편들에 대한 평균 압축 길이에 맞춰 그룹화함으로써 상기 버스트 그룹을 생성하는 단계인 것을 특징으로 하는 이미지 처리 방법.

청구항 15

제11항에 있어서, 상기 (b) 단계는

상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간을 상기 제2 주소공간과 매칭시켜 제2 매핑 정보를 생성하는 단계인 것을 특징으로 하는 이미지 처리 방법.

청구항 16

제15항에 있어서, 상기 (b) 단계는

상기 버스트 그룹 주소를 기초로 상기 메모리로부터 특정 버스트 그룹을 읽고 상기 이미지 압축 절편 주소를 기초로 상기 버스트 그룹에 포함된 복수의 이미지 압축 절편들을 추출하는 단계인 것을 특징으로 하는 이미지 처리 방법.

발명의 설명

기술 분야

[0001] 본 발명은 이미지 처리 기술에 관한 것으로, 보다 상세하게는 효과적인 버스트 액세스 및 랜덤 액세스를 지원하는 무손실 이미지 데이터 압축이 가능한 이미지 처리 장치 및 방법에 관한 것이다.

배경 기술

[0003] 일반적으로, 손실 압축(lossy compression) 알고리즘은 이미지 데이터를 압축하면 프레임 버퍼 중에서 필요한 픽셀 데이터만 랜덤 액세스가 가능한 장점이 있지만, 원본 데이터가 손실되는 단점을 가지고 있다. 반면, 무손실 압축(lossless compression) 알고리즘은 이미지 데이터를 압축하게 되면 원본 데이터가 손실되지 않고 그대로 유지된다는 장점이 있지만, 압축된 후의 크기가 가변적이기 때문에 랜덤 액세스가 어렵고 외부메모리에서 액세스할 경우 효과적인 버스트 액세스가 어렵다는 단점을 가지고 있다.

[0004] 한국공개특허 제10-2007-0090165(2007.09.05)호는 디지털 이미지의 무손실 압축을 위한 시스템 및 방법에 관한 것으로, 3차원 중 임의의 곳에 위치하는 하나의 픽셀과 동일한 픽셀이 표시될 수 있게 함으로써, 즉, 같은 프레임, 또는 다른 프레임에 위치하는 픽셀과 동일한 픽셀이 표시될 수 있음으로써, 무손실 비디오 압축을 개선하기 위한 방법을 제공하는 것을 목적으로 한다.

[0005] 한국등록특허 제10-1100829(2011.09.16)호는 이미지 데이터 무손실 압축 및 해제하는 시스템 및 방법에 관한 것으로, 이미지 데이터의 픽셀 정보를 읽어서 투명색 또는 같은 색이 연속적인지 판단하여 연속적으로 존재하는 같은 픽셀에 대해서는 그 개수와 RGB 등을 저장함으로써 압축하고, 저장된 압축 파일은 Header에서 정보를 읽어 그에 따라 해제할 수 있는 효과를 제공한다.

선행기술문헌

특허문헌

- [0007] (특허문헌 0001) 한국공개특허 제10-2007-0090165(2007.09.05)호
(특허문헌 0002) 한국등록특허 제10-1100829(2011.09.16)호

발명의 내용

해결하려는 과제

- [0008] 본 발명의 일 실시예는 효과적인 버스트 액세스 및 랜덤 액세스를 지원하는 무손실 이미지 데이터 압축이 가능한 이미지 처리 장치 및 방법을 제공하고자 한다.
- [0009] 본 발명의 일 실시예는 무손실 압축 알고리즘으로 압축된 데이터를 그룹화하고 압축 데이터의 위치를 헤더 형태로 저장하여 랜덤 액세스를 지원할 수 있는 이미지 처리 장치 및 방법을 제공하고자 한다.
- [0010] 본 발명의 일 실시예는 압축된 이미지 블록 데이터들로 구성된 고정 길이의 버스트 그룹 단위로 메모리와의 데이터 교환이 가능한 이미지 처리 장치 및 방법을 제공하고자 한다.

과제의 해결 수단

- [0012] 실시예들 중에서, 이미지 처리 장치는 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 상기 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성하는 이미지 코딩부 및 상기 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹(Burst Group, BG)을 생성하여 상기 메모리의 제2 주소공간으로 변환하는 메모리 연산부를 포함한다.
- [0013] 상기 이미지 코딩부는 상기 복수의 이미지 절편들 각각에 대하여 무손실 압축 알고리즘을 적용하여 상기 복수의 이미지 압축 절편들을 생성할 수 있다.
- [0014] 상기 이미지 코딩부는 상기 복수의 이미지 절편들 각각의 첫 번째 이미지 픽셀 주소를 상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간과 매칭시켜 제1 매핑 정보를 생성한 후 주소 테이블에 저장할 수 있다.
- [0015] 상기 이미지 코딩부는 상기 메모리의 상기 제2 주소공간으로부터 상기 이미지를 불러오는 과정에서, 상기 복수의 이미지 압축 절편들을 복원하여 상기 복수의 이미지 절편들을 생성할 수 있다.
- [0016] 상기 메모리 연산부는 상기 복수의 이미지 압축 절편들을 차례대로 정렬한 후 상기 복수의 이미지 절편들에 대한 평균 압축 길이에 맞춰 그룹화함으로써 상기 버스트 그룹을 생성할 수 있다.
- [0017] 상기 메모리 연산부는 상기 복수의 이미지 압축 절편들 중 특정 버스트 그룹 경계와 중첩되는 이미지 압축 절편의 경우 해당 버스트 그룹 경계 다음에 위치하는 버스트 그룹의 첫 번째 바이트에 정렬시킬 수 있다.
- [0018] 상기 메모리 연산부는 단일 버스트 그룹 내부에 존재하는 상기 복수의 이미지 압축 절편들 중 특정 바이트 경계와 중첩되는 이미지 압축 절편의 경우 해당 바이트 경계 다음에 위치하는 바이트에 정렬시킬 수 있다.
- [0019] 상기 메모리 연산부는 상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간을 상기 제2 주소공간과 매칭시켜 제2 매핑 정보를 생성한 후 주소 테이블에 저장할 수 있다.
- [0020] 상기 메모리 연산부는 특정 버스트 그룹에 대한 메모리 주소에 해당하는 버스트 그룹 주소와 해당 버스트 그룹 내에서 각 이미지 압축 절편의 위치에 해당하는 이미지 압축 절편 주소를 이용하여 상기 제2 주소공간으로 변환할 수 있다.
- [0021] 상기 메모리 연산부는 상기 버스트 그룹 주소를 기초로 상기 메모리로부터 특정 버스트 그룹을 읽고 상기 이미지 압축 절편 주소를 기초로 상기 버스트 그룹에 포함된 복수의 이미지 압축 절편들을 추출할 수 있다.
- [0022] 실시예들 중에서, 이미지 처리 방법은 (a) 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 상기 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성하는 단계 및 (b) 상기 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹을 생성하여 상기 메모리의 제2 주소공간으로 변환하는 단계를 포함한다.
- [0023] 상기 (a) 단계는 상기 복수의 이미지 절편들 각각의 첫 번째 이미지 픽셀 주소를 상기 복수의 이미지 압축 절편

들 각각에 대한 상기 제1 주소공간과 매칭시켜 제1 매핑 정보를 생성하는 단계일 수 있다.

- [0024] 상기 (a) 단계는 상기 메모리의 상기 제2 주소공간으로부터 상기 이미지를 불러오는 과정에서, 상기 복수의 이미지 압축 절편들을 복원하여 상기 복수의 이미지 절편들을 생성하는 단계일 수 있다.
- [0025] 상기 (b) 단계는 상기 복수의 이미지 압축 절편들을 차례대로 정렬한 후 상기 복수의 이미지 절편들에 대한 평균 압축 길이에 맞춰 그룹화함으로써 상기 버스트 그룹을 생성하는 단계일 수 있다.
- [0026] 상기 (b) 단계는 상기 복수의 이미지 압축 절편들 각각에 대한 상기 제1 주소공간을 상기 제2 주소공간과 매칭시켜 제2 매핑 정보를 생성하는 단계일 수 있다.
- [0027] 상기 (b) 단계는 상기 버스트 그룹 주소를 기초로 상기 메모리로부터 특정 버스트 그룹을 읽고 상기 이미지 압축 절편 주소를 기초로 상기 버스트 그룹에 포함된 복수의 이미지 압축 절편들을 추출하는 단계일 수 있다.

발명의 효과

- [0029] 개시된 기술은 다음의 효과를 가질 수 있다. 다만, 특정 실시예가 다음의 효과를 전부 포함하여야 한다거나 다음의 효과만을 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0030] 본 발명의 일 실시예에 따른 이미지 처리 장치 및 방법은 무손실 압축 알고리즘으로 압축된 데이터를 그룹화하고 압축 데이터의 위치를 헤더 형태로 저장하여 랜덤 액세스를 지원할 수 있다.
- [0031] 본 발명의 일 실시예에 따른 이미지 처리 장치 및 방법은 압축된 이미지 블록 데이터들로 구성된 고정 길이의 버스트 그룹 단위로 메모리와의 데이터 교환이 가능할 수 있다.

도면의 간단한 설명

- [0033] 도 1은 본 발명의 일 실시예에 따른 이미지 처리 시스템을 설명하는 도면이다.
- 도 2는 도 1에 있는 이미지 처리 장치를 설명하는 블록도이다.
- 도 3은 도 1에 있는 이미지 처리 장치에서 수행되는 이미지 처리 과정을 설명하는 순서도이다.
- 도 4는 도 1에 있는 이미지 처리 장치에서 사용하는 주소 테이블의 저장 공간을 줄이기 위한 프레임 버퍼 파티셔닝 포맷을 설명하는 예시도이다.
- 도 5는 메모리에 있는 원래 파티션된 그룹 데이터와 메모리에 저장된 압축 블록 데이터를 설명하는 예시도이다.
- 도 6은 1920 × 1080 풀 HD 해상도 이미지의 주소 테이블의 일 실시예를 설명하는 도면이다.
- 도 7은 도 1에 있는 이미지 처리 장치의 전체적인 구조를 설명하는 개념도이다.
- 도 8은 도 1에 있는 이미지 처리 장치에서 생성하는 버스트 그룹의 포맷을 설명하는 예시도이다.
- 도 9는 도 1에 있는 이미지 처리 장치에서 버스트 그룹을 패킹하는 과정을 설명하는 예시도이다.
- 도 10은 본 발명의 일 실시예에 따른 이미지 처리 장치에 포함된 Write Cache를 설명하는 개념도이다.
- 도 11은 본 발명의 일 실시예에 따른 이미지 처리 장치에 사용하는 주소들 간의 관계를 설명하는 도면이다.
- 도 12은 본 발명의 일 실시예에 따른 이미지 처리 장치에 포함된 Read Cache를 설명하는 개념도이다.

발명을 실시하기 위한 구체적인 내용

- [0034] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해

되어서는 아니 될 것이다.

- [0035] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0036] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0037] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0038] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다"또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0039] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0040] 본 발명은 컴퓨터가 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현될 수 있고, 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- [0041] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0043] 도 1은 본 발명의 일 실시예에 따른 이미지 처리 시스템을 설명하는 도면이다.
- [0044] 도 1을 참조하면, 이미지 처리 시스템(100)은 사용자 단말(110), 이미지 처리 장치(130) 및 데이터베이스(150)를 포함할 수 있다.
- [0045] 사용자 단말(110)은 이미지 처리의 대상이 되는 이미지를 제공할 수 있는 컴퓨팅 장치에 해당할 수 있다. 사용자 단말(110)은 스마트폰, 노트북 또는 컴퓨터로 구현될 수 있으며, 반드시 이에 한정되지 않고, 태블릿 PC 등 다양한 디바이스로도 구현될 수 있다. 사용자 단말(110)은 이미지 처리 장치(130)와 네트워크를 통해 연결될 수 있고, 복수의 사용자 단말(110)들은 이미지 처리 장치(130)와 동시에 연결될 수 있다.
- [0046] 이미지 처리 장치(130)는 무손실 압축이 가지는 데이터의 손실이 없는 장점을 유지하면서 버스(bus)의 버스트 액세스 단위로 그룹화(grouping)하여 효과적인 버스트 액세스 및 랜덤 액세스를 지원할 수 있는 컴퓨터 또는 프로그램에 해당하는 서버로 구현될 수 있다. 이미지 처리 장치(130)는 사용자 단말(110)과 유선 네트워크 또는 블루투스, WiFi 등과 같은 무선 네트워크로 연결될 수 있고, 유선 또는 무선 네트워크를 통해 사용자 단말(110)과 통신을 수행할 수 있다. 일 실시예에서, 이미지 처리 장치(130)는 데이터베이스(150)와 연동하여 이미지 처리를 수행할 수 있고, 다양한 정보들을 관리할 수 있다. 한편, 이미지 처리 장치(130)는 도 1과 달리, 데이터베이스(150)를 내부에 포함하여 구현될 수 있다.
- [0047] 데이터베이스(150)는 이미지 처리 장치(130)가 이미지를 블록 단위로 분할하고 압축한 후 패킹을 통해 그룹화하는 과정에서 필요한 다양한 정보들을 저장할 수 있다. 예를 들어, 데이터베이스(150)는 사용자 단말(110)로부터 수신한 이미지 데이터를 저장할 수 있고, 이미지를 압축하여 생성한 그룹 정보를 저장할 수 있으며, 반드시 이

에 한정되지 않고, 이미지를 수신하여 그룹 단위로 압축 및 복원을 수행하는 과정에서 다양한 형태로 수집 또는 가공된 정보들을 저장할 수 있다.

[0049] 도 2는 도 1에 있는 이미지 처리 장치를 설명하는 블록도이다.

[0050] 도 2를 참조하면, 이미지 처리 장치(130)는 이미지 코딩부(210), 메모리 연산부(230) 및 제어부(250)를 포함할 수 있다.

[0051] 이미지 코딩부(210)는 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성할 수 있다. 보다 구체적으로, 이미지 코딩부(210)는 원래의 이미지를 특정 사이즈의 이미지 절편으로 분할할 수 있고, 압축하여 이미지 압축 절편을 생성할 수 있다. 예를 들어, 이미지 코딩부(210)는 원래의 이미지를 4×4 , 8×8 또는 16×8 사이즈 등에 해당하는 이미지 절편들로 분할할 수 있고, 만약 이미지의 크기가 800×600 에 해당하는 경우 4×4 사이즈의 이미지 절편 300개(20×15)로 분할할 수 있다.

[0052] 여기에서, 이미지 절편은 이미지를 구성하는 부분 이미지로서 픽셀 블록에 해당할 수 있고 이미지 압축 절편은 픽셀 블록을 압축하여 생성된 압축 픽셀 블록에 해당할 수 있다. 한편, 이미지 처리 장치(130)는 이미지 코딩부(210)를 통해 생성된 이미지 압축 절편들을 메모리의 제1 주소공간에 저장할 수 있다. 여기에서, 제1 주소공간은 이미지 압축 절편들이 저장되는 실제 메모리 상의 주소에 해당할 수 있고, 이미지 압축 절편들 각각에 대응될 수 있으며, 각각의 이미지 압축 절편과 연관되어 데이터베이스(150)에 저장될 수 있다. 이미지 처리 장치(130)에서 사용되는 이미지는 사진과 같은 정지 화면에 해당하거나 또는 동영상과 같은 연속된 화면의 한 프레임(Frame)에 해당할 수 있다.

[0053] 일 실시예에서, 이미지 코딩부(210)는 복수의 이미지 절편들 각각에 대하여 무손실 압축 알고리즘을 적용하여 복수의 이미지 압축 절편들을 생성할 수 있다. 무손실 압축 알고리즘은 이미지 압축에 있어서 압축 후 복원시에 원래의 정보를 그대로 유지하는 압축 방법에 해당할 수 있다. 예를 들어, 이미지 코딩부(210)는 2차원의 이미지 절편을 압축하여 1차원의 이미지 압축 절편을 생성할 수 있고, 1차원으로 압축된 이미지 압축 절편들은 순서대로 연결되어 메모리 상의 제1 주소공간에 저장될 수 있다. 다른 실시예에서, 이미지 코딩부(210)는 무손실 압축 알고리즘으로서 DPCM(Differential Pulse Code Modulation)으로 prediction을 한 후 $K=0, 1, 2, 3$ 에 대해서 골롬라이스 인코딩(golomb-rice encoding)을 하는 DPCM-GR 알고리즘을 이용하여 데이터를 압축할 수 있다.

[0054] 일 실시예에서, 이미지 코딩부(210)는 복수의 이미지 절편들 각각의 첫 번째 이미지 픽셀 주소를 복수의 이미지 압축 절편들 각각에 대한 제1 주소공간과 매칭시켜 제1 매핑 정보를 생성할 수 있다. 이미지 코딩부(210)는 복수의 이미지 절편들을 메모리에 저장하는 과정에서 이미지 절편의 첫 번째 픽셀이 저장되는 메모리 상의 주소를 주소 테이블(address table)에 저장할 수 있다. 또한, 이미지 코딩부(210)는 이미지 압축 블록이 저장되는 메모리 상의 주소를 할당할 수 있고, 할당된 주소는 각 이미지 압축 절편에 대한 제1 주소공간에 해당할 수 있다. 이미지 코딩부(210)는 이미지 절편에 할당된 메모리 주소와 압축을 통해 생성된 이미지 압축 절편에 할당된 제1 주소공간을 서로 매칭시켜 제1 매핑 정보를 생성할 수 있고 주소 테이블에 저장하여 보관할 수 있다.

[0055] 일 실시예에서, 이미지 코딩부(210)는 메모리의 제2 주소공간으로부터 이미지를 불러오는 과정에서, 복수의 이미지 압축 절편들을 복원하여 복수의 이미지 절편들을 생성할 수 있다. 이미지 코딩부(210)는 이미지 압축 절편을 기초로 압축을 해제하여 이미지 절편을 생성할 수 있으며, 이미지 압축 절편에서 이미지 절편으로의 복원 과정은 압축 과정과 반대로 수행될 수 있다. 또한, 이미지 코딩부(210)는 이미지 복원 과정에서 주소 테이블에 저장된 주소 정보들을 활용할 수 있다.

[0056] 메모리 연산부(230)는 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹(Burst Group, BG)을 생성하여 메모리의 제2 주소공간으로 변환할 수 있다. 여기에서, 버스트 그룹(Burst Group, BG)은 효과적인 메모리 액세스를 위하여 이미지 압축 절편들을 일정 단위로 묶은 이미지 압축 절편들의 집합에 해당할 수 있다. 하나의 이미지는 복수의 이미지 절편들로 분할될 수 있고, 각 이미지 절편은 이미지 압축 절편으로 압축될 수 있으며, 분할된 이미지 절편의 수 및 이미지 압축 절편으로 압축되는 압축률에 따라 하나의 이미지로부터 생성되는 버스트 그룹의 수는 가변적일 수 있다. 제2 주소공간은 버스트 그룹이 저장되는 메모리 상의 실제 주소와 해당 버스트 그룹 안에서 각 이미지 압축 절편의 위치 정보를 포함할 수 있다.

[0057] 일 실시예에서, 메모리 연산부(230)는 복수의 이미지 압축 절편들을 차례대로 정렬한 후 복수의 이미지 절편들에 대한 평균 압축 길이에 맞춰 그룹화함으로써 버스트 그룹을 생성할 수 있다. 메모리 연산부(230)는 이미지

절편의 정렬 순서에 맞춰 이미지 압축 절편을 차례대로 정렬할 수 있다. 하나의 이미지로부터 생성되는 버스트 그룹의 수는 가변적일 수 있으며, 버스트 그룹의 길이 자체는 고정적일 수 있다. 따라서, 메모리 연산부(230)는 버스트 그룹을 생성하기 전에 버스트 그룹의 길이를 결정할 수 있고, 버스트 그룹의 길이는 이미지 압축 절편의 평균 압축 길이와 동일하게 결정할 수 있다. 이미지 압축 절편의 압축률(Compression ratio)에 따라 하나의 버스트 그룹에 포함되는 이미지 압축 절편의 수 역시 가변적일 수 있다.

[0058] 일 실시예에서, 메모리 연산부(230)는 복수의 이미지 압축 절편들 중 특정 버스트 그룹 경계와 중첩되는 이미지 압축 절편의 경우 해당 버스트 그룹 경계 다음에 위치하는 버스트 그룹의 첫 번째 바이트에 정렬시킬 수 있다. 메모리 연산부(230)는 이미지 압축 절편을 순서대로 정렬시킬 수 있고, 버스트 그룹 길이에 따라 그룹화하는 과정에서 특정 이미지 압축 절편이 버스트 그룹의 경계와 중첩되는 경우 해당 이미지 압축 절편을 해당 버스트 그룹에서 제외하고 해당 버스트 그룹의 다음 순서에 위치하는 버스트 그룹의 첫 번째 이미지 압축 절편으로서 사용할 수 있다. 이를 위하여, 메모 연산부(230)는 버스트 그룹의 경계와 중첩하는 이미지 압축 절편을 다음 버스트 그룹의 첫 번째 바이트(byte)부터 시작하도록 정렬시킬 수 있다.

[0059] 일 실시예에서, 메모리 연산부(230)는 단일 버스트 그룹 내부에 존재하는 복수의 이미지 압축 절편들 중 특정 바이트 경계와 중첩되는 이미지 압축 절편의 경우 해당 바이트 경계 다음에 위치하는 바이트에 정렬시킬 수 있다. 하나의 버스트 그룹은 복수의 이미지 압축 절편들의 집합에 해당하고, 복수의 이미지 압축 절편들은 첫 번째 이미지 압축 절편부터 차례대로 연결되어 하나의 버스트 그룹을 형성할 수 있다. 이 때, 메모리 연산부(230)는 버스트 그룹에 관한 메모리 주소를 저장하는 주소 테이블의 저장 공간을 줄이기 위하여 하나의 버스트 그룹 내에서의 복수의 이미지 압축 절편들의 위치를 비트(bit) 단위가 아닌 바이트(byte) 단위로 정렬시킬 수 있다.

[0060] 예를 들어, 메모리 연산부(230)는 하나의 버스트 그룹을 생성하기 위하여 버스트 그룹의 첫 번째 바이트와 첫 번째 이미지 압축 절편을 정렬시킬 수 있고, 첫 번째 이미지 압축 절편의 뒤에 두 번째 이미지 압축 절편을 연결할 수 있다. 만약 두 번째 이미지 압축 절편이 바이트 경계에서 끝나는 경우 첫 번째 이미지 압축 절편의 뒤에 곧바로 연결되어 배치될 수 있지만, 두 번째 이미지 압축 절편이 바이트 경계와 중첩되는 경우에는 첫 번째 이미지 압축 절편의 뒤에 곧바로 연결될 수 없고 첫 번째 이미지 압축 절편이 끝나는 지점에서 처음으로 나타나는 바이트 경계에 정렬될 수 있다.

[0061] 일 실시예에서, 메모리 연산부(230)는 복수의 이미지 압축 절편들 각각에 대한 제1 주소공간을 제2 주소공간과 매칭시켜 제2 매핑 정보를 생성할 수 있다. 메모리 연산부(230)는 이미지 압축 절편이 저장되는 메모리 상의 주소인 제1 주소공간과 버스트 그룹에 관한 제2 주소공간을 매칭시킬 수 있다. 보다 구체적으로, 제1 주소공간은 각 이미지 압축 절편과 해당 이미지 압축 절편에 대한 메모리 주소를 포함할 수 있고, 제2 주소공간은 각 버스트 그룹과 해당 버스트 그룹에 대한 메모리 주소 및 해당 버스트 그룹에 속한 이미지 압축 절편들에 대한 주소를 포함할 수 있다. 따라서, 메모리 연산부(230)는 버스트 그룹 내에 속하는 이미지 압축 절편을 기초로 제1 주소공간과 제2 주소공간을 매칭시킴으로써 제2 매핑 정보를 생성할 수 있고 주소 테이블에 저장하여 보관할 수 있다.

[0062] 일 실시예에서, 메모리 연산부(230)는 특정 버스트 그룹에 대한 메모리 주소에 해당하는 버스트 그룹 주소와 해당 버스트 그룹 내에서 각 이미지 압축 절편의 위치에 해당하는 이미지 압축 절편 주소를 이용하여 제2 주소공간으로 변환할 수 있다. 보다 구체적으로, 메모리 연산부(230)는 버스트 그룹 주소와 이미지 압축 절편 주소 간의 합(sum)으로서 제2 주소공간을 표현할 수 있다.

[0063] 일 실시예에서, 메모리 연산부(230)는 버스트 그룹 주소를 기초로 메모리로부터 특정 버스트 그룹을 읽고 이미지 압축 절편 주소를 기초로 버스트 그룹에 포함된 복수의 이미지 압축 절편들을 추출할 수 있다. 메모리 연산부(230)는 버스트 그룹 주소에 대응되는 메모리 상의 위치로부터 버스트 그룹에 대한 데이터를 읽을 수 있고, 이미지 압축 절편 주소를 통해 해당 버스트 그룹 내에 위치한 각 이미지 압축 절편들을 분리할 수 있다.

[0064] 제어부(250)는 이미지 처리 장치(130)의 전체적인 동작을 제어하고, 이미지 코딩부(210) 및 메모리 연산부(230) 간의 제어 흐름 또는 데이터 흐름을 관리할 수 있다.

[0066] 도 3은 도 1에 있는 이미지 처리 장치에서 수행되는 이미지 처리 과정을 설명하는 순서도이다.

[0067] 도 3을 참조하면, 이미지 처리 장치(130)는 이미지 코딩부(210)를 통해 이미지를 메모리의 제1 주소공간에 저장하는 과정에서, 이미지를 복수의 이미지 절편들로 분할한 후 압축을 통해 복수의 이미지 압축 절편들을 생성할

수 있다(단계 S310). 이미지 처리 장치(130)는 메모리 연산부(230)를 통해 복수의 이미지 압축 절편들의 패킹을 통해 가변적인 버스트 그룹을 생성하여 메모리의 제2 주소공간으로 변환할 수 있다(단계 S330).

- [0069] 도 4는 도 1에 있는 이미지 처리 장치에서 사용하는 주소 테이블의 저장 공간을 줄이기 위한 프레임 버퍼 파티셔닝 포맷을 설명하는 예시도이다.
- [0070] 도 4를 참조하면, 프레임 버퍼(frame buffer)가 아닌 파티션 그룹(partition group) 내의 매크로 블록(macro block)들의 주소는 매핑(mapping)되기 때문에 주소 테이블의 저장 공간을 줄일 수 있다. $m \times n$ 해상도를 가지는 이미지의 경우 각 파티션 그룹의 크기는 $m \times 16$ 에 해당하고 파티션 그룹 수는 실링(ceiling)($n/\text{매크로 블록의 높이}$)을 사용하여 산출될 수 있다. 파티션 그룹 내의 16×16 매크로 블록은 길이 1 비트의 압축 블록 데이터(Compressed Block Data, CBD)로 압축된 다음 해당 파티션 그룹 시작 주소에서 순차적으로 저장될 수 있다. 여기에서, 파티션 그룹 내의 압축된 블록의 주소는 주소 테이블(address table)에 저장될 수 있다.
- [0072] 도 5는 메모리에 있는 원래 파티션된 그룹 데이터와 메모리에 저장된 압축 블록 데이터를 설명하는 예시도이다.
- [0073] 도 5를 참조하면, 이미지 처리 장치(130)는 2차원(2-D) 매크로 블록 데이터를 압축하여 1차원(1-D) CBD로 변환할 수 있다. 이미지 처리 장치(130)는 파티션 그룹의 첫 번째 매크로 블록 데이터를 파티션 주소의 시작 부분에 저장된 첫 번째 CBD로 압축할 수 있다. 또한, 이미지 처리 장치(130)는 나머지 매크로 블록 데이터에 대해 CBD로 압축하여 다음 주소에 순차적으로 저장할 수 있다. 이미지 처리 장치(130)는 CBD를 연속적으로 저장함으로써 버스트 액세스가 가능하도록 할 수 있다. 또한, 이미지 처리 장치(130)는 CBD를 저장하는 동안 CBD에 대한 랜덤 액세스를 가능하게 하기 위해 각 데이터의 시작 위치를 저장할 수 있다.
- [0075] 도 6은 1920×1080 풀 HD 해상도 이미지의 주소 표의 일 실시예를 설명하는 도면이다.
- [0076] 도 6을 참조하면, 이미지 처리 장치(130)에서 수행되는 주소 매핑은 파티션 그룹 내부의 CBD를 다루기 때문에 각 CBD 조각의 주소는 15 비트가 필요하며 x 좌표는 11 비트, y 좌표는 4 비트에 해당할 수 있다(그림 (a)). 따라서, 총 8,100 개의 매크로 블록 항목이 있는 풀 HD 해상도 이미지의 경우 약 16KB의 주소 테이블 저장 공간이 필요할 수 있다. 결과적으로, 이미지 처리 장치(130)는 필요 저장 공간을 해당 프레임의 모든 CBD에 임의 액세스를 허용하는데 필요한 23KB의 저장 공간의 약 68%로 줄일 수 있다. 도 6에서, 주소 테이블의 크기는 x 좌표의 이미지 해상도 비트 수에 비례하지만 y 좌표의 이미지 해상도 비트 수와는 무관할 수 있다(그림 (b)). 따라서, 주소 테이블에 대한 저장 용량의 감소 비율은 이미지 해상도에 따라 증가할 수 있다.
- [0078] 도 7은 본 발명의 일 실시예에 따른 이미지 처리 장치의 전체적인 구조를 설명하는 개념도이다.
- [0079] 도 7을 참조하면, 이미지 처리 장치(130)는 이미지 코딩부(210)와 메모리 연산부(230)를 포함할 수 있고, 이미지 코딩부(210)는 Video decoder, Compressor 및 Decompressor를 포함하여 구현될 수 있으며, 메모리 연산부(230)는 Write Cache 및 Read Cache를 포함하여 구현될 수 있다.
- [0080] Video decoder는 프레임 버퍼(frame buffer)를 매크로 블록(macro block) 단위로 메모리에 저장할 수 있고, 메모리에 저장된 매크로 블록 데이터를 읽어 프레임 버퍼로 옮기는 동작을 수행할 수 있다. Compressor는 Video decoder로부터 수신한 매크로 블록 데이터(original block data)를 압축하여 압축 블록 데이터(compressed block data)를 생성할 수 있고 Write Cache에 제공할 수 있다. Decompressor는 Read Cache로부터 수신한 압축 블록 데이터를 복원하여 매크로 블록 데이터를 Video decoder에 제공할 수 있다.
- [0081] Write Cache는 Compressor로부터 수신한 압축 블록 데이터를 기초로 버스트 그룹을 생성하여 버스트 그룹 단위로 메모리에 저장(burst write)할 수 있다. 이때 변경된 위치 정보는 주소 테이블(address table)에 저장될 수 있다. Read Cache는 Video decoder가 매크로 블록 데이터(original block data)를 랜덤으로 읽는 경우 주소 테이블에 저장된 위치 정보를 기초로 버스트 그룹 단위로 데이터를 읽을(burst read) 수 있고 Decompressor에게 압축 블록 데이터를 제공할 수 있다.
- [0082] Write Cache 및 Read cache는 memory bus access 효율을 높이기 위해서 메모리의 데이터를 버스트 액세스할 수

있다. 이를 위해서 Write Cache는 압축 블록 데이터를 버스트 그룹 단위로 패킹하고 이를 메모리에 burst write 할 수 있다. 또한, Write Cache의 버퍼는 더블(double) 구조로 구현될 수 있기 때문에 패킹과 burst write를 동시에 수행할 수 있다. Read Cache는 버스트 그룹 단위로 캐시 라인(cache line)을 갱신할 수 있고, 캐시 라인에 저장된 데이터를 재사용할 수 있다. 만약 캐시 라인을 갱신할 때 모든 버퍼가 유효 데이터를 가진 경우, Read Cache는 저장된 시간의 순서대로 캐시 라인을 대체할 수 있다. Read Cache는 이미지의 컬러 요소(color component) 별로 독립된 캐시 라인을 가질 수 있다.

[0083] 도 7에서, original block data는 압축과 패킹 과정을 통해 대응되는 address도 두 차례 변경될 수 있다. 첫째로, original block data의 address인 XY는 데이터를 압축한 후 address XY'로 변경될 수 있다. 둘째로, 버스트 그룹에 복수의 compressed block data를 패킹하면 address XY'가 address XY''로 변경될 수 있다. Address XY''는 burst group address(BG address)와 compressed block address(CB address)의 합으로 표현될 수 있다. 여기에서, BG address는 메모리에 저장된 burst group data의 주소에 해당할 수 있고, CB address는 버스트 그룹 안에서 각 compressed block data의 위치에 해당할 수 있다. XY'에서 XY''으로 매핑이 완료된 경우 매핑 정보는 주소 테이블에 저장될 수 있다.

[0085] 도 8은 도 1에 있는 이미지 처리 장치에서 생성하는 버스트 그룹의 포맷을 설명하는 예시도이다.

[0086] 도 8을 참조하면, 이미지 처리 장치(130)는 압축 블록 데이터를 기초로 버스트 그룹을 생성할 수 있다. 버스트 그룹은 압축 블록 데이터들의 집합으로서 특정 형식의 포맷을 통해 형성될 수 있다. 버스트 그룹의 길이는 compressed macro block data의 평균 크기인 1 byte로 고정될 수 있고, 버스트 그룹의 내용은 compressed block data를 변수 n개만큼 포함할 수 있다.

[0087] 여기에서, 변수 n은 압축률에 비례해서 증가할 수 있지만 compressed block data를 정렬하면서 줄어들 수 있다. 이미지 처리 장치(130)는 데이터를 정렬할 때 메모리 읽기 버스(read bus)의 효율을 높이기 위해서 compressed block data가 버스트 그룹의 경계(boundary)를 넘지 않도록 할 수 있다. 즉, 이미지 처리 장치(130)는 버스트 그룹에서 1st compressed block data를 1st byte에 정렬할 수 있다. 또한, 이미지 처리 장치(130)는 주소 테이블의 저장 공간을 줄이기 위해 CB address를 bit address가 아닌 byte address에 정렬할 수 있다. 따라서, m번째 compressed block data는 j번째 byte에 정렬될 수 있다.

[0089] 도 9는 도 1에 있는 이미지 처리 장치에서 버스트 그룹을 패킹하는 과정을 설명하는 예시도이다.

[0090] 도 9를 참조하면, 이미지 처리 장치(130)는 16x4 형태로 데이터를 압축하고 평균 압축률이 2인 경우 8개의 블록 데이터를 버스트 그룹으로 패킹할 수 있다. 각 블록 데이터는 3, 1.51, 3, 1.5, 2, 2, 2 및 2의 압축률(Compression Ratio, CR)을 가질 수 있다. 이러한 조건에서 버스트 그룹은 16x16 macro block data의 평균 압축 길이인 128 byte 길이를 가질 수 있다. 이미지 처리 장치(130)는 8개의 original block data를 128 byte 길이로 패킹하여 3개의 버스트 그룹을 생성할 수 있다.

[0091] 특히, 이미지 처리 장치(130)는 버스트 그룹 생성 과정에서 3번의 데이터 정렬을 수행할 수 있다. 도 9에서, 3번째 compressed block data는 바이트 경계(byte boundary)와 중첩(crossing)되기 때문에 다음 바이트(next byte)에 정렬될 수 있고, 네 번째와 일곱 번째 compressed block data는 버스트 그룹 경계(burst group boundary)와 중첩되기 때문에 다음 버스트 그룹의 첫 번째 바이트에 정렬될 수 있다.

[0093] 도 10은 본 발명의 일 실시예에 따른 이미지 처리 장치에 포함된 Write Cache를 설명하는 개념도이다.

[0094] 도 10을 참조하면, 이미지 처리 장치(130)는 Write Cache를 포함하여 구현될 수 있고, Write Cache의 구조는 도 10의 좌측 그림과 같고 address aligner의 상세 구조는 도 10의 우측 그림과 같을 수 있다. 도 10의 좌측 그림에서, AddrAligner는 XY와 bit length를 이용해서 switch BG와 각종 address들을 계산하여 모든 component들의 데이터 위치를 제어할 수 있다. Shifter는 compressed block data를 CB address만큼 쉬프트(shift)할 수 있고, Selector0은 switch BG가 활성화(enable)되면 compressed block data를 저장하는 BG buffer를 바꿀 수 있다. Select1은 valid BG가 가리키는 BG data를 BG address가 가리키는 메모리에 burst write할 수 있다.

[0095] 도 10의 우측 그림에서, ByteAligner는 XY를 byte address와 bit address으로 분리할 수 있다. Bit address 값

에 따라 byte address가 곧 byte aligned address가 되거나 1 증가한 값이 byte aligned address가 될 수 있다. Byte aligned address는 버스트 그룹 크기 단위로 분리될 수 있다(split). 상위 address는 BG address를 구하는데 사용될 수 있고 하위 address는 CB address를 구하는데 사용될 수 있다.

[0096] 만약 switch BG가 활성화(enable)된 경우 CB address는 clear되고 BG address는 1만큼 증가할 수 있다. switch BG는 current compressed block data를 next BG에 패킹할 때 활성화될 수 있다. 즉, Byte aligned address와 byte aligned length를 더한 값이 버스트 그룹 길이인 1 byte보다 크면 활성화될 수 있다. byte aligned length는 byte aligned address와 마찬가지로 방법으로 bit length를 바이트 정렬하여 결정될 수 있다.

[0098] 도 11은 본 발명의 일 실시예에 따른 이미지 처리 장치에 사용하는 주소들 간의 관계를 설명하는 도면이다.

[0099] 도 11을 참조하면, 이미지 처리 장치(130)는 BG 길이가 128 바이트인 경우 XY'를 XY''로 변환할 수 있다. XY'는 길이가 35 비트인 비트 주소(bit address)에 해당하고, XY'는 35'h7_0123_4567이며 XY'의 바이트 정렬 주소(byte-aligned address)는 32'hE014_68AD에 해당할 수 있다. switch BG가 비활성화(disable)되면 바이트 정렬 주소의 상위 25 비트는 BG 주소(25'h1C0_48D1)에 해당될 수 있고 바이트 정렬 주소의 하위 7 비트는 CB 주소(7'h2D)에 해당할 수 있다. 여기에서, 비트 주소 XY''는 35'h7_0123_4568에 해당할 수 있다. switch BG가 활성화(enable)되면 BG 주소는 기존 BG 주소를 1씩 증가시켜 얻을 수 있고(즉, 25'h1C0_48D2), CB 주소는 0으로 초기화될 수 있다. 따라서 XY''는 35'h7_0123_4800이 될 수 있다. 만약 XY'가 바이트 주소(byte address)인 경우 바이트 정렬 주소에 관한 계산은 건너뛸 수 있고, switch BG에 관한 주소 계산 만이 수행될 수 있다.

[0101] 도 12는 본 발명의 일 실시예에 따른 이미지 처리 장치에 포함된 Read Cache를 설명하는 개념도이다.

[0102] 도 12를 참조하면, 이미지 처리 장치(130)는 Read Cache를 포함하여 구현될 수 있고, Read Cache의 구조는 도 12와 같을 수 있다. 도 12에서, Splitter는 XY address를 BG address와 CB address로 분리할 수 있다. Shifter는 BG data에서 CB address가 가리키는 compressed block data를 선택하여 전송할 수 있다. Selector0는 버퍼의 BGnAddr field들 중에서 BG address를 찾을 수 있고, 해당 버퍼의 BG data를 선택할 수 있다. Match component는 BGnAddr field들에 BG address가 없으면, BG address의 데이터를 메모리에 요청할 수 있다. Read Storage component는 BGAddr과 BG data로 구성된 버퍼들로 구성될 수 있다. Read storage는 color component 별로 각각 분리될 수 있고, 해당 버퍼 엔트리(entry)는 서로 다를 수 있다. Selector1은 메모리에서 읽어들이는 BG data를 라운드로빈(round robin) 순서에 따라 Read Storage 내에 존재하는 버퍼를 갱신할 수 있다.

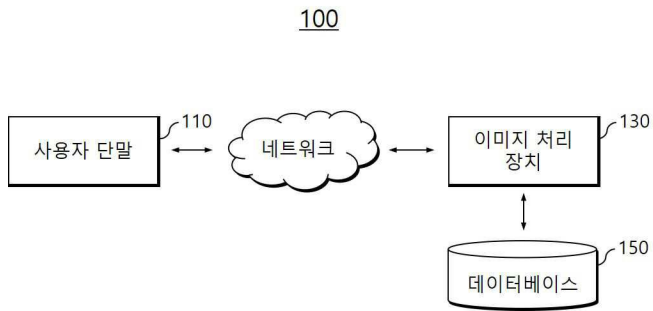
[0104] 상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

부호의 설명

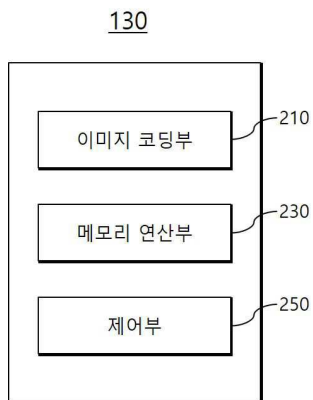
[0106] 100: 이미지 처리 시스템
110: 사용자 단말 130: 이미지 처리 장치
150: 데이터베이스
210: 이미지 코딩부 230: 메모리 연산부
250: 제어부

도면

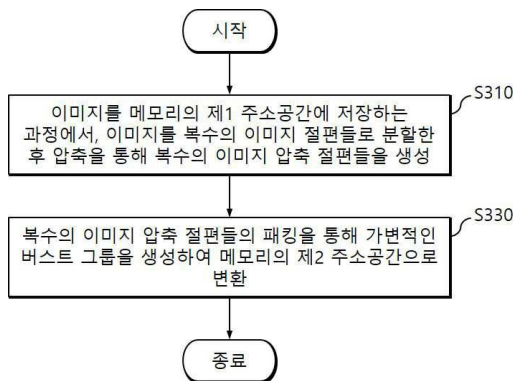
도면1



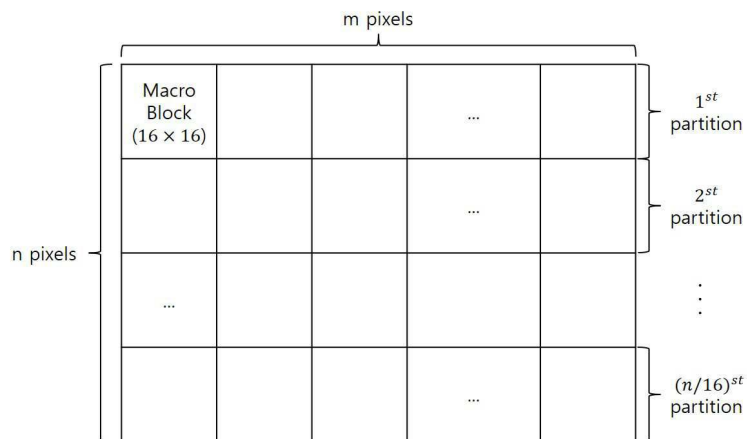
도면2



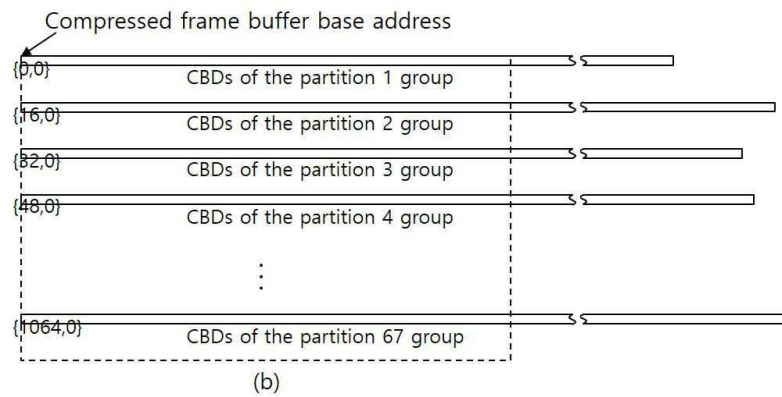
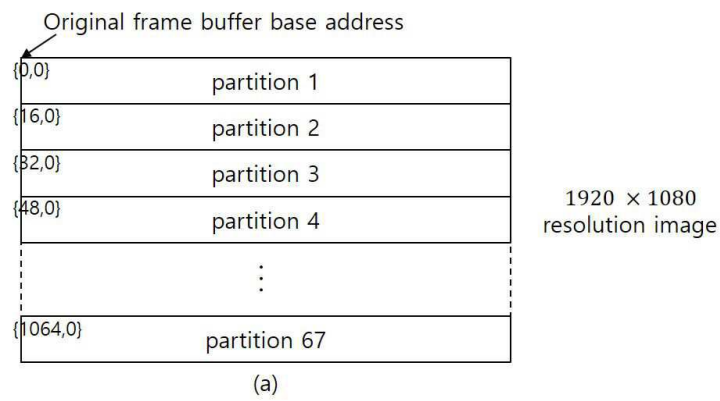
도면3



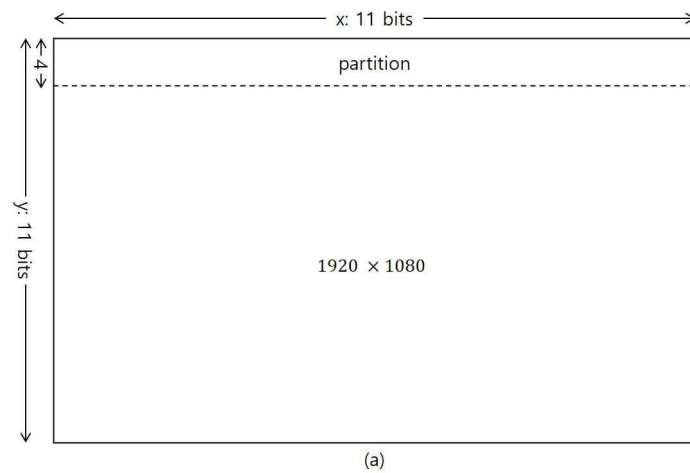
도면4



도면5



도면6



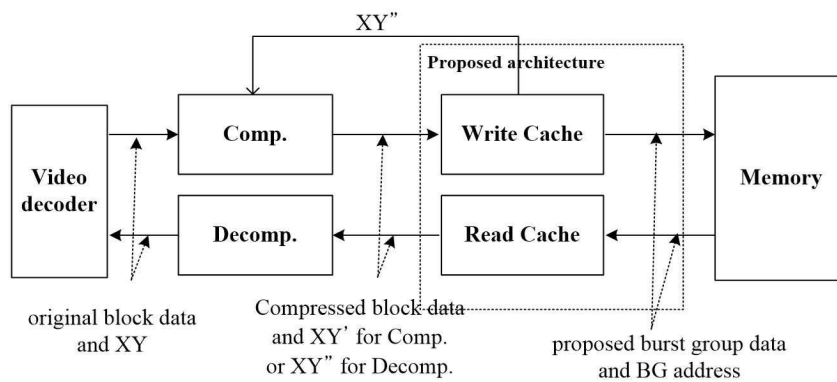
(a)

←15 bits→

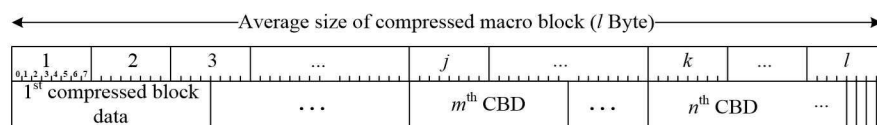
{0,0}	{1,0}	{2,0}	{3,0}	{4,0}	{5,0}	...	{119,0}
{0,1}	{1,1}	{2,1}	{3,1}	{4,1}	{5,1}	...	{119,0}
{0,2}	{1,2}	{2,2}	{3,2}	{4,2}	{5,2}	...	{119,0}
{0,3}	{1,3}	{2,3}	{3,3}	{4,3}	{5,3}	...	{119,0}
{0,4}	{1,4}	{2,4}	{3,4}	{4,4}	{5,4}	...	{119,0}
...	...	...	...	...	...	...	...
{0,66}	{1,66}	{2,66}	{3,66}	{4,66}	{5,66}	...	{119,0}

(b)

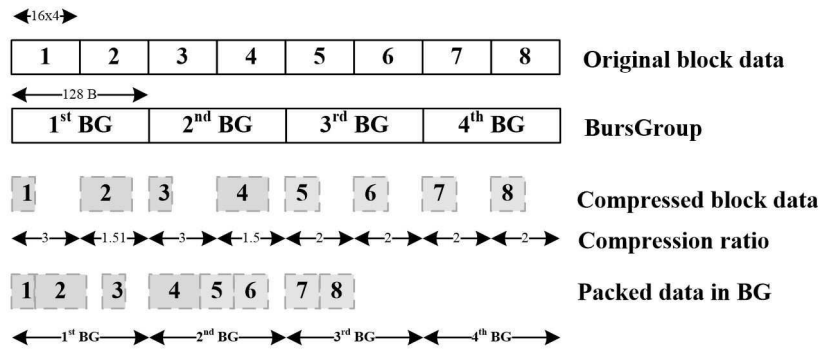
도면7



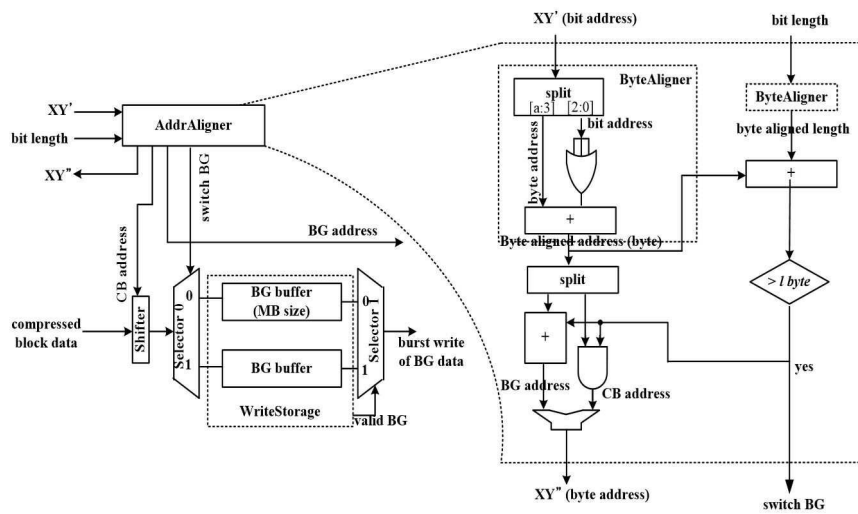
도면8



도면9



도면10



도면11

XY' (bit address)	[31:0, 2:0]	35'h7_0123_4567	
byte aligned address	[31:0]	32'hE014_68AD	
switch BG	enable	0	1
XY'' (bit address)	[31:0, 2:0]	35'h7_0123_4568	35'h7_0123_4800
BG address	[31:7]	25'h1C0_48D1	25'h1C0_48D2
CB address	[6:0]	7'h2D	7'h00

도면12

