(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0043506
(43) 공개일자 2016년04월21일

(51) 국제특허분류(Int. Cl.)
H04N 19/103 (2014.01) *H04N 19/107* (2014.01)
H04N 19/59 (2014.01)
(52) CPC특허분류
H04N 19/103 (2015.01)
H04N 19/107 (2015.01)
(21) 출원번호 10-2015-0140564
(22) 출원일자 2015년10월06일
심사청구일자 없음
(30) 우선권주장
1020140137917 2014년10월13일 대한민국(KR)

(71) 출원인
한국전자통신연구원
대전광역시 유성구 가정로 218 (가정동)
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
방건
대전광역시 유성구 지족로 362, 303동 303호 (지족동, 반석마을3단지아파트)
박광훈
경기도 성남시 분당구 수내로 201, 402동 402호 (분당동, 샛별마을삼부아파트)
(뒷면에 계속)
(74) 대리인
특허법인 무한

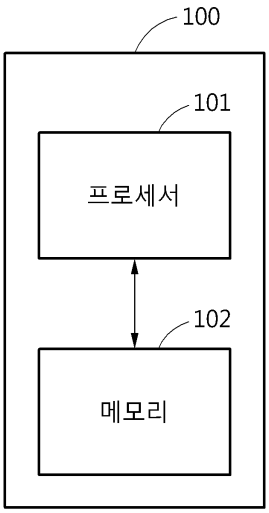
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **샘플링을 이용한 움직임 데이터 처리 방법**

(57) 요 약

샘플링을 이용한 움직임 데이터 처리 방법이 개시된다. 서브 샘플링 하고자 하는 프레임의 종류를 판단하고, 프레임의 종류에 따라 서브 샘플링의 비율을 다르게 결정하고 결정된 서브 샘플링의 비율을 이용하여 영상을 복구하는 구성이 개시된다. 또한, 복구하고자 하는 프레임의 종류를 판단하고, 프레임의 종류에 따라 움직임 데이터 복구의 비율을 다르게 결정하고 결정된 움직임 데이터 복구의 비율을 이용하여 움직임 데이터를 복구하는 구성이 개시된다.

대 표 도 - 도1



(52) CPC특허분류
H04N 19/59 (2015.01)
(72) 발명자
이광순
대전광역시 유성구 진잠로149번길 30, 208동 303호 (교촌동, 한승미메이드아파트)
허남호
세종특별자치시 도움1로 105 (종촌동)

허영수
경기도 수원시 영통구 봉영로1492번길 20, B01호 (영통동)

이 발명을 지원한 국가연구개발사업
과제고유번호 14-000-02-001
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터(IITP)
연구사업명 정보통신 방송연구개발사업
연구과제명 초고품질 콘텐츠 지원 UHD실감 방송/디지털 시네마/사이니지 융합서비스 기술 개발
기 여 율 1/1
주관기관 한국전자통신연구원
연구기간 2014.03.01 ~ 2015.02.28

명 세 서

청구범위

청구항 1

서브 샘플링하고자 하는 움직임 데이터를 포함하는 입력 프레임의 종류를 판단하는 단계; 및

상기 입력 프레임의 종류에 따라 상기 입력 프레임에 포함된 움직임 데이터에 대한 서브 샘플링의 비율을 다르게 결정하여 상기 입력 프레임의 움직임 데이터를 서브 샘플링하는 단계

를 포함하는 움직임 데이터 처리 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 움직임 정보를 저장하는 버퍼의 크기를 절약하기 위해 움직임 데이터를 샘플링(Subsampling)하기 위한 발명이다. 구체적으로, 비디오 인코딩을 위해 어떤 블록들에서 움직임 보상(Motion Compensation)을 위하여 움직임 예측(Motion Estimation)이 수행되는 경우, 움직임 예측의 결과로 출력되는 움직임 데이터를 포함하고 있는 각 복수개의 블록들에서 움직임 데이터들을 샘플링하여 이를 대표하는 값을 구하고 저장함으로써 요구되는 메모리의 크기를 줄이는 방법에 관한 발명이다.

배 경 기 술

[0002] 영상을 부호화하는 방법은 인접하는 영상 간의 유사성을 이용하여 한 영상으로부터 다른 영상을 예측하는 방법이 사용될 수 있다. 텍스쳐 영상과 깊이 영상은 복수의 카메라를 통하여 복수의 시점으로 획득될 수 있다.

[0003] 획득된 영상들은 독립적으로 부호화될 수도 있지만, 여러 개의 시점에서 획득한 영상은 인접한 시점 사이에 유사성이 존재하므로 이를 이용하면 더 높은 부호화 효율을 달성할 수 있다.

[0004] 이미 부호화된 영상이 있는 경우, 현재 부호화하고자 하는 블록을 다양한 방법을 통해 예측할 수 있다. 이러한 예측 방법들로는

[0005] 첫째 인접한 영상은 이미 부호화된 영상에 기초하여 시점 간 예측이 가능할 수 있다.

[0006] 둘째 같은 시점 내에서는 이미 부호화된 이전 프레임에 기초하여 프레임간 간 움직임 정보 예측이 가능할 수 있다.

[0007] 셋째 깊이 영상 부호화 시 이와 깊이 영상과 동일한 위치의 텍스쳐 영상의 움직임 정보를 상속 받아 사용하는 움직임 정보 상속 예측이 가능할 수 있다.

[0008] 이와 같은 예측을 통해 각 블록들이 부호화된 경우, 움직임 데이터가 생성될 수 있다. 이때, 각 블록의 움직임 데이터는 다른 영상에서 참조할 수 있기 때문에 버퍼에 저장되어야 한다. 따라서 큰 저장 공간이 소요될 수 있다.

[0009] 움직임 데이터를 저장하기 위해 필요한 메모리의 크기를 줄이기 위해 각 블록들은 대표하는 값을 구하는 서브 샘플링 방법이 사용될 수 있다. 그러나, 종래에는 프레임의 종류에 상관 없이 서브 샘플링의 비율을 동일하게 적용하여 압축 효율이 크지 않았다.

발명의 내용

해결하려는 과제

[0010] 본 발명의 일실시예로서 서브 샘플링하고자 하는 입력 프레임의 종류를 판단하고, 입력 프레임의 종류에 따라 움직임 데이터들의 서브 샘플링의 비율을 다르게 결정하고 결정된 움직임 데이터들의 서브 샘플링의 비율을 이용하여 입력 프레임을 움직임 데이터들을 서브 샘플링을 통한 대표값들을 구하고 이를 대표 움직임 데이터로서 저장하여 메모리의 크기를 감소시키는 방법을 제공한다.

[0011] 본 발명의 일실시예로서 움직임 데이터 복구를 수행하고자 하는 입력 프레임의 종류를 판단하고, 입력 프레임의 종류에 따라 움직임 데이터 복구의 비율을 다르게 결정하고 결정된 움직임 데이터 복구의 비율을 이용하여 입력 프레임의 움직임 데이터 복구를 수행함으로써 움직임 데이터를 저장하기 위해 필요한 메모리의 크기를 감소시키는 방법을 제공한다.

과제의 해결 수단

[0012] 본 발명의 일실시예에 따른 움직임 데이터 저장 방법은, 서브 샘플링하고자 하는 입력 프레임의 종류를 판단하는 단계, 상기 입력 프레임의 종류에 따라 상기 입력 프레임에 포함된 움직임 데이터에 대한 서브 샘플링의 비율을 다르게 결정하여 상기 입력 프레임을 서브 샘플링하는 단계를 포함할 수 있다.

[0013] 상기 입력 프레임의 종류를 판단하는 단계는 상기 입력 프레임이 텍스쳐 영상에 포함되는지 또는 깊이에 포함되는지 판단하고, 상기 서브 샘플링하는 단계는, 상기 입력 프레임이 텍스쳐 영상에 포함된 경우의 서브 샘플링의 비율은 상기 입력 프레임이 깊이 영상에 포함된 경우의 서브 샘플링의 비율보다 작을 수 있다.

[0014] 상기 입력 프레임의 종류를 판단하는 단계는 상기 입력 프레임과 참조 프레임의 시간적 간격을 판단하고, 상기 서브 샘플링하는 단계는 상기 시간적 간격에 기초하여 서브 샘플링의 비율을 결정할 수 있다.

[0015] 본 발명의 일실시예에 따른 움직임 데이터 처리 방법은, 깊이 영상을 구성하는 복수의 입력 프레임들 각각에 대해 인코딩하는 단계, 텍스쳐 영상을 구성하는 복수의 입력 프레임들 각각에 대해 인코딩하는 단계, 상기 인코딩을 통해 출력된 깊이 영상의 움직임 데이터에 대해 서브 샘플링하는 단계 및 상기 인코딩을 통해 출력된 텍스쳐 영상의 움직임 데이터에 대해 서브 샘플링하는 단계를 포함하고, 상기 깊이 영상의 움직임 데이터에 대한 서브 샘플링의 비율은 상기 텍스쳐 영상의 움직임 데이터에 대한 서브 샘플링의 비율보다 클 수 있다.

[0016] 상기 깊이 영상의 움직임 데이터에 대해 서브 샘플링하는 단계는, 상기 깊이 영상의 복수의 입력 프레임들에 대해 인코딩이 완료된 이후에, 상기 깊이 영상의 입력 프레임에 설정된 N*N크기의 엑세스 유닛에 대해 1/M에 대응하는 엑세스 유닛의 움직임 데이터를 서브 샘플링하는 단계를 포함할 수 있다.

[0017] 상기 깊이 영상의 움직임 데이터에 대해 서브 샘플링하는 단계는, 상기 N*N크기의 엑세스 유닛에 대한 인코딩이 완료된 이후에, 상기 깊이 영상의 입력 프레임에 설정된 2N*2N크기의 엑세스 유닛에 대해 1/(4*M)에 대응하는 엑세스 유닛의 움직임 데이터를 서브 샘플링하는 단계를 더 포함할 수 있다.

[0018] 본 발명의 다른 실시예에 따른 움직임 데이터 처리 방법은 움직임 데이터 복구를 수행하고자 하는 입력 프레임의 종류를 판단하는 단계 및 상기 입력 프레임의 종류에 따라 상기 입력 프레임에 포함된 움직임 데이터에 대한 움직임 데이터 복구의 비율을 다르게 결정하여 상기 입력 프레임의 움직임 데이터 복구를 수행하는 단계를 포함할 수 있다.

[0019] 상기 입력 프레임의 종류를 판단하는 단계는 상기 입력 프레임이 텍스쳐 영상에 포함되는지 또는 깊이 영상에 포함되는지 판단하고, 상기 움직임 데이터 복구를 수행하는 단계는, 상기 입력 프레임이 텍스쳐 영상에 포함된 경우의 움직임 데이터 복구의 비율은 상기 입력 프레임이 깊이 영상에 포함된 경우의 움직임 데이터 복구의 비율보다 작을 수 있다.

[0020] 상기 입력 프레임의 종류를 판단하는 단계는 상기 입력 프레임과 참조 프레임의 시간적 간격을 판단하고, 상기 움직임 데이터 복구를 수행하는 단계는 상기 시간적 간격에 기초하여 움직임 데이터 복구의 비율을 결정할 수 있다.

[0021] 본 발명의 일실시예에 따른 움직임 데이터 처리 방법은, 깊이 영상의 움직임 데이터에 대해 움직임 데이터 복구를 수행하는 단계 및 텍스쳐 영상의 움직임 데이터에 대해 움직임 데이터 복구를 수행하는 단계를 포함하고, 상기 깊이 영상의 움직임 데이터에 대한 움직임 데이터 복구의 비율은 상기 텍스쳐 영상의 움직임 데이터에 대한 움직임 데이터 복구의 비율보다 클 수 있다.

[0022] 상기 깊이 영상의 움직임 데이터에 대해 움직임 데이터 복구를 수행하는 단계는, 상기 깊이 영상의 입력 프레임에 설정된 2N*2N크기의 엑세스 유닛에 대해 4M에 대응하는 엑세스 유닛의 움직임 데이터의 움직임 데이터 복구를 수행하는 단계를 포함할 수 있다.

[0023] 상기 깊이 영상의 움직임 데이터에 대해 움직임 데이터 복구를 수행하는 단계는, 상기 2N*2N크기의 엑세스 유닛에 대한 디코딩이 완료된 이후에, 상기 깊이 영상의 입력 프레임에 설정된 N*N크기의 엑세스 유닛에 대해 M에 대응하는 엑세스 유닛의 움직임 데이터의 움직임 데이터 복구를 수행하는 단계를 더 포함할 수 있다.

발명의 효과

[0024] 본 발명의 일실시예로서 서브 샘플링하고자 하는 입력 프레임의 종류를 판단하고, 입력 프레임의 종류에 따라 서브 샘플링의 비율을 다르게 결정하고 결정된 서브 샘플링의 비율을 이용하여 입력 프레임을 서브 샘플링함으로써 움직임 데이터를 저장하기 위해 필요한 메모리의 크기를 감소시킬 수 있다.

[0025] 본 발명의 일실시예로서 움직임 데이터 복구를 수행하고자 하는 입력 프레임의 종류를 판단하고, 입력 프레임의 종류에 따라 움직임 데이터 복구의 비율을 다르게 결정하고 결정된 움직임 데이터 복구의 비율을 이용하여 입력 프레임의 움직임 데이터 복구를 수행함으로써 움직임 데이터를 저장하기 위해 필요한 메모리의 크기를 감소시킬 수 있다.

도면의 간단한 설명

[0026] 도 1은 본 발명의 일실시예에 따른 움직임 데이터 처리 장치를 나타낸 도면이다.

도 2는 본 발명의 일실시예에 따른 입력 프레임, 엑세스 유닛 및 블록의 관계와 블록의 움직임 데이터를 나타낸 도면이다.

도 3(a)는 본 발명의 일실시예에 따른 입력 프레임의 종류에 관계없이 도 2(a)의 각 블록들에 포함된 움직임 데이터(정보)를 동일한 비율로 도 2(b)와 같이 좌측상단의 블록만 샘플링하여 대표값으로 삼아 처리하는 방법을 나타낸 도면이고, 도 3(b)는 입력 프레임이 텍스처 영상이냐 깊이 영상에 따라 다른 비율로 샘플링하여 대표값을 구하는 처리 방법을 나타낸 도면이다.

도 4는 본 발명의 일실시예에 따른 입력 프레임의 종류에 따라 다른 비율로 점진적으로(progressive) 서브 샘플링되는 움직임 데이터 처리 방법을 나타낸 도면이다.

도 5는 본 발명의 일실시예에 따른 입력 프레임의 종류에 따라 다른 비율로 즉시(immediate) 서브 샘플링되는 움직임 데이터 처리 방법을 나타낸 도면이다.

도 6은 본 발명의 일실시예에 따른 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

도 7은 본 발명의 일실시예에 따른 깊이 영상에 속한 프레임인지에 따라 다른 서브 샘플링의 비율로 처리되는 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

도 8은 본 발명의 일실시예에 따른 시간적 간격 판단에 기초하여 다른 서브 샘플링의 비율로 처리되는 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

도 9은 본 발명의 일실시예에 따른 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

도 10은 본 발명의 일실시예에 따른 깊이 영상인지에 따라 다른 움직임 데이터 복구의 비율로 처리되는 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

도 11는 본 발명의 일실시예에 따른 시간적 간격 판단에 기초하여 다른 움직임 데이터 복구의 비율로 처리되는 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0027] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0028] 도 1은 본 발명의 일실시예에 따른 움직임 데이터 처리 장치(100)를 나타낸 도면이다. 움직임 데이터 처리 장치(100)는 프로세서(101) 및 메모리(102)를 포함할 수 있다.

[0029] 영상은 복수의 입력 프레임으로 구성될 수 있다. 프로세서(101)는 입력 프레임을 블록 단위로 처리할 수 있다. 영상의 부호화 방법에서 프로세서(101)는 각 블록을 코딩하여 움직임 데이터를 추출하고, 이를 각 블록에 대응하여 메모리(102)에 저장할 수 있다. 각 블록에 대응하여 저장된 움직임 데이터는 입력 프레임 간 예측에 사용될 수 있다. 프로세서(101)는 입력 프레임 간 예측을 수행하기 전에 움직임 데이터를 서브 샘플링하여 움직임 데이터를 위해 요구되는 메모리의 크기를 줄일 수 있다. 프레임은 픽쳐(picture)라고 지칭될 수 있다.

[0030] 프로세서(101)는 화소 값을 직접 전송하는 대신 입력 프레임 간 예측을 통해 움직임 데이터와 레지듀얼(residual)을 부호화 하여 압축 효율을 증가시킬 수 있다. 움직임 데이터를 부호화하는 데 요구되는 데이터의

양을 축소하기 위하여, 프로세서(101)는 참조 프레임의 움직임 데이터를 참조하여 블록의 움직임 데이터와 레지듀얼을 나타낼 수 있다. 이를 통해 프로세서(101)는 압축 효율을 더욱 향상시킬 수 있다.

[0031] 부호화가 완료된 참조 프레임의 움직임 데이터들은 이후 부호화되는 입력 프레임에서 참조될 수 있다. 즉, 프로세서(101)는 예측(prediction)을 통해 압축 효율을 높일 수 있다. 예측 방법으로 시점간 예측(inter-view motion prediction)과 시점 내 예측(temporal motion prediction)이 사용될 수 있다. 또한, 깊이(depth) 영상의 경우 깊이 영상과 쌍을 이루는 텍스쳐(texture) 영상에 속하는 동일 위치의 블록의 움직임 데이터를 참조하는 움직임 파라미터 상속(motion parameter inheritance)이 사용될 수 있다.

[0032] 시점간 예측은 인접한 시점 간의 움직임 데이터가 유사하다는 성질을 이용한 방법이다. 시점간 예측에서 현재 부호화할 블록과 동일한 위치의 인접한 시점의 블록의 움직임 데이터가 참조된다. 즉, 시점간 예측은 동일한 시각, 동일한 위치에서 인접한 시점의 움직임 데이터를 통해 부호화하는 방법이다.

[0033] 시점 내 예측은 인접한 시간의 움직임 데이터는 유사하다는 성질을 이용한 방법이다. 시점 내 예측에서 현재 부호화할 블록과 동일한 위치의 인접한 시간의 블록의 움직임 데이터가 참조된다. 즉, 시점 내 예측은 동일한 시점, 동일한 위치에서 인접한 시간의 움직임 데이터를 통해 부호화하는 방법이다.

[0034] 블록에 저장된 움직임 데이터는 예측에 사용될 수 있기 때문에 모두 저장될 필요가 있다. 따라서 움직임 데이터 저장을 위해 큰 크기의 메모리가 요구될 수 있다. 요구되는 메모리의 크기를 축소하기 위해 다운 샘플링 기법이 사용될 수 있다. 다운 샘플링 기법은 하나의 블록의 움직임 데이터를 그 블록이 속한 일정한 영역 내의 복수의 블록의 대표값으로 사용하는 방법을 의미한다.

[0035] 다운 샘플링의 비율이 높을수록 움직임 데이터의 압축 비율이 증가하여 요구되는 메모리의 크기는 감소한다. 하지만, 다운 샘플링의 비율이 높을수록 하나의 움직임 데이터가 더 큰 영역에 속한 블록의 움직임 데이터를 대표하기 때문에, 이를 복구하는 경우 움직임 데이터의 예측 효율이 감소할 수 있다. 결과적으로 화질대 비트율(Peak Signal to Noise Ratio, PSNR)이 감소할 수 있다.

[0036] 화질대 비트율이 높은 종류의 입력 프레임은 화질대 비트율이 낮은 종류의 입력 프레임보다 높은 다운 샘플링의 비율이 사용될 수 있다. 따라서 입력 프레임의 종류에 따라 다른 다운 샘플링의 비율을 사용하여 다운 샘플링이 수행되는 경우, 더 높은 압축 효율이 달성될 수 있다.

[0037] 영상의 종류에는 텍스쳐 영상과 깊이 영상이 있다. 프로세서(101)가 깊이 영상에 속한 입력 프레임이 포함하는 움직임 데이터를 일정 비율로 다운 샘플링하는 경우 텍스쳐 영상에 속한 입력 프레임에 대해 다운 샘플링하는 경우 보다 화질대 비트율이 더 높을 수 있다. 따라서 프로세서(101)가 깊이 영상과 텍스쳐 영상의 다운 샘플링의 비율을 달리 처리한다면 전체적으로 더 높은 수준의 압축률이 달성될 수 있다. 깊이 영상은 깊이맵(depth map)으로 지칭될 수도 있다.

[0038] 다운 샘플링은 2단계의 서브 샘플링으로 구성될 수 있다. 서브 샘플링은 하나의 블록의 움직임 데이터를 그 블록이 속한 일정한 영역 내의 복수의 블록의 대표값으로 사용하는 방법을 의미한다.

[0039] 각 단계의 서브 샘플링에서 프로세서(101)는 샘플링되는 영역의 크기를 달리 함으로써 더 높은 압축률을 달성할 수 있다.

[0040] 입력 프레임의 종류에 따라 다른 비율로 다운 샘플링된 경우, 프로세서(101)는 다운 샘플링에 대응하는 비율로 입력 프레임의 종류에 따라 움직임 데이터 복구를 수행할 수 있다. 프로세서(101)는 대표 블록의 움직임 정보에 기초하여 일정 영역에 속한 블록의 움직임 정보들을 복구할 수 있다. 여기서 일정 영역은 엑세스 유닛(access unit, AU)에 대응할 수 있다.

[0041] 움직임 데이터 복구는 서브 샘플링을 통하여 특정 블록이 대표 블록으로 설정되는 경우, 움직임 데이터 복구 비율에 대응하는 복수의 블록의 움직임 데이터를 대표 블록이 포함하는 움직임 데이터로 복구하는 것을 의미한다. 일실시예로 움직임 데이터 복구는 복구하고자 하는 복수의 블록의 움직임 데이터를 대표 블록의 움직임 데이터와 동일하게 설정함으로써 수행될 수 있다.

[0042] 도 2는 본 발명의 일실시예에 따른 입력 프레임, 엑세스 유닛(access unit, AU), 및 블록의 관계와 블록의 움직임 데이터를 나타낸 도면이다.

[0043] 입력 영상은 복수의 입력 프레임을 포함할 수 있다. 입력 프레임은 복수의 블록으로 구성될 수 있다. 프로세서(101)는 다운 샘플링의 처리 단위로서 적어도 하나의 블록을 포함하는 엑세스 유닛(210)을 설정할 수 있다.

여기서 각각의 블록(230)은 움직임 데이터를 포함할 수 있다. 블록들이 포함하고 있는 움직임 데이터는 다른 블록의 예측에 사용될 수 있으므로 메모리(102)에 저장될 필요가 있다. 예를 들어, 도 2 (a)와 같이, 4*4 크기의 블록을 최소 블록의 크기로 사용하는 경우, 1820*1088 FULL HD 영상에서 130,560개의 움직임 데이터가 저장될 수 있다.

[0044] 각 시점에 대한 텍스쳐 영상과 깊이 영상의 각 블록의 움직임 데이터들이 모두 메모리(102)에 저장되어야 하므로, 영상 부호화 또는 복호화에 필요한 메모리 량은 크게 증가될 수 있다. 예를 들어, 하나의 시점에 대한 텍스쳐 영상의 움직임 데이터의 수가 130,560개인 경우, N개의 시점이 존재한다면 텍스쳐 영상과 깊이 영상 각각에 대한 움직임 데이터의 수는 2N*130,560가 될 수 있다.

[0045] 일실시예로서, 프로세서(101)가 움직임 데이터를 저장하는데 필요한 메모리 량을 축소하기 위하여 도 2 (b)와 같이 서브 샘플링이 사용될 수 있다. 프로세서(101)는 입력 프레임(211)에 대해 일정한 크기의 엑세스 유닛(220)을 설정한 후, 엑세스 유닛(220)의 움직임 데이터들을 특정 블록의 움직임 데이터로 서브 샘플링할 수 있다.

[0046] 예를 들어, 프로세서(101)는 입력 프레임(211)에 대해 엑세스 유닛0, 엑세스 유닛1, 엑세스 유닛2 및 엑세스 유닛 3을 설정할 수 있다. 프로세서(101)는 다운 샘플링을 통해 엑세스 유닛이 포함하는 복수의 블록 중 하나 이상을 대표 블록으로 설정할 수 있다. 각 블록은 움직임 데이터를 포함하며, 프로세서(101)는 복수의 블록의 움직임 데이터를 대표 블록의 움직임 데이터로 다운 샘플링할 수 있다. 대표 블록은 엑세스 유닛에 포함되는 복수의 블록 중 좌상단의 블록으로 설정될 수 있다. 하지만, 대표 블록은 이에 한정되지 않으며, 엑세스 유닛에 포함된다면 어느 위치에 있는 블록이든 대표 블록으로 설정될 수 있다.

[0047] 도 2 (b)에서 프로세서(101)는 4개의 블록을 포함하는 엑세스 유닛 0(220)을 설정할 수 있다. 프로세서(101)는 엑세스 유닛 0에 포함되는 4개의 블록 중 블록 0을 대표 블록으로 설정할 수 있다. 이를 통해 움직임 데이터를 저장하는데 소요되는 메모리의 수치는 1/4로 축소될 수 있다. 엑세스 유닛0을 참조하는 다른 엑세스 유닛들이 블록 1~3의 움직임 데이터를 참조하는 경우, 블록0의 움직임 데이터가 참조될 수 있다.

[0048] 엑세스 유닛의 크기를 N개의 블록을 포함하도록 설정하는 경우, 하나 이상의 블록이 N개의 블록을 대표할 수 있다. 일실시예로, N개의 블록을 하나의 블록이 대표하는 경우, 움직임 데이터를 저장하는데 소요되는 메모리의 수치는 1/N로 감소될 수 있다. 하지만, N의 값이 증가할수록 움직임 데이터 예측의 정확도가 감소할 수 있다. 이로 인해, 화질 대 비트율(Peak Signal to Noise Ratio, PSNR)이 감소될 수 있다. 따라서 움직임 데이터의 다운 샘플링의 비율은 압축 효율과 화질 대 비트율을 고려하여 결정되어야 한다. 다시 말해서, 움직임 데이터의 예측의 정확도가 임계값보다 낮아지는 것을 막기 위하여 움직임 데이터의 서브 샘플링의 비율은 임계값 이하로 제한 될 수 있다.

[0049] 도 3(a)는 본 발명의 일실시예에 따른 입력 프레임의 종류에 관계없이 도 2(a)의 각 블록들에 포함된 움직임 데이터(정보)를 동일한 비율로 도 2(b)와 같이 좌측상단의 블록만 샘플링하여 대표값으로 삼아 처리하는 방법을 나타낸 도면이고, 도 3(b)는 입력 프레임이 텍스쳐 영상이냐 깊이 영상에 따라 다른 비율로 샘플링하여 대표값을 구하는 처리 방법을 나타낸 도면이다.

[0050] 도 3 (a), (b)에서 사각형들(301, 302, 303 및 304)은 움직임 데이터를 저장하기 위해 소요되는 메모리의 크기를 나타낸다. 도 3 (a)에서 프로세서(101)는 입력 프레임의 종류에 관계 없이 동일한 샘플링의 비율로 움직임 정보를 서브 샘플링할 수 있다. 따라서, 서브 샘플링된 이후 입력 프레임의 종류에 관계 없이 같은 크기의 축소 비율을 가질 수 있다.

[0051] 깊이 영상에 속한 입력 프레임의 경우 텍스쳐 영상에 속한 입력 프레임에 비해 더 높은 샘플링의 비율로 샘플링되는 경우에도 임계값 이상의 화질 대 비트율이 유지될 수 있다. 따라서, 도 3 (b)에서 프로세서(101)는 입력 프레임의 종류에 따라 서브 샘플링의 비율을 다르게 결정할 수 있다. 구체적으로 깊이 영상에 포함된 프레임에 대응하는 서브 샘플링 비율은 텍스쳐 영상에 포함된 프레임에 대응하는 서브 샘플링 비율보다 클 수 있다. 결과적으로 서브 샘플링된 이후에, 깊이 영상에 포함된 프레임에 대응하는 메모리의 크기는 텍스쳐 영상에 포함된 프레임에 대응하는 메모리의 크기보다 작을 수 있다.

[0052] 도 4는 본 발명의 일실시예에 따른 입력 프레임의 종류에 따라 다른 비율로 점진적으로 서브 샘플링되는 움직임 데이터 처리 방법을 나타낸 도면이다.

[0053] 다운 샘플링은 2단계의 서브 샘플링으로 구성될 수 있다. 프로세서(101)는 적어도 하나 이상의 블록을 포함하는 엑세스 유닛을 처리 단위로 설정할 수 있다. 다운 샘플링은 2단계의 서브 샘플링으로 구성될 수 있다. 1단

계 서브 샘플링(420)에서 프로세서(101)는 엑세스 유닛에 포함되는 움직임 데이터들을 서브 샘플링할 수 있다. 그 결과 엑세스 유닛의 움직임 데이터들은 하나의 움직임 데이터로 대표될 수 있다. 1단계 서브 샘플링이 완료된 후, 2단계 서브 샘플링(410)에서 프로세서(101)는 입력 프레임이 포함하는 엑세스 유닛들의 서브 샘플링된 움직임 데이터를 다시 서브 샘플링할 수 있다.

[0054] 깊이 영상의 코딩의 경우, 움직임 파라미터 상속의 정확도는 텍스쳐 영상보다 높을 수 있다. 따라서, 깊이 영상을 코딩하는 경우 움직임 데이터에 대해 더 높은 다운 샘플링의 비율을 사용하더라도 텍스쳐 영상에 비해 높은 수준의 정확도가 유지될 수 있다.

[0055] 깊이 영상은 복수의 입력 프레임들을 포함할 수 있다. 복수의 입력 프레임들에 대해 인코딩이 수행되어 각 블록에 대한 움직임 데이터가 도출될 수 있다. 1단계 서브 샘플링(420)에서 깊이 영상의 입력 프레임에 설정된 N*N크기의 엑세스 유닛에 대해 프로세서(101)은 X 에 대응하는 움직임 데이터를 서브 샘플링할 수 있다. 예를 들어, X는 1/M일 수 있다.

[0056] N*N크기의 엑세스 유닛에 대한 인코딩이 완료된 이후에, 프로세서(101)는 2단계 서브 샘플링(410)에서 깊이 영상의 입력 프레임에 설정된 2N*2N크기의 엑세스 유닛에 대해 Y에 대응하는 움직임 데이터를 서브 샘플링할 수 있다. 예를 들어, Y는 1/(4*M)일 수 있다.

[0057] 【표 1】

	각각의 깊이 영상에 대한 코딩이 완료된 후(After each depth is coded)	엑세스 유닛에 대한 코딩이 완료된 후(After AU is coded)
앵커 영상 (Anchor)	1/4 (8x8 unit)	1/16 (16x16 unit)
케이스 1	1/64 (32x32)	1/64 (32x32)
케이스 2	1/64 (32x32)	1/256 (64x64, LCU)
케이스 3	1/256 (64x64, LCU)	1/256 (64x64, LCU)
케이스 4	1/256 (64x64, LCU)	1/1024 (128x128, 4LCU)
케이스 5	1/1024 (128x128, 4LCU)	1/1024 (128x128, 4LCU)

[0058] 현재 버전의 3D-HEVC는 움직임 데이터(motion data)를 저장하기 위해 사용되는 버퍼의 크기를 줄이기 위하여, 움직임 데이터에 대해 서브 샘플링을 수행한다. 각 픽쳐가 코딩된 이후에, 8*8 유닛(unit)의 경우에, 좌상단 4*4 블록은 움직임 데이터의 대표로서 사용되고, 움직임 데이터 버퍼의 1/4 감소를 초래한다. 엑세스 유닛(AU)가 코딩된 이후에, 16*16 유닛의 경우에, 좌상단 4*4 블록은 또한 움직임 데이터의 대표로서 사용되고, 움직임 데이터 버퍼의 1/16 감소를 초래한다.

[0059] 현재 버전의 3D-HEVC에서, 움직임 데이터 압축은 텍스쳐 영상과 깊이 영상 양자에 대해 동일한 비율로 수행된다. 본 발명은 깊이 영상에 대해 더 높은 압축 비율을 제안한다. 코딩된 프레임으로부터 도출된 움직임 데이터는 AMVP 또는 머지 모드(merge mode)를 위해 사용될 수 있다. 텍스쳐 영상의 코딩의 경우에, 시점간 움직임(inter-view motion)을 사용하는 움직임 예측이 자주 수행된다. 저장된 움직임 데이터의 사용이 필요하기 때문에, 너무 크게 서브 샘플링 비율을 증가시키는 것은 시점간 움직임 파라미터 예측(inter-view motion parameter prediction)의 효율성을 떨어뜨리는 것을 초래할 것이고, 코딩 효율성의 감소를 초래할 것이다.

[0060] 그러나, 깊이 영상의 경우에, 오직 깊이 영상에 대응하는 텍스쳐 영상의 움직임 데이터를 사용하는 움직임 파라미터 상속(motion parameter inheritance)의 정확성은 매우 높다. 이것은, 텍스쳐 코딩과 달리 더 높은 압축 비율을 깊이 움직임 데이터(depth motion data)에 사용하는 것이 코딩 효율성에 문제를 야기하지 않을 것임을 의미한다.

[0061] 표 1은 깊이 움직임 데이터에 적용될 수 있는 서브 샘플링 비율을 제안한다. 각 유닛에 대해, 좌상단 서브-블록(sub-block)은 현재의 3D-HEVC와 동일한 방식으로 움직임 데이터의 대표로서 사용된다. 케이스 1, 3 및 5는 각 깊이 영상이 코딩된 이후에 즉시 서브 샘플링을 수행한다. 반면, 케이스 2 및 4는 점진적인 서브 샘플링을 수행한다. 비율 1/256은 LCU 수준과 동일하다.

[0062] 도 4는 깊이 영상에 대해 점진적인 움직임 데이터 압축에 대해 선택적인 서브 샘플링의 비율(selective subsample ratio)을 사용하는 표 1의 케이스 2를 나타낸다.

[0063] 도 4와 도 5 및 표 1은 블록을 4*4로 설정한 일실시예다. 사각형의 크기는 움직임 데이터를 저장하는데 필요한 메모리의 크기를 나타낼 수 있다. 다운 샘플링은 2단계 서브 샘플링으로 구성될 수 있다. 표 1의 케이스 1, 3

및 5는 도 5에 대응하며, 1단계 서브 샘플링 만으로 다운 샘플링 비율이 달성되는 예시들이다. 표 1의 케이스 2 및 4는 도 4에 대응하고, 2단계 서브 샘플링을 통해 다운 샘플링 비율이 달성되는 예시들이다.

[0064] 표 1에서 프로세서(101)는 앵커 영상에서 8*8 크기의 엑세스 유닛에 대해 1/4 에 대응하는 움직임 데이터를 1단계 서브 샘플링할 수 있다. 그 결과 필요한 메모리의 크기는 1/4로 축소될 수 있다. 프로세서(101)는 앵커 영상에서 16*16 크기의 엑세스 유닛에 대해 1/16에 대응하는 움직임 데이터를 2단계 서브 샘플링할 수 있다.

[0065] 케이스 2에서 프로세서(101)는 32*32 크기의 엑세스 유닛에 대해 1/64에 대응하는 움직임 데이터를 1단계 서브 샘플링할 수 있다. 그 결과 필요한 메모리의 크기는 1/64로 축소될 수 있다. 프로세서(101)는 64*64 크기의 엑세스 유닛에 대해 1/256에 대응하는 움직임 데이터를 2단계 서브 샘플링할 수 있다.

[0066] 케이스 4에서 프로세서(101)는 64*64 크기의 엑세스 유닛에 대해 1/256에 대응하는 움직임 데이터를 1단계 서브 샘플링할 수 있다. 그 결과 필요한 메모리의 크기는 1/256로 축소될 수 있다. 프로세서(101)는 128*128 크기의 엑세스 유닛에 대해 1/1024에 대응하는 움직임 데이터를 2단계 서브 샘플링할 수 있다.

[0067] 도 5는 본 발명의 일실시예에 따른 입력 프레임의 종류에 따라 다른 비율로 즉시 서브 샘플링되는 움직임 데이터 처리 방법을 나타낸 도면이다.

[0068] 1단계 서브 샘플링(520)에서 프로세서(101)는 엑세스 유닛에 포함되는 움직임 데이터들을 서브 샘플링할 수 있다. 그 결과 엑세스 유닛의 움직임 데이터들은 하나의 움직임 데이터로 대표될 수 있다. 1단계 서브 샘플링이 완료된 후, 2단계 서브 샘플링(510)에서 프로세서(101)는 입력 프레임이 포함하는 엑세스 유닛들의 서브 샘플링된 움직임 데이터를 다시 서브 샘플링할 수 있다. 하지만, 2단계 서브 샘플링(510)은 서브 샘플링의 비율이 1/1이므로 다운 샘플링은 사실상 1단계 서브 샘플링(520)에서 마무리된다.

[0069] 예를 들어, 표 1의 케이스 1에서 프로세서(101)는 32*32 크기의 엑세스 유닛에 대해 1/64에 대응하는 움직임 데이터를 서브 샘플링할 수 있다. 케이스 3에서 프로세서(101)는 64*64 크기의 엑세스 유닛에 대해 1/256에 대응하는 움직임 데이터를 서브 샘플링할 수 있다. 케이스 5에서 프로세서(101)는 128*128 크기의 엑세스 유닛에 대해 1/1024에 대응하는 움직임 데이터를 서브 샘플링할 수 있다.

[0070] 결과적으로, 도 5 및 표 1의 케이스 1, 3 및 5는 1단계 서브 샘플링에서 다운 샘플링의 비율이 달성되는 점에서 도 4 및 표 1의 케이스 2 및 4와 차이가 있다.

[0071] 도 6은 본 발명의 일실시예에 따른 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

[0072] 단계(610)에서 프로세서(101)는 서브 샘플링하고자 하는 입력 프레임의 종류를 판단할 수 있다. 입력 프레임의 종류에 따라 다르게 처리하는 이유는 입력 프레임의 종류에 따라 동일한 샘플링의 비율에 대해서 화질 대 비트율이 다를 수 있기 때문이다. 다시 말해서, 비교적 높은 화질 대 비트율을 가지는 종류의 영상은 더 높은 샘플링의 비율을 사용해도 임계값 이상의 화질 대 비트율을 유지할 수 있으므로, 더 높은 압축률을 달성할 수 있다. 따라서, 입력 프레임의 종류에 따라 서브 샘플링의 비율을 다르게 설정할 경우 움직임 데이터를 저장하는데 소요되는 메모리의 크기를 줄일 수 있다.

[0073] 단계(620)에서 프로세서(101)는 입력 프레임의 종류에 따라 서브 샘플링의 비율을 다르게 결정하고 결정된 샘플링의 비율에 따라 단계(630)에서 영상을 서브 샘플링할 수 있다. 일실시예로 입력 프레임의 종류는 텍스쳐 영상에 속한 입력 프레임과 깊이 영상에 속한 입력 프레임으로 구분될 수 있다. 다른 실시예로 입력 프레임의 종류는 참조 프레임과 입력 프레임의 시간적 간격에 기초하여 결정될 수 있다.

[0074] 도 6에 따라 움직임 데이터에 대해 서브 샘플링이 수행된 경우, 프로세서(101)는 움직임 예측을 수행할 수 있다. 프로세서(101)는 움직임 예측을 위해 참조될 움직임 데이터를 지닌 참조 블록의 좌표를 계산할 수 있다. 프로세서(101)는 계산된 참조 블록의 좌표를 기초로 예측하려는 블록의 움직임 데이터를 예측할 수 있다.

[0075] 예를 들어, 4*4 블록 4개로 이루어진 블록 집합에 대해 좌 상단의 블록의 움직임 데이터가 대표값으로 사용되는 경우, 복구 하려는 블록의 x좌표 및 y 좌표를 8로 나눈 후 다시 8을 곱해주면 대표값이 위치한 좌표가 계산될 수 있다. 입력 프레임의 종류에 따라 움직임 데이터 복구의 비율이 달라지기 때문에, 참조 블록의 좌표를 계산하는 과정 또한 입력 프레임의 종류에 따라 달라질 수 있다. 따라서 움직임 예측이 수행되는 프레임이 깊이 영상에 속한 경우, 더 큰 범위에 대해 참조 블록의 좌표를 설정할 수 있다.

[0076] 도 7은 본 발명의 일실시예에 따른 깊이 영상인지에 따라 다른 서브 샘플링의 비율로 처리되는 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

[0077] 단계(710)에서 프로세서(101)는 입력 프레임이 텍스쳐 영상에 포함된 것인지 깊이 영상에 포함된 것인지 판단할 수 있다. 프로세서(101)는 입력 프레임의 종류를 판단한 후, 텍스쳐 영상인 경우 단계(730)에서 제1 샘플링의 비율을 선택하고 입력 프레임의 종류가 깊이 영상인 경우 단계(720)에서 제2 샘플링의 비율을 결정할 수 있다. 단계(730)에서 프로세서(101)는 결정된 샘플링의 비율로 영상을 서브 샘플링 샘플링할 수 있다. 여기서, 제2 샘플링의 비율은 제1 샘플링의 비율보다 크다.

[0078] 도 8은 본 발명의 일실시예에 따른 시간적 간격 판단에 기초하여 다른 서브 샘플링의 비율로 처리되는 서브 샘플링을 이용한 움직임 데이터 처리 방법을 나타낸 도면이다.

[0079] 프로세서(101)는 입력 프레임의 종류를 판단하는 경우, 단계(810)에서 서브 샘플링 하려는 영상의 입력 프레임과 참조 프레임의 시간적 간격을 판단할 수 있다. 즉, 프로세서(101)는 서브 샘플링 하려는 입력 프레임과 참조 프레임의 시간적 간격이 특정 수치라고 판단할 수 있다.

[0080] 단계(820)에서 프로세서(101)는 시간적 간격에 기초하여 서브 샘플링의 비율을 결정할 수 있다. 샘플링 하려는 입력 프레임과 참조 프레임 간의 시간적 간격이 높을수록 참조 프레임의 움직임 데이터를 활용한 예측 부호화 효율이 감소하기 때문에, 움직임 데이터의 서브 샘플링의 비율을 높이더라도 신호대 잡음비의 영향이 작을 수 있다. 따라서, 프로세서(101)는 시간적 간격이 클수록 더 높은 서브 샘플링의 비율을 결정할 수 있다.

[0081] 도 9는 본 발명의 일실시예에 따른 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

[0082] 단계(910)에서 프로세서(101)는 움직임 데이터 복구를 수행하고자 하는 입력 프레임의 종류를 판단할 수 있다. 단계(920)에서 프로세서(101)는 입력 프레임의 종류에 따라 움직임 데이터 복구의 비율을 다르게 결정할 수 있다. 단계(930)에서 프로세서(101)는 결정된 움직임 데이터 복구의 비율에 따라 움직임 데이터의 움직임 데이터 복구를 수행할 수 있다.

[0083] 일실시예로, 단계(930)에서 프로세서(101)는 깊이 영상에 속한 입력 프레임의 움직임 데이터에 대해 움직임 데이터 복구를 수행할 수 있다. 프로세서(101)는 텍스쳐 영상에 속한 입력 프레임의 움직임 데이터에 대해 움직임 데이터 복구를 수행할 수 있다. 이 경우, 깊이 영상에 속한 입력 프레임의 움직임 데이터에 대한 움직임 데이터 복구의 비율은 텍스쳐 영상에 속한 입력 프레임의 움직임 데이터에 대한 움직임 데이터 복구의 비율보다 클 수 있다.

[0084] 도 10은 본 발명의 일실시예에 따른 깊이 영상인지에 따라 다른 움직임 데이터 복구의 비율로 처리되는 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

[0085] 단계(1010)에서 프로세서(101)는 입력 영상이 텍스쳐 영상인지 깊이 영상인지 판단할 수 있다. 프로세서(101)는 입력 프레임의 종류가 텍스쳐 영상에 속한 프레임인 경우 단계(1020)에서 제1 움직임 데이터 복구의 비율을 선택하고, 입력 프레임의 종류가 깊이 영상에 속한 프레임인 경우 단계(1030)에서 제2 움직임 데이터 복구의 비율을 선택할 수 있다. 여기서 제2 움직임 데이터 복구의 비율은 제1 움직임 데이터 복구의 비율보다 크다.

[0086] 일실시예로 움직임 데이터 복구는 2단계로 구성될 수 있다. 1단계 움직임 데이터 복구에서 깊이 영상의 입력 프레임에 설정된 2N*2N크기의 엑세스 유닛에 대해 프로세서(101)는 Z에 대응하는 움직임 데이터의 움직임 데이터 복구를 수행할 수 있다. 예를 들어, Z는 4M일 수 있다.

[0087] 2N*2N크기의 엑세스 유닛에 대한 디코딩이 완료된 이후에, 프로세서(101)는 2단계 움직임 데이터 복구에서 깊이 영상의 입력 프레임에 설정된 N*N크기의 엑세스 유닛에 대해 W에 대응하는 움직임 데이터의 움직임 데이터 복구를 수행할 수 있다. 예를 들어, W는 M일 수 있다.

[0088] 도 11은 본 발명의 일실시예에 따른 시간적 간격 판단에 기초하여 다른 움직임 데이터 복구의 비율로 처리되는 움직임 데이터 복구를 이용하여 움직임 데이터를 복구하는 움직임 데이터 처리 방법을 나타낸 도면이다.

[0089] 단계(1110)에서 프로세서(101)는 입력 프레임과 참조 프레임의 시간적 간격을 판단할 수 있다. 단계(1120)에서 프로세서(101)는 시간적 간격에 기초하여 움직임 데이터 복구의 비율을 결정할 수 있다. 입력 프레임과 참조 프레임 간의 시간적 간격이 높을수록 참조 프레임의 움직임 데이터를 활용한 예측 부호화 효율이 감소하기 때문에, 움직임 데이터의 샘플링의 비율을 높이더라도 신호대 잡음비의 영향이 작을 수 있다. 따라서, 프로세서(101)는 시간적 간격이 클수록 더 높은 움직임 데이터 복구의 비율을 결정할 수 있다. 단계(930)에서 프로세서(101)는 결정된 움직임 데이터 복구의 비율에 따라 움직임 데이터의 움직임 데이터 복구를 수행할 수 있다.

[0090] 본 발명의 실시 예에 따른 방법들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0091] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0092] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0093] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어부로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0094] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

부호의 설명

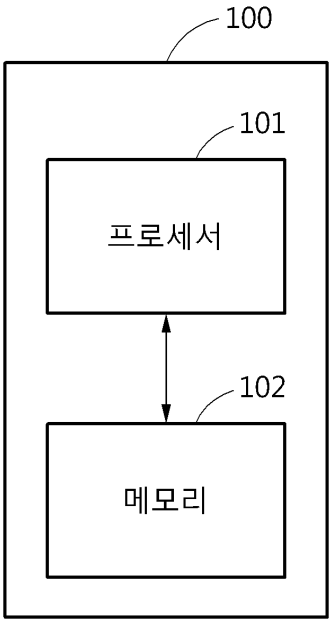
[0095] 100: 움직임 데이터 처리 장치

101: 프로세서

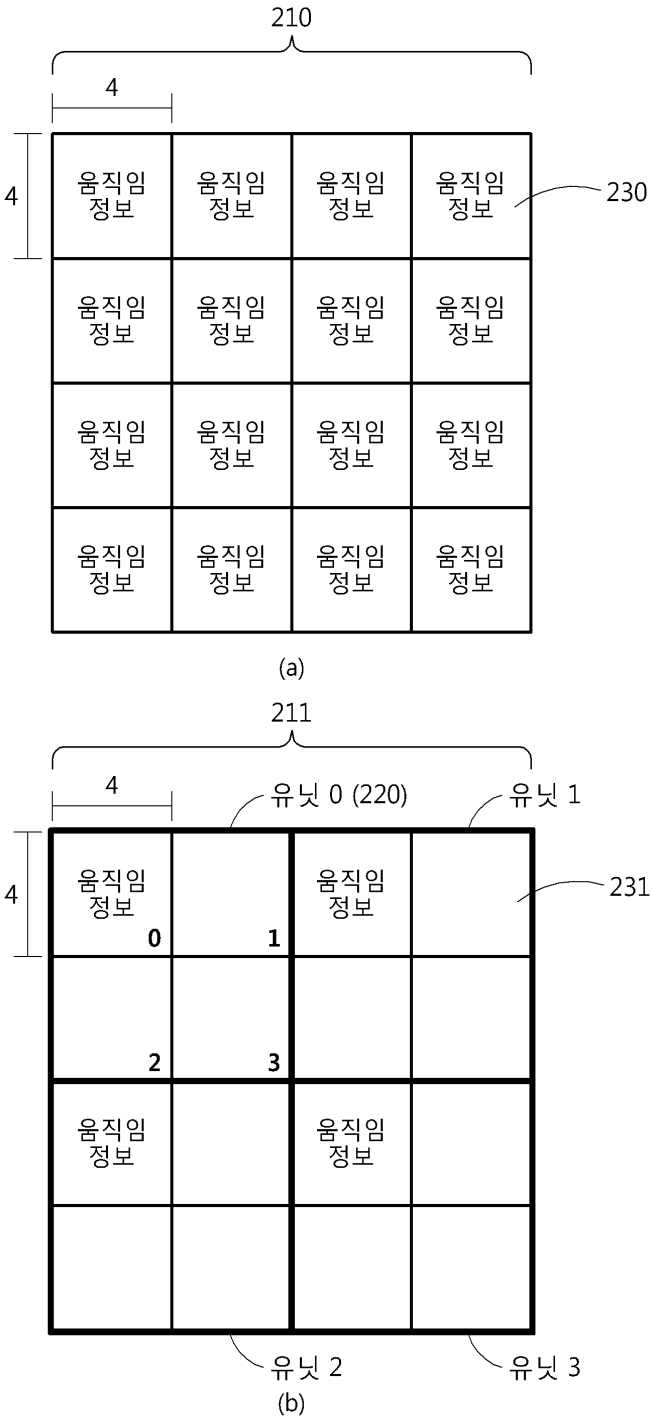
102: 메모리

도면

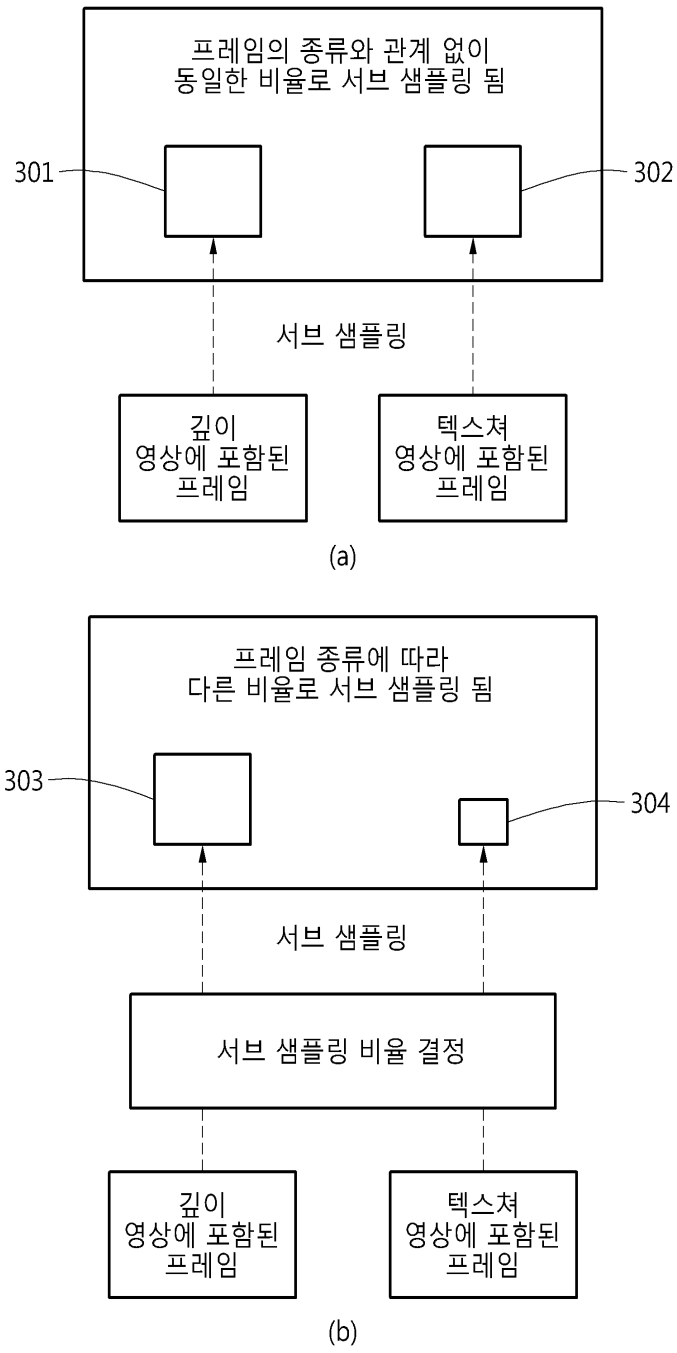
도면1



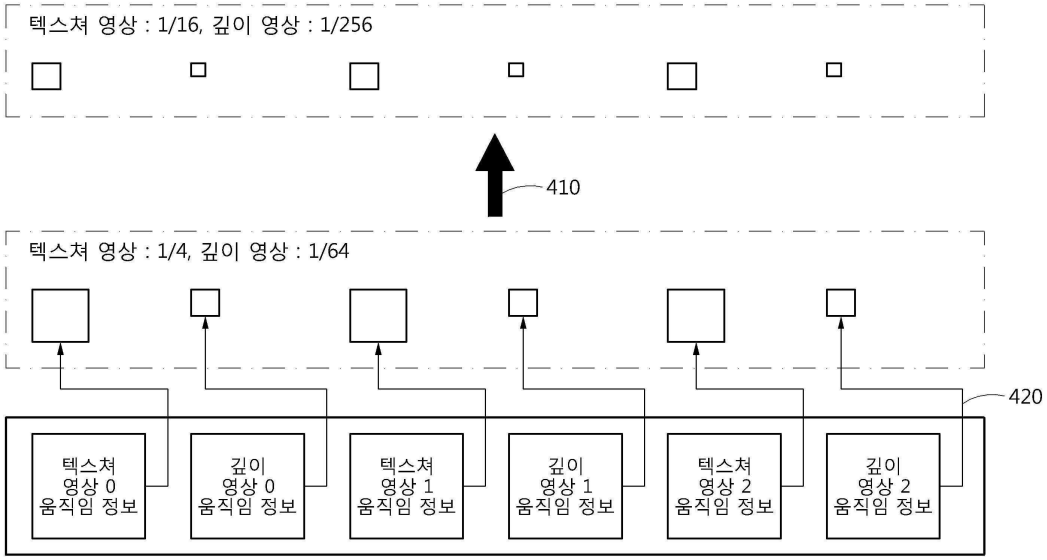
도면2



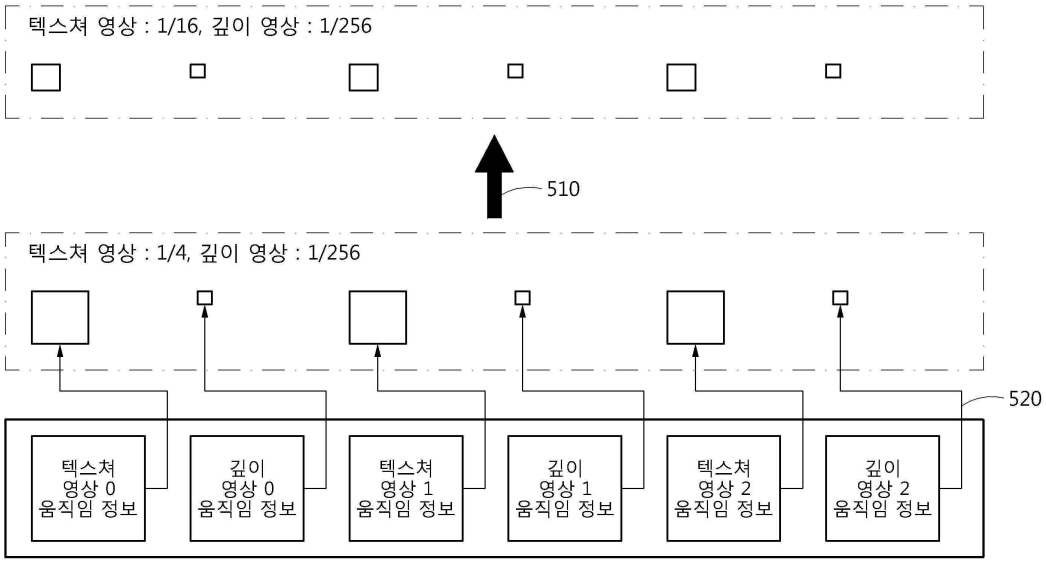
도면3


프레임의 종류와 관계 없이
동일한 비율로 서브 샘플링 됨
301
302
서브 샘플링
깊이
영상에 포함된
프레임
텍스쳐
영상에 포함된
프레임
(a)
프레임 종류에 따라
다른 비율로 서브 샘플링 됨
303
304
서브 샘플링
서브 샘플링 비율 결정
깊이
영상에 포함된
프레임
텍스쳐
영상에 포함된
프레임
(b)

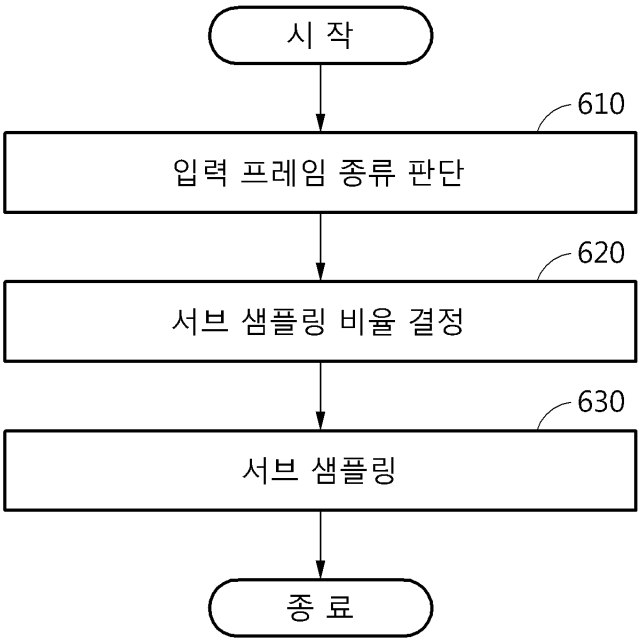
도면4


텍스쳐 영상 : 1/16, 깊이 영상 : 1/256
410
텍스쳐 영상 : 1/4, 깊이 영상 : 1/64
420
텍스쳐 영상 0 움직임 정보
깊이 영상 0 움직임 정보
텍스쳐 영상 1 움직임 정보
깊이 영상 1 움직임 정보
텍스쳐 영상 2 움직임 정보
깊이 영상 2 움직임 정보

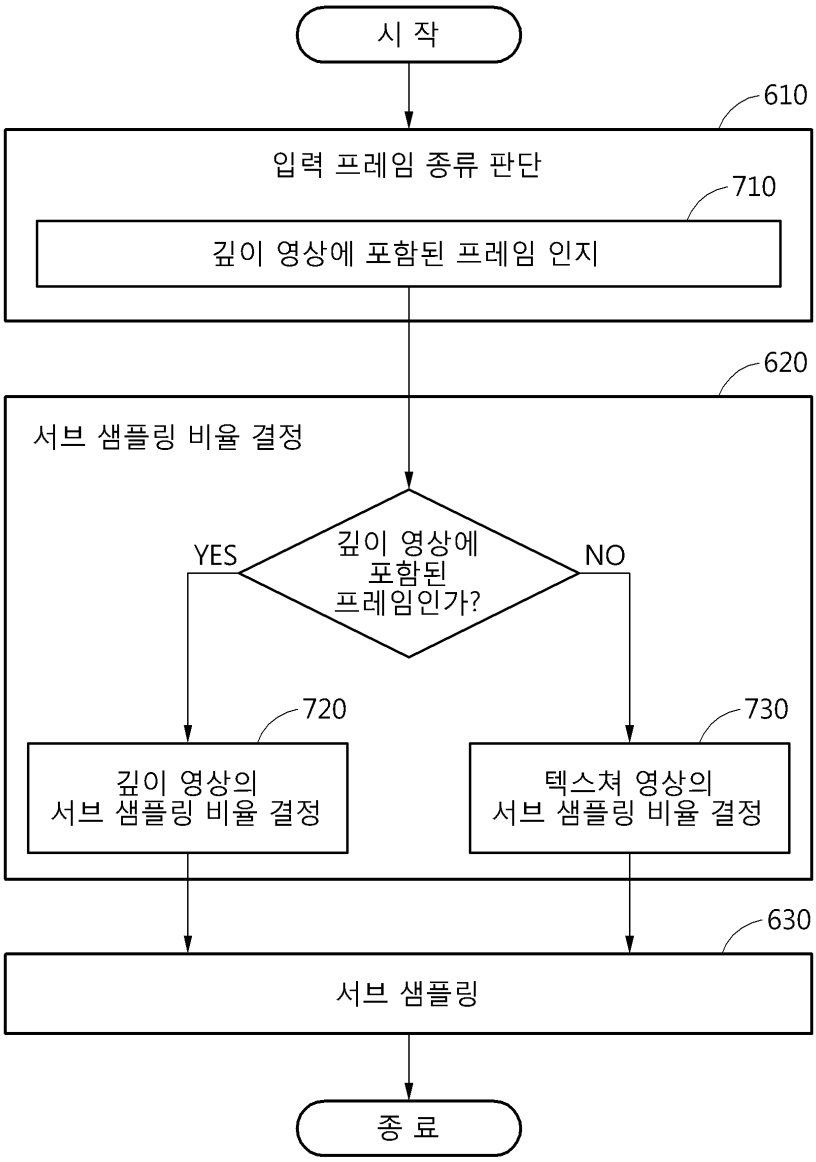
도면5


텍스쳐 영상 : 1/16, 깊이 영상 : 1/256
510
텍스쳐 영상 : 1/4, 깊이 영상 : 1/256
520
텍스쳐 영상 0 움직임 정보
깊이 영상 0 움직임 정보
텍스쳐 영상 1 움직임 정보
깊이 영상 1 움직임 정보
텍스쳐 영상 2 움직임 정보
깊이 영상 2 움직임 정보

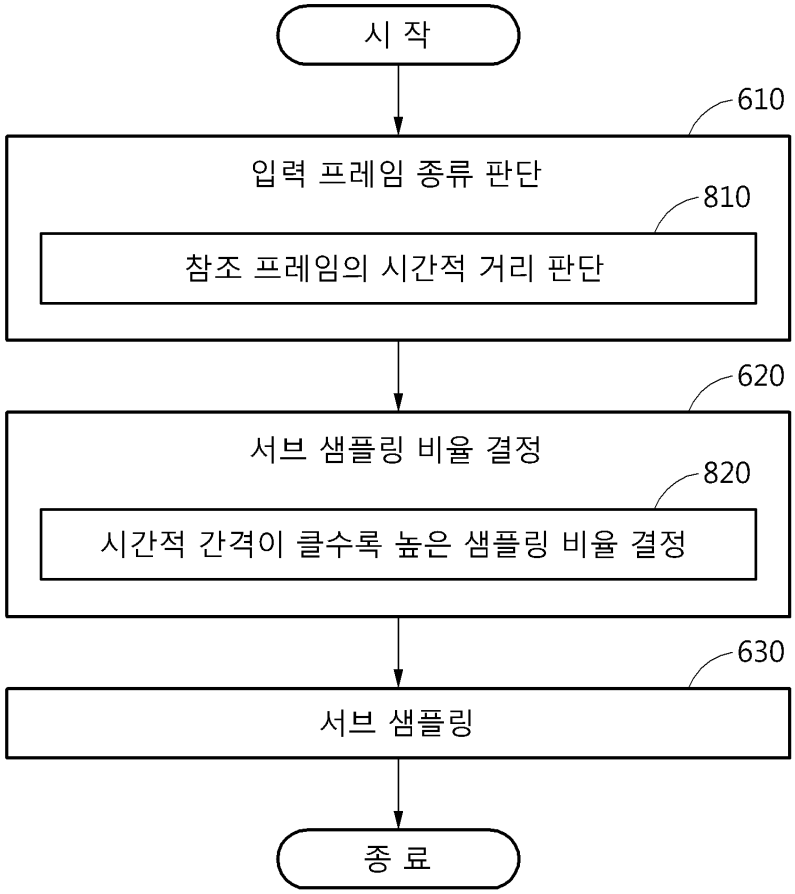
도면6



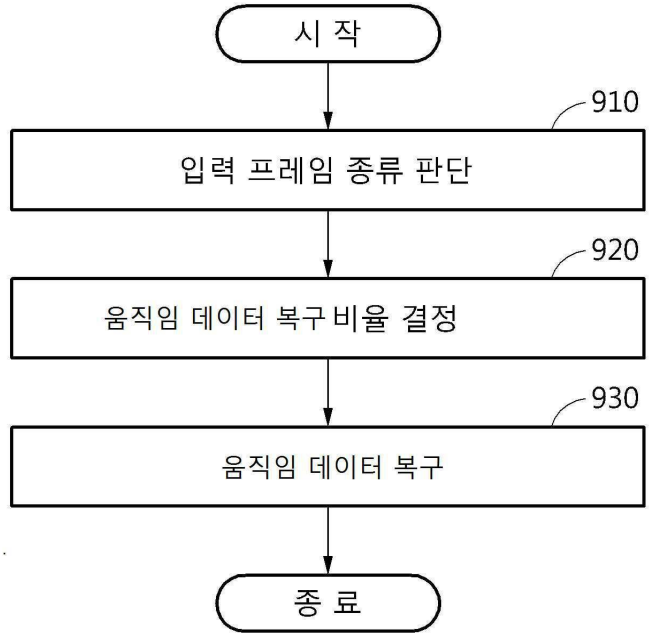
도면7


시 작
610
입력 프레임 종류 판단
710
깊이 영상에 포함된 프레임 인지
620
서브 샘플링 비율 결정
깊이 영상에 포함된 프레임인가?
YES
NO
720
깊이 영상의 서브 샘플링 비율 결정
730
텍스쳐 영상의 서브 샘플링 비율 결정
630
서브 샘플링
종 료

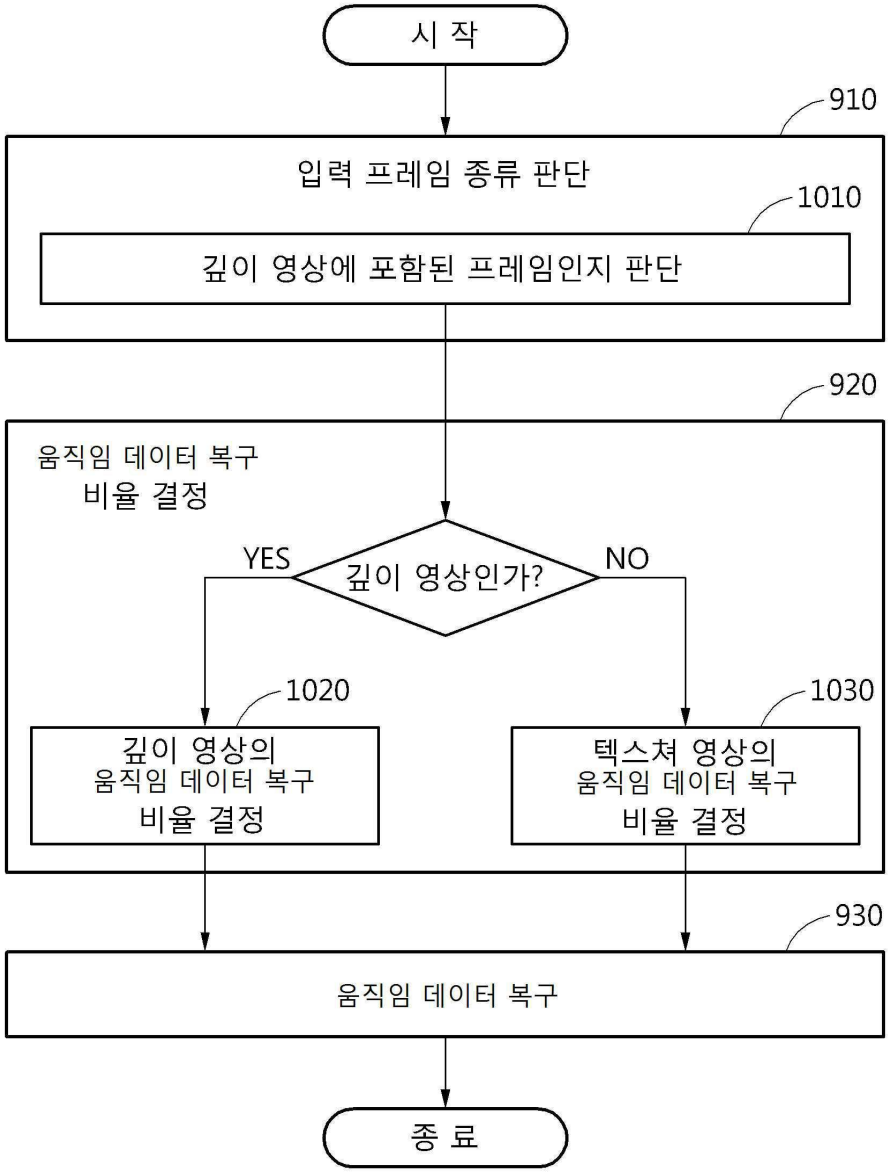
도면8


시 작
610
입력 프레임 종류 판단
810
참조 프레임의 시간적 거리 판단
620
서브 샘플링 비율 결정
820
시간적 간격이 클수록 높은 샘플링 비율 결정
630
서브 샘플링
종 료

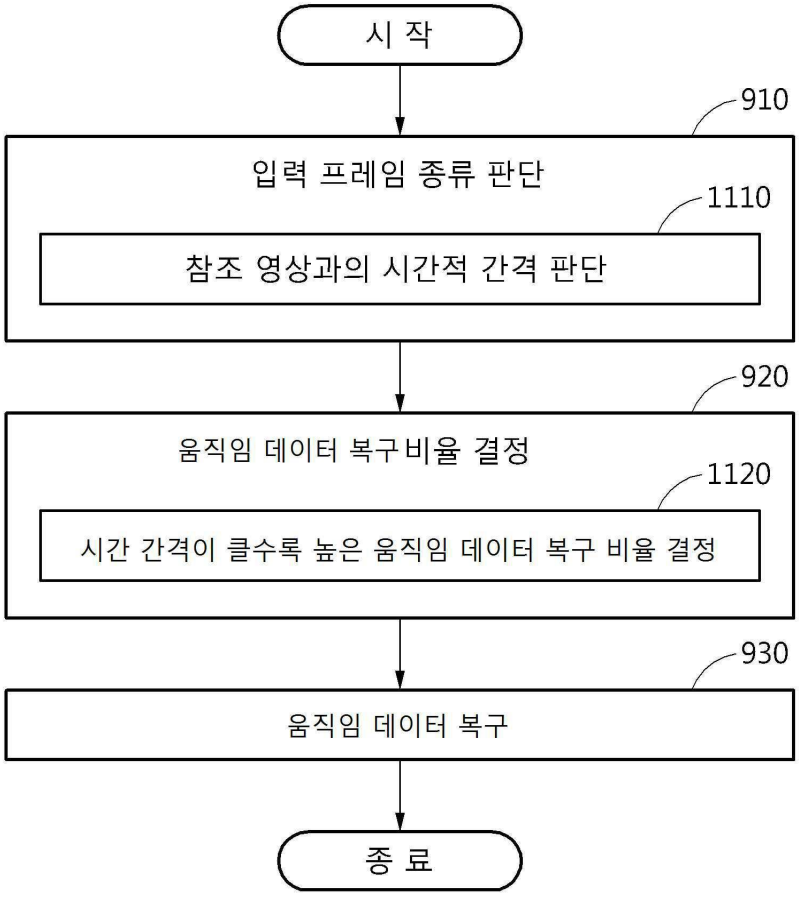
도면9


시 작
910
입력 프레임 종류 판단
920
움직임 데이터 복구 비율 결정
930
움직임 데이터 복구
종 료

도면10


시 작
910
입력 프레임 종류 판단
1010
깊이 영상에 포함된 프레임인지 판단
920
움직임 데이터 복구
비율 결정
YES
깊이 영상인가?
NO
1020
깊이 영상의
움직임 데이터 복구
비율 결정
1030
텍스쳐 영상의
움직임 데이터 복구
비율 결정
930
움직임 데이터 복구
종 료

도면11


시 작
910
입력 프레임 종류 판단
1110
참조 영상과의 시간적 간격 판단
920
움직임 데이터 복구비율 결정
1120
시간 간격이 클수록 높은 움직임 데이터 복구 비율 결정
930
움직임 데이터 복구
종 료