(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2025년06월19일
(11) 등록번호 10-2822928
(24) 등록일자 2025년06월16일

(51) 국제특허분류(Int. Cl.)
H04N 21/231 (2011.01) *G06N 3/092* (2023.01)
H04N 21/81 (2011.01)
(52) CPC특허분류
H04N 21/231 (2013.01)
G06N 3/092 (2023.01)
(21) 출원번호 10-2024-0032203
(22) 출원일자 2024년03월06일
심사청구일자 2024년03월06일
(56) 선행기술조사문헌
KR1020200126822 A*
US20200177939 A1*
US20210295726 A1*
US20210192792 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
송민석
서울특별시 서초구 남부순환로 2311-12, 107동 402호 (방배동, 래미안방배아트힐)
이춘광
인천광역시 부평구 경인로 805-1, 101호 (십정동)
(74) 대리인
양성보

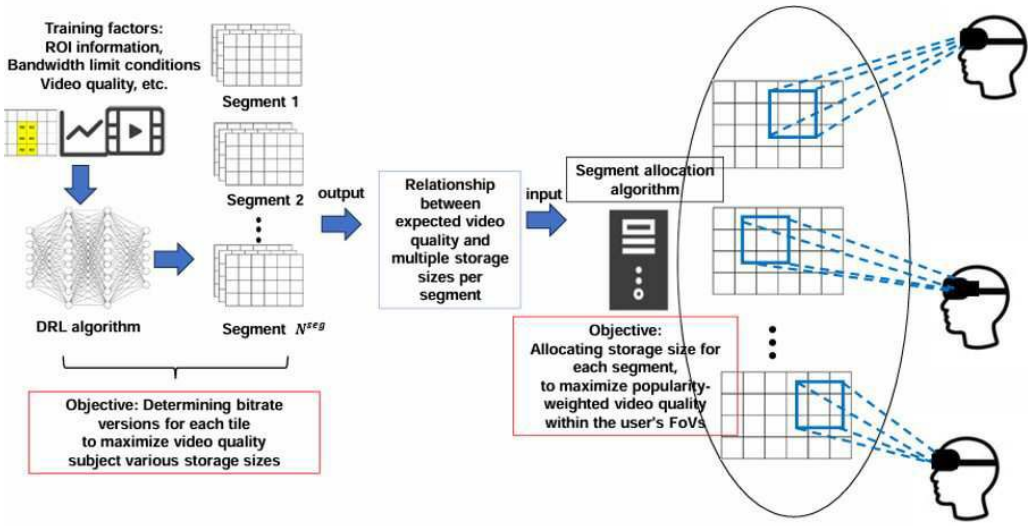
전체 청구항 수 : 총 4 항 심사관 : 우정훈

(54) 발명의 명칭 **심층 강화 학습(DRL)을 사용하여 비디오 품질을 최적화하는 360도 비디오 타일 스토리지 관리 기법**

(57) 요 약

심층 강화 학습(DRL)을 사용하여 비디오 품질을 최적화하는 360도 비디오 타일 스토리지 관리 기법이 개시된다. 일 실시예에 따른 스토리지 관리 시스템에 의해 수행되는 비디오 스토리지 관리 방법은, 스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링하는 단계; 및 상기 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정하는 단계를 포함할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
H04N 21/816 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711198227
과제번호 RS-2023-00261534
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송기술 국제공동연구
연구과제명 강화학습을 활한 대규모 비디오 스트리밍 시스템의 에너지 최적화 기술 개발
과제수행기관명 인하대학교 산학협력단
연구기간 2023.07.01 ~ 2023.12.31
공지예외적용 : 있음

명 세 서

청구범위

청구항 1

스토리지 관리 시스템에 의해 수행되는 비디오 스토리지 관리 방법에 있어서,

스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링하는 단계; 및

상기 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정하는 단계

를 포함하고,

상기 결정하는 단계는,

서로 다른 관심 영역을 가진 훈련 세그먼트에 대해 대역폭 제한 및 FOV 분포를 기반으로 비디오 품질을 최대화하는 세그먼트에 할당된 스토리지 크기에 대한 값을 탐색하도록 학습하고, 상기 학습이 종료됨에 따라 구축된 학습 모델을 이용하여 세그먼트에 저장할 비트율 버전을 결정하고, 스토리지 용량 제한 조건 내에서 각 세그먼트에 대해 스토리지 크기 인덱스를 선택하여 인기도 가중 비디오 품질(popularity-weighted video quality, PWVQ)을 최대화하는 단계를 포함하고,

상기 심층 강화 학습 환경은, 결투 DQN(dueling deep Q-network, DQN)을 포함하고, 지정된 스토리지 용량 제한 제약 조건 내에서 각 타일의 버전 조합을 결정하는 행동 공간, 에이전트가 대역폭과 타일 접근율을 학습할 수 있는 관찰 공간, 행동에 대한 기대 비디오 품질을 도출하는 보상 모델로 구성되는 것

을 포함하는 비디오 스토리지 관리 방법.

청구항 2

삭제

청구항 3

제1항에 있어서,

상기 결정하는 단계는,

세그먼트에 할당된 스토리지 크기에 기초하여 비디오 품질을 최대화하는 세그먼트에 저장할 비트율 버전을 결정하는 세그먼트 버전 결정 문제를 정의하는 단계

를 포함하는 비디오 스토리지 관리 방법.

청구항 4

삭제

청구항 5

삭제

청구항 6

제1항에 있어서,

상기 결정하는 단계는,

각 세그먼트에 대해 스토리지 크기 인덱스의 값을 초기화하고, 인기도 가중 비디오 품질의 값이 최대화될 때까지 스토리지 크기 인덱스의 값을 반복적으로 수정하는 단계

를 포함하는 비디오 스토리지 관리 방법.

청구항 7

스토리지 관리 시스템에 있어서,

스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링하는 모델링부; 및

상기 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정하는 버전 결정부

를 포함하고,

상기 버전 결정부는,

서로 다른 관심 영역을 가진 훈련 세그먼트에 대해 대역폭 제한 및 FOV 분포를 기반으로 비디오 품질을 최대화하는 세그먼트에 할당된 스토리지 크기에 대한 값을 탐색하도록 학습하고, 상기 학습이 종료됨에 따라 구축된 학습 모델을 이용하여 세그먼트에 저장할 비트율 버전을 결정하고, 스토리지 용량 제한 조건 내에서 각 세그먼트에 대해 스토리지 크기 인덱스를 선택하여 인기도 가중 비디오 품질(popularity-weighted video quality, PWVQ)을 최대화하는 것을 포함하고,

상기 심층 강화 학습 환경은, 결투 DQN(dueling deep Q-network, DQN)을 포함하고, 지정된 스토리지 용량 제한 제약 조건 내에서 각 타일의 버전 조합을 결정하는 행동 공간, 에이전트가 대역폭과 타일 접근율을 학습할 수 있는 관찰 공간, 행동에 대한 기대 비디오 품질을 도출하는 보상 모델로 구성되는 것

을 포함하는 스토리지 관리 시스템.

발명의 설명

기 술 분 야

[0001] 아래의 설명은 비디오 스토리지 관리 기술에 관한 것이다.

배 경 기 술

[0003] 360도 비디오는 각 세그먼트가 여러 개의 타일로 구성되어 있고, 각 타일은 다시 여러 버전으로 트랜스코딩되어 저장되기 때문에 상당한 스토리지 공간이 필요하다. 따라서, 모든 트랜스코딩 버전을 저장하는 것은 비현실적이므로 제한된 스토리지 공간을 효율적으로 활용하는 것이 필수적이다.

[0004] 타일 기반 적응형 스트리밍 방식은 360도 비디오 스트리밍 서비스에서 널리 사용되어 왔으며, 많은 연구에서 타일 기반 적응형 스트리밍 방식을 조사하고 개선하기 위해 노력하고 있다. 예를 들어, 제한된 네트워크 대역폭에서 스트리밍 품질을 향상시키는 것을 목표로 서버 측 비디오 분석을 통해 시맨틱(semantic) 정보를 활용하여 미래 FoV 예측을 향상시키는 360도 비디오 스트리밍 시스템이 제안된 바 있다. 또한, 다양한 네트워크 조건에서 예측 오류와 사용자 머리 움직임에 따라 타일 품질을 최적화하는 360도 비디오의 FoV 적응형 스트리밍 기법이 소개되었다. 또한, FoV 예측 오류를 완화하기 위해 설계된 확률론적 타일 프리페칭 시스템이 제시된 바 있다. 그러나, 이러한 연구들은 주로 네트워크 대역폭을 최적화하기 위해 설계되어 360도 스트리밍의 전반적인 품질에 영향을 미칠 수 있는 서버 측 리소스 제약에 대해 고려를 하지 않았다.

[0005] 360도 비디오 스트리밍 시스템의 비트율 래더를 구체적으로 결정하기 위해 FoV 적응형 스트리밍 시스템을 위한 360도 비디오를 준비하기 위한 이론적 모델이 탐구되었다. 이 연구에서는 비디오 품질이 강조되는 영역의 공간 크기와 총 비트율 간의 절충점을 다루었으나, 비디오 인기도와 스토리지 크기 제한 조건을 고려하지 않고 있다.

[0006] 또한, 타일 기반 360도 비디오 시스템에서 비트율 래더를 결정하기 위한 휴리스틱 알고리즘이 제시되었다. 여기에는, 다양한 대역폭을 가진 클라이언트의 비디오 품질을 향상시키기 위한 클래스별 최적화와 전체 스토리지 제한을 충족하도록 클래스별 솔루션을 조정하는 글로벌 최적화가 포함되었다. 이 연구는 정확한 타일 인기도와 네트워크 조건을 미리 알고 있다는 가정 하에 작동한다. 그러나 실제로는 사용자와 콘텐츠에 따라 FoV가 달라

지고 네트워크 상태를 예측할 수 없으므로 오프라인에서 정확한 결정을 내리는 것은 사실상 불가능하다.

[0007] 그러나 이러한 접근 방식 중 어느 것도 서버 측의 스토리지 제한 제약 조건 하에서 비트율 래더를 구성하도록 특별히 맞춤화된 것은 없었다. 이로 인해, 네트워크 상태에 따라 달라지는 대역폭 요구 사항, 타일 접근율, 콘텐츠에 따라 달라지는 비디오 품질 등 다양한 요소를 통합해야 하는 문제가 있어서 비효율적이다.

[0008] 이에, 스토리지 제한 내에서 타일 버전을 결정할 때 심층 강화 학습을 사용하여 FoV 및 네트워크 조건과 관련된 불확실성을 효과적으로 해결하기 위한 기술이 요구된다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 360도 비디오 서버에서 발생하는 높은 스토리지 요구량 문제를 효과적으로 해결하고자 한다. 보다 상세하게는, 네트워크 상태에 따라 달라지는 대역폭 요구 사항, 타일 접근율, 콘텐츠에 따른 비디오 품질, 비디오 세그먼트 인기도 등의 다양한 변수를 고려하여 360도 비디오 스토리지 제한 조건에서 비디오 품질을 최대화하는 심층 강화 학습(Deep Reinforce Learning)을 활용한 새로운 스토리지 공간 관리 기법을 제안한다.

과제의 해결 수단

[0012] 스토리지 관리 시스템에 의해 수행되는 비디오 스토리지 관리 방법은, 스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링하는 단계; 및 상기 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정하는 단계를 포함할 수 있다.

[0013] 상기 심층 강화 학습 환경은, 결투 DQN(dueling deep Q-network, DQN)을 포함하고, 지정된 스토리지 용량 제한 제약 조건 내에서 각 타일의 버전 조합을 결정하는 행동 공간, 에이전트가 대역폭과 타일 접근율을 학습할 수 있는 관찰 공간, 행동에 대한 기대 비디오 품질을 도출하는 보상 모델로 구성될 수 있다.

[0014] 상기 결정하는 단계는, 세그먼트에 할당된 스토리지 크기에 기초하여 비디오 품질을 최대화하는 세그먼트에 저장할 비트율 버전을 결정하는 세그먼트 버전 결정 문제를 정의하는 단계를 포함할 수 있다.

[0015] 상기 결정하는 단계는, 서로 다른 관심 영역을 가진 훈련 세그먼트에 대해 대역폭 제한 및 FOV 분포를 기반으로 비디오 품질을 최대화하는 세그먼트에 할당된 스토리지 크기에 대한 값을 탐색하도록 학습하고, 상기 학습이 종료됨에 따라 구축된 학습 모델을 이용하여 세그먼트에 저장할 비트율 버전을 결정하는 단계를 포함할 수 있다.

[0016] 상기 결정하는 단계는, 스토리지 용량 제한 조건 내에서 각 세그먼트에 대해 스토리지 크기 인덱스를 선택하여 인기도 가중 비디오 품질(popularity-weighted video quality, PWVQ)을 최대화하는 인기도 인식 스토리지 할당 문제(Popularity-aware storage allocation problem, PSAP)를 정의하는 단계를 포함할 수 있다.

[0017] 상기 결정하는 단계는, 각 세그먼트에 대해 스토리지 크기 인덱스의 값을 초기화하고, 인기도 가중 비디오 품질의 값이 최대화될 때까지 스토리지 크기 인덱스의 값을 반복적으로 수정하는 단계를 포함할 수 있다.

[0018] 스토리지 관리 시스템은, 스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링하는 모델링부; 및 상기 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정하는 버전 결정부를 포함할 수 있다.

발명의 효과

[0020] 스토리지 제한, 대역폭 변화, FoV 분포에서 전반적인 인기도 가중 비디오 품질을 향상시킬 수 있다.

도면의 간단한 설명

[0022] 도 1은 일 실시예에 있어서, 스토리지 관리 시스템을 설명하기 위한 도면이다.

도 2는 일 실시예에 있어서, 기대 비디오 품질을 도출할 수 있는 품질 결정 알고리즘의 의사코드이다.

도 3은 일 실시예에 있어서, SVDA 훈련 단계를 설명하기 위한 알고리즘이다.

도 4는 일 실시예에 있어서, SVDA 결정 단계를 설명하기 위한 알고리즘이다.

도 5는 일 실시예에 있어서, 스토리지 할당 알고리즘을 설명하기 위한 도면이다.

도 6은 일 실시예에 있어서, 360도 비디오 세그먼트에서 측정된 평균 VMAF 값을 설명하기 위한 도면이다.

도 7은 일 실시예에 있어서, SVDP 실험에서 사용한 하이퍼 파라미터를 설명하기 위한 도면이다.

도 8은 일 실시예에 있어서, 스토리지 제한 조건에 따른 타 기법 대비 제안 기법의 PWVQ을 설명하기 위한 그래프이다.

도 9는 일 실시예에 있어서, Zipf 파라미터에 따른 타 기법 대비 제안 기법의 PWVQ을 설명하기 위한 그래프이다.

도 10는 일 실시예에 있어서, μ^{FoV} 값에 따른 타 기법 대비 제안 기법의 PWVQ을 설명하기 위한 그래프이다.

도 11은 일 실시예에 있어서, σ^{FoV} 값에 따른 PWVQ을 설명하기 위한 도면이다.

도 12는 일 실시예에 있어서, 대역폭 제한 분포의 평균값에 따른 타 기법 대비 제안 기법의 PWVQ을 설명하기 위한 그래프이다.

도 13은 일 실시예에 있어서, 스토리지 관리 시스템의 구성을 설명하기 위한 블록도이다.

도 14는 일 실시예에 있어서, 비디오 스토리지 관리 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0025] 실시예에서는 시야각(field of view, FoV) 분포에 기반한 결투(형) 심층 Q 네트워크(DQN) 알고리즘과 전체 비디오 인기도를 고려하는 탐욕 알고리즘을 결합한 새로운 스토리지 공간 관리 동작에 대하여 설명하기로 한다. 먼저, 에이전트가 다양한 저장 용량 제한 조건에서 최대 비디오 품질을 얻기 위해 각 타일의 버전을 결정할 수 있는 결투 DQN 환경을 모델링한다. 결투 DQN 환경은 1) 지정된 스토리지 제한 내에서 각 타일의 버전 조합을 결정하는 행동 공간, 2) 에이전트가 다양한 대역폭과 타일 접근율을 학습할 수 있는 관찰 공간, 3) 다양한 행동에 대한 기대 비디오 품질을 도출하는 보상 모델로 구성된다. 저장 용량 제한과 기대 비디오 품질을 연관시키는 결투 DQN 모델을 기반으로, 인기도 가중 비디오 품질을 최대화하기 위해 저장 용량 제한 내에서 여러 비디오 세그먼트 중에서 버전을 선택하는 탐욕적인 알고리즘을 제시한다. 시뮬레이션 실험을 통해 다양한 스토리지 제한, 대역폭 변화, FoV 분포에서 다른 기법에 비해 0.75%에서 33.79%(평균 13.7%)에 이르는 전반적인 인기도 가중 비디오 품질 개선이 입증되었다.

[0026] 도 1은 일 실시예에 있어서, 스토리지 관리 시스템을 설명하기 위한 도면이다.

[0027] 도1은 360도 비디오를 저장하는 스토리지 관리 시스템의 아키텍처를 나타낸 것이다. 각 비디오는 세그먼트로 나뉘며, 각 세그먼트는 다시 타일로 세분화된다. 그런 다음, 각 타일을 서로 다른 비트율 버전으로 트랜스코딩할 수 있다. 일반 비디오에 비해 360도 세그먼트의 전체 스펙트럼을 표현하는 데 필요한 스토리지 요구량이 증가하기 때문에 모든 비트율 버전을 저장하는 것은 비현실적이다. 따라서, 본 발명의 주요 목적은 사용 가능한 스토리지 공간의 제한 내에서 저장에 적합한 비트율 버전을 선택하는 것이다.

[0028] N^{seg} 를 총 세그먼트의 수라고 하고 각 세그먼트는 N^{tile} 직사각형 타일로 분할되며, 각 타일은 최대 N^{ver} 개의 서로 다른 비트율 버전을 가질 수 있다. 설명을 쉽게 하기 위해 세그먼트 i에서 j 번째 타일의 k 번째 비트율 버전, $(i = 1, \ldots, N^{\mathrm{seg}}, j = 1, \ldots, N^{\mathrm{tile}}, \text{ and } k = 1, \ldots, N^{\mathrm{ver}})$ 를 나타내는 $v_{i,j,k}$를 도입한다.

[0029] $Q_{i,j,k}$, $S_{i,j,k}$ 및 $B_{i,j,k}$ 는 각각 버전 $V_{i,j,k}$ 의 비디오 품질, 파일 크기 및 비트율을 나타낸다. 넷플릭스에서 개발한 비디오 다중 방법 평가 융합(video multi-method assessment fusion, VMAF) 메트릭은 QoE(quality of experience)를 정확하게 표현하는 데 널리 사용되고 효과적이기 때문에 비디오 품질을 정량화하

는 데 사용한다.

[0030] 360도 비디오 스트리밍 시스템에서 일반적으로 사용자는 세그먼트의 일부 타일만 볼 수 있으며, 이러한 FoV 타일은 사용자의 관심을 이끌기 때문에 관심 영역(ROI) 타일과 밀접한 관계를 보인다. 일반적으로 비디오 세그먼트 처리를 통해 식별할 수 있는 관심 객체가 포함된 비디오의 영역을 ROI로 지정할 수 있다. 360도 비디오 세그먼트 간의 콘텐츠 차이로 인해 ROI 타일의 위치는 세그먼트마다 다르다.

[0031] 물론, ROI 타일은 사람들의 시선을 사로잡는 이미지를 포함하므로 FoV 영역의 일부가 될 가능성이 높지만, 이 확률은 콘텐츠의 특성과 사용자의 개인적 성향에 크게 영향을 받는다. 따라서, FoV의 중앙 타일이 ROI 타일 집합에 속할 확률은 확률적 정규 분포 $\mathcal{N}(\mu^{\mathrm{FoV}}, \sigma^{\mathrm{FoV}})$로 표시된다.

[0032] 네트워크 상태는 시간과 위치에 따라 달라질 수 있다. 예를 들어, 인터넷 속도가 빠른 경우 사용자는 높은 비트율 버전의 FoV 타일을 수신할 수 있다. 반대로 사용 가능한 대역폭이 충분하지 않은 경우에는 서버에 저장되어 있더라도 높은 비트율 버전의 타일을 FoV로 재생하는 것이 비현실적일 수 있다. 따라서, 다양한 네트워크 조건을 수용하려면 적절한 비트율 버전을 사용자에게 제공해야 한다. 사용자가 요구하는 이러한 비트율 조건을 모델링하기 위해 $\mathcal{N}(\mu^{\mathrm{bw}}, \sigma^{\mathrm{bw}})$로 표시되는 정규 분포를 사용한다.

[0033] 비디오 세그먼트의 인기도는 일반적으로 Zipf 분포를 따르며, 여기서, p_i는 비디오 세그먼트 i의 인기도, $(i = 1, \ldots, N^{\mathrm{seg}})$를 나타낸다. Zipf 분포에서 소수의 비디오 세그먼트는 인기가 높을 것으로 예상되는 반면, 대다수는 인기가 상당히 낮을 것으로 예상된다. 주문형 비디오(VoD) 시스템에서는 비디오 인기를 쉽게 예측할 수 있으며 비교적 안정적으로 유지된다. 따라서, p_i는 선험적으로 알려져 있다고 가정한다.

[0034] VR 기기의 특성에 따라 일반적으로 직사각형으로 표시되는 FoV의 타일 수를 결정할 수 있다. 사용자가 FoV에만 집중하면 이 영역에 대해서만 전송이 최적화되고 나머지 타일에 대해서는 가장 낮은 비트율 버전이 전송된다. 비디오 품질 계산에서는 FoV 영역만 고려한다. FoV 영역 내의 개별 타일 간에 VMAF가 달라지면 시청 경험이 저하될 수 있다. 따라서, 세그먼트 내의 개별 타일의 비디오 품질을 평가하기보다는 전체 세그먼트의 전체 비디오 품질을 평가하는 것이 바람직하다. 접근 방식은 FoV 영역 내 모든 타일의 VMAF 평균 점수에서 표준 편차를 감산하여(빼서) 각 세그먼트의 비디오 품질을 결정하는 것이다.

[0035] 1. 전반적인 문제 공식화

[0036] 이진 변수 $X_{i,j,k}$를 도입하여 $V_{i,j,k}$의 저장 여부를 표현한다. $X_{i,j,k} = 1\$$이면 $V_{i,j,k}$이 서버에 저장되고, 그렇지 않으면 저장되지 않는다. 그러면, 각 세그먼트 i에 대해 $X_{i,j,k} = 1$인 비트율 버전의 집합은 다음과 같은 표현을 통해 $\vec{V_i}$로 나타낼 수 있다.

[0037] $$\vec{V_i} = \{V_{i,j,k} | X_{i,j,k} = 1 . \} .$$

[0038] 최고 비트율 버전이 있어야 하기 때문에 트랜스코딩을 통해 더 낮은 비트율 버전을 쉽게 생성할 수 있도록 최고 비트율 버전을 저장해야 한다고 가정한다. 또한, 어떤 네트워크 조건에서도 세그먼트를 전송할 수 있도록 하려면 가장 낮은 비트율 버전을 저장하는 것이 필수적이다. 따라서, 다음 조건을 충족해야 한다.

[0039] $$\forall i \text{ and } j, X_{i,j,1} = 1 \text{ and } X_{i,j,N^{\mathrm{ver}}} = 1 .$$

[0040] 요청에 대한 FoV 분포와 대역폭 분포를 고려할 때 모든 타일 버전의 누적 비트율은 사용 가능한 대역폭을 초과해서는 안 되며 서버에 저장되어야 한다. 실제로 이것은 세그먼트 i의 전송으로 인한 비디오 품질이 $\vec{V_i}$의 함수이며, $Q_i(\vec{V_i})$로 표현된다는 것을 의미한다.

[0041] S^{limit} 은 타일 버전에 대한 서버의 스토리지 공간 제한 제약 조건(최고 및 최저를 제외)을 나타낸다.

[0042] $S_{i,j,k}$ 는 버전 $V_{i,j,k}$ 의 스토리지 크기를 메가바이트(MB) 단위로 나타낸다. 스토리지 공간 제한 제약 조건은 다음과 같이 표현할 수 있다.

[0043] $$\sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{tile}} \sum_{k=2}^{N^{ver}-1} X_{i,j,k} S_{i,j,k} \leq S^{limit} \quad (1)$$

[0044] 전반적인 비디오 품질을 향상시키려면 인기 있는 세그먼트가 더 높은 비디오 품질로 전송되도록 하는 것이 필수적이다. 이를 정량화하기 위해 인기도 가중 비디오 품질(popularity-weighted video quality, PWVQ)이라는 지표를 도입하여 $p_i Q_i^{exp}(\vec{V_i})$ 로 계산한다. 이제 사용자의 FoV 내에서 PWVQ를 최대화하는 것을 목표로 각 타일에 저장할 비트율 버전을 결정하기 위한 버전 선택 문제(version selection problem, VSP)를 다음과 같이 공식화한다.

[0045] $$\text{Maximize } \sum_{i=1}^{N^{seg}} p_i Q_i^{exp}(\vec{V_i})$$

$$\text{Subject to } \sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{tile}} \sum_{k=2}^{N^{ver}-1} X_{i,j,k} S_{i,j,k} \leq S^{limit}. \quad (2)$$

[0046] 2. 본 발명 솔루션

[0047] VSP의 최적 해는 각 세그먼트의 $X_{i,j,k}$ 조합뿐만 아니라 세그먼트의 인기도와 각 세그먼트에 할당된 스토리지 용량에 따라 달라진다.

[0048] 세그먼트에 스토리지 공간을 할당하면 여러 비트율 버전을 저장할 수 있으므로 결과적으로 전송되는 세그먼트의 비디오 품질이 향상된다.

[0049] 그러나, $X_{i,j,k}$ 의 조합은 대역폭과 FoV 분포의 영향을 받는다. 따라서, VSP를 푸는 것은 계산적으로 어려우므로 다음과 같이 VSP를 두 개의 하위 문제로 나누는 방식을 채택한다.

[0050] 1) 세그먼트 버전 결정 문제(Segment version determination problem, SVDP): SVDP는 다양한 스토리지 제한 제약 조건을 고려하면서 각 타일에 대한 비트율 버전 조합을 결정한다.

[0051] 2) 인기도 인식 스토리지 할당 문제(Popularity-aware storage allocation problem, PSAP): PSAP는 각 세그먼트에 할당된 스토리지 크기를 계산하여 PWVQ를 최대화하는 것을 목표로 한다. 이 프로세스는 스토리지 제한 제약 조건을 준수하면서 비디오의 인기도를 고려한다.

[0052] 3.SVDP정의

[0053] 세그먼트 i에 할당된 최소 스토리지 크기를 $\sum_{j=1}^{N^{tile}} (S_{i,j,1} + S_{i,j,N^{ver}})$ 로 계산한 S_i^{min} 라고 한다. 세그먼트 i에 대해 모든 비트율 버전이 저장된다고 가정하면, 세그먼트 i의 최대 저장 요구량은 S_i^{max} 로 표시되며, $\sum_{j=1}^{N^{tile}} \sum_{k=1}^{N^{ver}} S_{i,j,k}$ 에 의해 계산된다.

[0054] SVDP의 목표는 세그먼트 i에 할당된 스토리지 크기가 최저 및 최고 비트율 버전을 제외한 $S_{i,n}^{seg}$ 일 때 세그먼트 i에 저장할 버전을 결정하는 것이며, 여기서, $(n = 0, \ldots, N^{st_unit})$ 는 스토리지 크기 인덱스의 수를 나타낸다. 그러면, $S_{i,n}^{seg}$ 은 다음과 같이 계산된다.

[0055] $$S_{i,n}^{\text{seg}} = n\frac{S_i^{\max} - S_i^{\min}}{N^{\text{st_unit}}} \quad (3)$$

[0056] 세그먼트 i에 할당된 스토리지 크기가 $S_{i,n}^{\text{seg}}$ 일 때 $X_{i,j,k,n}$의 값으로 $X_{i,j,k}$ 를 정의한다.

[0057] 그러면, 세그먼트 i에 대한 스토리지 크기가 $S_{i,n}^{\text{seg}}$ 인 벡터는 $\overrightarrow{V_{i,n}}$ 로 표시되어 다음과 같이 계산된다.

[0058] $$\overrightarrow{V_{i,n}} = \{V_{i,j,k} | X_{i,j,k,n} = 1\}.$$

[0059] 사용자에게 전달되는 기대 비디오 품질은 $Q_i^{\exp}(\overrightarrow{V_{i,n}})$ 로 표시되며, 벡터 $\overrightarrow{V_{i,n}}$의 함수로 계산할 수 있다.

[0060] SVDP는 세그먼트 i에 대해 $S_{i,n}^{\text{seg}}$ 의 스토리지 공간이 할당될 때 $\overrightarrow{V_{i,n}}$을 결정한다. 목표는 $Q_i^{\exp}(\overrightarrow{V_{i,n}})$를 최대화하는 것이며, 이는 다음과 같이 표현할 수 있다.

[0061] $$\text{Maximize } Q_i^{\exp}(\overrightarrow{V_{i,n}})$$
$$\textit{Subject to} \quad X_{i,j,k,n} = 0,1$$
$$\sum_{j=1}^{N^{\text{tile}}} \sum_{k=2}^{N^{\text{ver}}-1} X_{i,j,k,n} S_{i,j,k} \le S_{i,n}^{\text{seg}}. \quad (4)$$

[0062] 4.SVDP 솔루션

[0063] SVDP를 해결하기 위해 다음과 같은 이유로 DRL 알고리즘을 제안한다.

[0064] -FoV 분포에 따라 가변적인 FoV 위치: FoV는 분명 ROI와 관련이 있지만, 사용자 및 콘텐츠 특성에 따라 예측하기 어렵다는 문제가 있다. 따라서, ROI와 분포 모델을 모두 기반으로 FoV를 생성하는 것이 중요하며, 후속 분석 결과도 이에 따라 통합되어야 한다. 그러나, 휴리스틱 알고리즘을 사용할 때 고정된 FoV를 선택하면 효과가 제한될 수 있다.

[0065] -선택한 비트율-버전 조합에 따라 비디오 품질이 가변적이다: 세그먼트의 비디오 품질은 단순히 각 타일의 VMAF 점수를 합산하여 계산되는 대신 FoV의 개별 타일에 걸친 VMAF 점수의 편차를 고려해야 한다. 따라서, 각 세그먼트의 비디오 품질 지수를 정확하게 계산하기 위해서는 각 타일에 대한 비트율을 설정해야 한다.

[0066] -대역폭 제한 조건에 따라 비디오 품질이 가변적이다: 각 FoV 타일에 대해 선택되는 버전은 대역폭 조건에 따라 달라지며, 이로 인해 비디오 품질에 차이가 발생한다. 비트율 선택 결정은 네트워크 대역폭에 기반한 비트율 선택 알고리즘을 실행하여 비트율과 VMAF점수를 동시에 고려해야 한다. 따라서, 기존의 휴리스틱 기법으로는 비디오 품질을 결정론적으로 예측하는 것이 불가능하다고 간주된다.

[0067] 5. PSAP정의

[0068] SVDP를 풀면 비디오 품질 $Q_i^{\exp}(\overrightarrow{V_{i,n}})$와 해당 $S_{i,n}^{\text{seg}}$ 값$\left(n = 0, \ldots, N^{\text{st_unit}}\right)$의 관계를 정립할 수 있다.

[0069] 이를 기반으로 각 세그먼트 i에 대해 스토리지 크기 인덱스 Y_i, 여기서 $\left(Y_i = 0, \ldots, N^{\text{st_unit}}\right)$를 선택하도록 PSAP를 공식화할 수 있다. 목표는 다음과 같이 스토리지 제한 제약 조건이 적용되는 모든 세그먼트에서 $\sum_{i=1}^{N^{\text{seg}}} p_i Q_i^{\exp}(\overrightarrow{V_{i,Y_i}})$로 표현되는 전체 PWVQ를 극대화하는 것이다.

$$\text{Maximize} \sum_{i=1}^{N^{\text{seg}}} p_i Q_i^{\text{exp}}\left(\overrightarrow{V_{i,Y_i}}\right),$$

$$\textit{Subject to} \quad Y_i \in 1,..,N^{\text{st_unit}}$$

$$\sum_{i=1}^{N^{\text{seg}}} S_{i,Y_i}^{\text{seg}} \leq S^{\text{limit}}. \qquad (5)$$

[0070]

[0071] 6.PSAP솔루션

[0072] PSAP는 다중 배낭 문제(multiple choice knapsack problem, MCKP)로 축소할 수 있기 때문에 NP난이도로 간주된다. MCKP에서 물건은 각각 무게와 이득이 있는 아이템 집합으로 구성된다. 목표는 각 물건에서 정확히 하나의 물건을 골라 배낭에 넣는 것으로, 배낭 무게 제한을 준수하면서 총 이득을 극대화하는 것이다. 비슷한 맥락에서 PSAP는 스토리지를 배낭으로 개념화하여, 스토리지 용량의 제약 내에서 각 세그먼트 i에 대한 Y_i 값을 선택하여 전체 PWVQ를 최대화하는 것을 목표로 한다. 탐욕 알고리즘이 MCKP를 처리하는 데 효과적인 것으로 알려져 있으므로, 본 발명은 탐욕적인 방식을 선택한다.

[0073] C. SDVP 알고리즘 설계 및 구현

[0074] 결투 DQN(dueling deep Q-network, DQN)은 강화학습에 특화된 신경망 구조를 활용하여 식 (4)와 같이 SVDP에서 $Q_{i,j,k,n}^{\text{exp}}\left(\overrightarrow{V_{i,n}}\right)$의 최대화를 목표로 $X_{i,j,k,n}$ 값을 결정하는 데 사용되며, 이는 기존 DQN 대비 노이즈에 대해 강력한 학습 효과를 보이기 때문이다. 결투 DQN에서 상태값 함수는 미래의 행동과 관계없이 현재 상태의 가치를 표현하는 반면, 이득 함수는 주어진 상태에서 현재 행동이 다른 행동에 비해 상대적인 가치를 나타낸다. 이 두 함수를 기반으로 각 상태의 Q값은 신경망을 통해 지속적으로 업데이트되며 손실 최소화라는 목표를 향해 노력한다.

[0075] SVDA는 훈련 단계와 결정 단계로 구성된다. 훈련 단계에서 결투 DQN은 학습 과정의 결과로 Q값을 업데이트하는 데 사용되며, 에이전트가 최종적으로 가장 높은 Q값을 달성할 수 있도록 한다.

[0076] N^{sample}이 훈련 세그먼트의 수를 나타내는 경우, 결투 DQN은 각 훈련 세그먼트에 대해 N^{episode}에피소드에 걸쳐 학습을 진행한다. 따라서, 이 과정은 훈련 모델을 구축하는 데 필요한 총 $N^{\text{sample}}N^{\text{episode}}$에피소드를 누적한다.

[0077] 에이전트는 서로 다른 ROI를 가진 각 훈련 세그먼트 ii에 대해 대역폭 제한 및 FoV 분포를 기반으로 $Q_{i,n}^{\text{exp}}\left(\overrightarrow{V_{i,n}}\right)$의 최대화를 위해 $S_{i,n}^{\text{seg}}$의 다양한 값에 대한 $X_{i,j,k,n}$ 값을 찾도록 훈련받게 된다. 훈련 단계가 끝나면 학습 모델이 설정되고, 이 모델은 결정 단계에서 모든 세그먼트에 걸쳐 $S_{i,n}^{\text{seg}}$과 $Q_{i,n}^{\text{exp}}\left(\overrightarrow{V_{i,n}}\right)$간의 관계를 설정하는 데 사용할 수 있다.

[0078] 각 에피소드는 에이전트가 환경의 현재 상태에 대한 관찰을 수신한 후 해당 상태에 따라 조치를 취하는 일련의 시간 단계로 구성된다. 그런 다음, 에이전트는 해당 행동에 대한 보상을 받아 환경을 새로운 상태로 전환하고 누적 보상 합계를 최대화하기 위해 학습한다. 따라서, 훈련 및 결정 단계는 행동 공간, 관찰 공간, 보상 모델로 구성된다.

[0079] 행동 공간 A는 다음과 같이 비트율 버전의 집합으로 표현된다.

[0080] $$A = \{V_{i,1,2}, \ldots, V_{i,1,N^{\text{ver}}-1}, \ldots, V_{i,N^{\text{tile}},2}, \ldots, V_{i,N^{\text{tile}},N^{\text{ver}}-1}\}$$

[0081] 시간 단계 t에서 결투 DQN 알고리즘은 행동 $a_{i,t}$를 선택한다. 예를 들어, $a_{i,t} = V_{i,j,k}$이면 에이전트가

저장할 것으로 결정되는 것은 $V_{i,j,k}$ 이다.

[0082] 각 에피소드가 시작될 때 알고리즘 실행 중 스토리지 공간을 기록하는 임시 변수(즉, S^{tmp})가 0으로 초기화된다. 각 동작(예를 들면, $V_{i,j,k}$)이 수행될 때마다 S^{tmp}가 $S^{\text{tmp}} + S_{i,j,k}$ 로 업데이트하여 현재 사용된 스토리지 크기를 기록한다. 이 단계는 세그먼트 i 의 스토리지 공간 제한(즉, $S_{i,n}^{\text{seg}}$)을 초과하지 않을 때까지 반복된다. 따라서, $S^{\text{tmp}} > S_{i,n}^{\text{seg}}$ 직전에 에피소드가 완료된 것으로 간주한다.

[0083] 시간 단계 t, S_t 에서의 관측 공간은 상태 집합으로 표현할 수 있다.

[0084] $$S_t^{\text{obj}} = \{\overrightarrow{A_{i,t}^{\text{cur}}}, S_t^{\text{tmp}}, Q_i^{\text{VMAF}}, S_i^{\text{size}}, S_{i,n}^{\text{seg}}, S^{\text{bw}}, S^{\text{ROI}}\}$$

[0085] 여기서 각 상태는 S_t^{obj} 의 각 상태는 다음과 같이 요약할 수 있다.

[0086] - $\overrightarrow{A_{i,t}^{\text{cur}}} = \{a_{i,1}, \ldots, a_{i,t}\}$ 이므로, $\overrightarrow{A_{i,t}^{\text{cur}}}$ 는 에이전트가 1단계에서 t 단계까지 선택한 동작의 시퀀스를 나타낸다.

[0087] - S_t^{tmp} 는 시간 단계 $\$t\$$ 에서 $\overrightarrow{A_{i,t}^{\text{cur}}}$ 의 작업 시퀀스에 해당하는 버전이 차지하는 스토리지 크기를 나타낸다.

[0088] - $Q_i^{\text{VMAF}} = \{Q_{i,1,1}, \ldots Q_{i,N^{\text{tile}},N^{\text{ver}}}\}$, 여기서, $Q_{i,j,k}$ 는 버전 $V_{i,j,k}$ 에 대한 VMAF 값을 나타낸다.

[0089] - $S_i^{\text{size}} = \{S_{i,1,1}, \ldots, S_{i,N^{\text{tile}},N^{\text{ver}}}\}$, 여기서, $S_{i,j,k}$ 는 버전 $V_{i,j,k}$ 의 파일 크기(MB)를 나타낸다.

[0090] - $S_{i,n}^{\text{seg}}$ 는 방정식~(3)을 사용하여 계산할 수 있다.

[0091] -대역폭 분포 $\mathcal{N}(\mu^{\text{bw}}, \sigma^{\text{bw}})$에 따라 N^{req} 대역폭 한계값 집합이 생성되어 $S_i^{\text{bw}} = \{B_1, \ldots, B_{N^{\text{req}}}\}$ 의 상태를 형성한다.

[0092] - $S_i^{\text{ROI}} = \{I_{i,1}^{\text{ROI}}, \ldots, I_{i,N^{\text{tile}}}^{\text{ROI}}\}$, 여기서, $I_{i,j}^{\text{ROI}}$ 는 세그먼트 i의 타일 j가 ROI 타일인지 아닌지를 나타내는 이항 변수이다.

[0093] 에이전트가 누적 보상을 극대화하는 것을 목표로 학습할 수 있도록 각 행동 선택에 대해 보상 모델을 제공해야 한다. SVDP의 목표가 비디오 품질을 극대화하는 것이므로 에이전트가 시간 단계 t에서 최대 값인 $Q_i^{\exp}\left(\overrightarrow{A_{i,t}^{\text{cur}}}\right)$를 달성하도록 보상 모델을 설계해야 한다. 그러나, $Q_i^{\exp}\left(\overrightarrow{A_{i,t}^{\text{cur}}}\right)$는 각 타일의 비트율 버전 조합, FoV 및 대역폭 분포에 따라 상당한 변동성을 보인다.

[0094] 따라서, 유일한 결정론적 비디오 품질 값을 도출하는 것은 불가능하며, 시뮬레이션 기반 알고리즘을 사용하여 $Q_i^{\exp}\left(\overrightarrow{A_{i,t}^{\text{cur}}}\right)$의 최대 기대값을 확률적으로 계산해야 한다.

[0095] 이를 위해 도 2에 설명된 대로 $\overline{A_{i,t}^{\text{cur}}}$ 를 기반으로 예상 최대 비디오 품질을 결정하는 품질 결정 알고리즘(QDA)을 개발한다. 대역폭 제한 및 FoV 분포를 확률적으로 시뮬레이션하기 위해 N^{req} 요청 집합을 생성한다. 예를 들어, $\mathcal{N}(\mu^{\text{FoV}}, \sigma^{\text{FoV}})$ 를 기반으로 $S_{i,m}^{\text{FoV}}$ 로 표시되는 FoV 타일 인덱스 집합과 $S_{i,m}^{\text{Non-FoV}}$ 로 표시되는 비 FoV 타일 인덱스 세트가 요청 $m(m = 1, \ldots, N^{\text{req}})$ 세그먼트 i의 요청에 대해 N^{req} 요청 집합을 생성한다. 또한, $\mathcal{N}(\mu^{\text{bw}}, \sigma^{\text{bw}})$ 에 따라 대역폭 제한 값 집합이 생성되어 $S^{\text{bw}} = \{B_1^{\text{lim}}, \ldots, B_{N^{\text{req}}}^{\text{lim}}\}$ 가 형성된다. 이후, 각 요청 m에 대해 $(m = 1, \ldots, N^{\text{req}})$, 각 타일 j에 대한 버전 인덱스 $(j = 1, \ldots, N^{\text{tile}})$ 를 대역폭 제한 조건 B_m^{lim} 을 준수하면서 선택해야 FoV 타일의 비디오 품질을 최대화하는 것을 목표로 한다.

[0096] 임시 변수 z_j, $(j = 1, \ldots, N^{\text{tile}})$ 는 알고리즘 실행 과정에서 j 번째 타일에 대해 전송된 버전 인덱스를 기록하는 데 사용된다. 당연히, 가장 낮은 버전이 전달되는 비 FoV 타일의 경우 z_j 는 1로 설정된다. $b_{i,j,k}$ 가 버전 $V_{i,j,k}$ 의 비트율을 나타내면, 요청 $m(m = 1, \ldots, N^{\text{req}})$ 에 대해 다음 조건이 유지되어야 한다.

[0097] $$\sum_{j=1}^{N^{\text{tile}}} b_{i,j,z_j} \leq B_m . (6)$$

[0098] FoV 타일의 비디오 품질을 최대화하면서 식 (6)을 만족하는 것은 MCKP의 변형에 해당할 수 있으며, 이 경우 탐욕적인 솔루션이 매우 효과적이다. 따라서, QDA는 다음과 같이 탐욕적인 방식으로 설계되었다.

[0099] -z_j 는 FoV 타일에 대해 가장 높은 버전 인덱스인 N^{ver} 로 초기화된다. 식(6)이 만족되지 않으면, 필요한 조치는 부등식(6)이 만족될 때까지 z_j 를 감소시키는 탐욕 단계를 구현하는 것이다.

[0100] -세그먼트 i의 각 FoV 타일 j에 대해 $H_{i,j,k}$, $(k = 1, \ldots, N^{\text{ver}} - 1)$ 의 변수는 다음과 같이 정의된다.

[0101] $$H_{i,j,k} = \frac{Q_{i,j,z_j} - Q_{i,j,k}}{B_{i,j,z_j} - B_{i,j,k}}$$

[0102] 비디오 품질 손실($H_{i,j,k}$ 의 분자)를 최소화하면서 최대 대역폭 감소($H_{i,j,k}$ 의 분모)를 얻기 위해, $H_{i,j,k}$ 값의 목록 L에서 가장 낮은 값을 선택한 다음 z_{j^*} 의 값을 k^* 로 줄인다.

[0103] -식 (6)이 만족될 때까지 앞서 언급한 단계가 반복된다.

[0104] 앞서 언급한 세 단계를 완료하면 z_j 의 값을 확정하여 세그먼트의 비디오 품질 지수를 결정할 수 있다. 이러한 품질 지수는 기본적으로 FoV 타일의 비디오 품질 지수를 합산하여 설정되지만, 타일 간 비디오 품질의 변동은 QoE에 악영향을 미칠 수 있다. 따라서, μ_m^{req} 와 σ_m^{req} 가 각각 요청 m에 대한 FoV 타일 전반의 VMAF 값의 평균

과 표준 편차를 나타내면 요청 m에 대한 비디오 품질은 μ_m^{req}와 σ_m^{req} 간 뺄셈을 통해 계산할 수 있다. 마지막으로, 각 요청에 대한 이러한 품질 지수의 평균으로 확률적으로 기대되는 비디오 품질 값 $Q_i^{exp}\left(\overrightarrow{A_{i,t}^{cur}}\right)$를 구할 수 있으며, 이는 $\frac{\sum_{m=1}^{N^{req}} \mu_m^{req} - \sigma_m^{req}}{N^{req}}$로 계산할 수 있다.

[0105] 세그먼트 i에 대한 시간 단계 t의 보상은 $r_{i,t}$로 표시되며, 에이전트가 수행한 행동으로 인해 비디오 품질이 향상됨을 나타낸다. 따라서, $Q_i^{exp}\left(\overrightarrow{A_{i,t}^{cur}}\right)$와 $Q_i^{exp}\left(\overrightarrow{A_{i,t-1}^{cur}}\right)$의 차이에 따라 계산할 수 있다. 설명을 쉽게 하기 위해 다음과 같이 가정한다: $Q_i^{exp}\left(\overrightarrow{A_{i,0}^{cur}}\right) = 0$이라고 가정한다. 그러면 이 차이는 다음과 같이 QDA의 실행을 통해 결정된다.

$$r_{i,t} = Q_i^{exp}\left(\overrightarrow{A_{i,t}^{cur}}\right) - Q_i^{exp}\left(\overrightarrow{A_{i,t-1}^{cur}}\right) \quad (7)$$

[0106]

[0107] 도 3은 일 실시예에 있어서, SVDA 훈련 단계를 설명하기 위한 알고리즘이고, 도 4는 일 실시예에 있어서, SVDA 결정 단계를 설명하기 위한 알고리즘이다.

[0108] 훈련 단계에서 에이전트는 도 3에 설명된 절차와는 별개로 결투 DQN 방식에 따라 학습 효율을 높이기 위해 Q값을 독립적으로 업데이트한다.

[0109] N^{sample}이 훈련 세그먼트의 수라면, 합성 워크로드는 다음과 같이 각 세그먼트에 대해 생성될 수 있다. FoV 타일과 대역폭 제한 값은 각각 분포 $\mathcal{N}(\mu^{FoV}, \sigma^{FoV})$, $\mathcal{N}(\mu^{bw}, \sigma^{bw})$를 사용하여 생성된다. 그 후, N^{ROI} 타일을 무작위로 선택하면 S_i^{ROI} 집합을 형성할 수 있다. 그런 다음, 0와 N^{st_unit}사이에서 무작위로 선택된 스토리지 인덱스에 대해 총 $N^{episode}$의 훈련 작업이 수행되며, 각 작업은 스토리지 제한 제약 조건이 충족될 때까지 계속된다. 학습 단계가 성공적으로 완료되면 학습 모델을 설정할 수 있으며, 학습 모델은 이후 결정 단계에서 저장 장치에 저장된 실제 비트율 버전을 결정하는 데 사용된다.

[0110] 결정 단계에서는 도 4에 설명된 대로 세그먼트 i와 스토리지 인덱스 n에 대해 $\overrightarrow{V_{i,n}}$와 $Q_i^{exp}(\overrightarrow{V_{i,n}})$를 계산하기 위해 훈련 모델이 로드된다. 이를 위해, 각 세그먼트에 대한 실제 정보가 훈련 모델의 입력 파라미터로 전달된다. 이에 대한 응답으로 모델은 각 세그먼트 i와 스토리지 인덱스 n에 대한 $\overrightarrow{V_{i,n}}$와 $Q_i(\overrightarrow{V_{i,n}})$의 값을 제공한다. 이러한 값은 다음 섹션에서 설명하는 대로 스토리지 할당 알고리즘에서 효과적으로 사용할 수 있다.

[0111] 도 5는 일 실시예에 있어서, 스토리지 할당 알고리즘을 설명하기 위한 도면이다.

[0112] PSAP를 해결하기 위해 아래와 같이 초기화 및 업데이트 단계로 구성된 도 5에 설명된 대로 스토리지 할당 알고리즘(storage allocation algorithm, SAA)이라는 탐욕 알고리즘을 개발한다.

[0113] -초기화: Y_i의 모든 값은 각 세그먼트 i에 대해 N^{st_unit}로 임시 초기화되며, 이는 각 타일의 모든 비트율 버전을 저장하여 가장 큰 스토리지 공간을 할당한다는 것을 나타낸다. 이를 위해서는 할당된 스토리지 공간을 줄여야 하는데, 이는 스토리지 공간의 양을 S^{limit} 이하로 줄임으로써 달성할 수 있지만 이러한 감소는 비디오 품질 저하로 이어진다.

[0114] -갱신: 이 갱신 단계는 $\sum_{i=1}^{N^{seg}} S_{i,Y_i}^{seg}$ 까지 Y_i 값을 반복적으로 수정하면서 예상 PWVQ 값인 $\sum_{i=1}^{N^{seg}} p_i Q_i^{exp}(\overrightarrow{V_{i,Y_i}})$ 가 최대화될 때까지 반복적으로 수정한다. 이를 위해 $G_{i,n}$의 변수를 사용한다. $(n = 0, \ldots, Y_i - 1)$의 변수는 Y_i에서 n으로 저장 인덱스가 감소할 때마다 다음과 같이 정의된다.

$$G_{i,n} = p_i \frac{Q_i^{exp}(\overrightarrow{V_{i,Y_i}}) - Q_i^{exp}(\overrightarrow{V_{i,n}})}{S_{i,Y_i}^{seg} - S_{i,n}^{seg}} \quad (8)$$

[0115]

[0116] 비디오 품질 저하($G_{i,n}$ 의 분자)를 최소화하면서 가능한 한 많은 스토리지 공간($G_{i,n}$의 분모)을 확보하기 위해 $G_{i,n}$ 값의 목록 L에서 가장 낮은 G_{i^*,n^*}의 값이 선택된다. 그 후, Y_{i^*}의 값이 n^*로 감소하고 $G_{i,n}$의 값이 업데이트된다. 이 단계는 $\sum_{i=1}^{N^{seg}} S_{i,Y_i}^{seg} \leq S^{limit}$ 가 될 때까지 반복된다.

[0117] 발명의 효과를 평가하기 위해 시뮬레이션 실험이 수행될 수 있다. 기본 구성으로 N^{tile}은 24로 고정하고, N^{ver}은 7로 설정하고 각 타일의 비트율이 다음과 같이 지정되도록 구성했다. 0.2Mbps, 0.4Mbps, 0.6Mbps, 1Mbps, 1.5Mbps, 2Mbps. 각 버전 $V_{i,j,k}$에 대해 VMAF 값은 2~14 사이의 표준 편차를 갖는 정규 분포에 따라 생성된다. 각 타일의 평균 VMAF 값은 150개의 세그먼트에서 측정되었으며, 각 세그먼트에는 24개의 타일이 포함되어 있으며, 결과는 도 6에 자세히 나와 있다.

[0118] 설명을 쉽게 하기 위해 S^{limit}의 값은 달성 가능한 최대 스토리지 크기 대비 백분율로 표시된다. 이 최대 크기는 모든 버전이 저장되는 시나리오에 해당하며, $\sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{tile}} \sum_{k=2}^{N^{ver}-1} S_{i,j,k}$로 계산된다.

[0119] 기본 설정으로 R^{st}로 표시되는 이 백분율 값은 40%로 설정된다. 세그먼트 내의 ROI 타일 수는 6~9개 사이에서 무작위로 선택되므로 세그먼트 간 가변성을 허용한다. 또한, N^{st_unit}은 10의 값으로 구성된다. FoV는 직사각형으로 표시되며 3×3 사각형 타일로 구성된다.

[0120] 비디오 인기도 분포는 기울기를 나타내는 매개변수 θ가 있는 Zipf 분포를 따른다. θ 값이 낮을수록 인기도가 더 적은 비디오 집합에 집중된다. θ의 기본 설정은 0.3이며, 실제 VoD 데이터에서 도출된 측정값을 기준으로 가장 가까운 10분의 1로 반올림된다.

[0121] 확률 분포 $\mathcal{N}(\mu^{FoV}, \sigma^{FoV})$의 기본 파라미터 값으로 μ^{FoV}=0.7 및 σ^{FoV}=0.1이 있다. 각 사용자의 대역폭 수요를 나타내는 대역폭 분포의 파라미터인 $\mathcal{N}(\mu^{bw}, \sigma^{bw})$의 경우, μ^{bw}는 13Mbps로 설정하고 σ^{bw}는 2Mbps로 고정한다.

[0122] 훈련 과정에서 500개의 개별 비디오 세그먼트가 사용되었으며, 각 세그먼트는 350개의 에피소드로 훈련되어 누적 총 175,000개의 에피소드가 생성되었다. 이러한 매개변수 외에도 훈련 과정에 사용된 하이퍼 파라미터는 도 7에 요약되어 있다.

[0123] 비디오 품질을 극대화하도록 특별히 설계된 다섯 가지 벤치마크 접근 방식에 대해 본 발명에서 제안한 방식을

비교 분석했다. 이 벤치마크들에서는 각 버전 $V_{i,j,k}$에 매개변수 $G_{i,j,k}$가 할당되며, $p_i \times \frac{Q_{i,j,k}}{S_{i,j,k}}$로 계산된다. 저장할 버전을 선택할 때 $G_{i,j,k}$ 값이 더 큰 버전을 우선적 선택하는데, 이는 탐욕적인 접근 방식이 일반적으로 배낭 문제를 해결하는 데 효과적이기 때문이다. 이를 바탕으로 다음과 같이 설명한다:

[0124] -ROI 타일 우선 탐욕 기법(RTF): RTF는 저장 제한 조건인 식 (1)을 위반하지 않는 한 처음에 $G_{i,j,k}$ 값이 더 높은 ROI 타일 버전만 저장한다. 모든 ROI 타일을 고려한 후에도 스토리지 공간이 남아있는 경우, $G_{i,j,k}$ 값이 더 높은 비ROI 타일은 식 (1)이 만족될 때까지 저장한다.

[0125] -ROI 타일 우선순위 지정 없는 탐욕 기법(GPR): GPR은 RTF와 유사하지만 ROI 타일에 우선순위를 지정하지 않는다는 점에서 차이가 있다.

[0126] -ROI 타일과 가까운 거리 선호 탐욕 기법(GFN): RTF와 마찬가지로 GFN도 처음에는 ROI 타일에 우선순위를 둔다. 그러나, 비ROI 타일을 처리할 때 ROI 타일과의 거리를 반영하도록 매개변수 $G_{i,j,k}$를 수정하여 RTF에서 벗어난다. 구체적으로, $G_{i,j,k}$는 모든 ROI 타일로부터 $V_{i,j,k}$의 평균 거리로 나누는 방식으로 수정된다.

[0127] -인기 세그먼트 우선 기법(PSF): PSF는 식 (1)이 만족될 때까지 가장 인기 있는 세그먼트의 모든 버전을 순차적으로 저장하여 인기 있는 세그먼트의 우선 순위를 지정한다.

[0128] -ROI 타일에 가장 가까운 거리 우선 기법(NRF): NRF는 처음에 가장 낮은 버전부터 가장 높은 버전까지 오름차순으로 ROI 타일의 버전을 저장한다. 그 다음에는 마찬가지로 ROI 타일과 가장 가까운 가장 바깥쪽에 있는 타일의 버전을 저장한다. 이 프로세스는 계속되며, 식 (1)이 만족될 때까지 각 후속 바깥쪽에 동일한 방식으로 버전이 저장된다.

[0129] 본 발명에서 제안한 기법은 타 기법과 비교한 결과는 다음과 같다

[0130] 1. 스토리지 제한 조건에 따른 효과

[0131] 도8과 같이 저장 용량 제한 조건의 영향을 조사한다. 본 발명의 접근 방식은 두 가지 알고리즘(SVDA와 SAA)을 통합하므로 이를 SVDA+SAA라고 부른다. 모든 스토리지 제한 제약 조건에서 SVDA+SAA는 벤치마크 방식에 비해 0.75%에서 29.26%까지의 PWVQ가 개선되고 우수한 성능을 보여준다. 이러한 차이는 스토리지 크기가 작을 때(예를 들면, $R^{st} = 20\%$) 특히 두드러지며, 이는 SVDA+SAA가 스토리지 크기 제한을 해결하는 데 더 효과적이라는 점을 강조한다.

[0132] 그러나, $R^{st} = 80\%$에서는 스토리지가 모든 방식에서 대부분의 비트율 버전을 수용할 수 있으므로 성능 격차가 최소화된다.

[0133] 다른 방식 중에서도 PSF는 ROI 타일에 우선순위를 두지 않기 때문에 일반적으로 가장 낮은 성능을 보인다. 그러나, $R^{st} = 20\%$에서는 NRF가 최악의 성능을 보이는 반면, PSF는 상대적으로 더 나은 성능을 보인다. 이는 제한된 스토리지 크기를 다룰 때 세그먼트 인기가 가장 중요한 요소가 된다는 것을 시사한다. 다른 방식 중에서도 GFN이 다른 방식보다 우수한 성능을 보이는 이유는 ROI 타일과 비ROI 타일 저장 간의 균형을 유지하여 전반적인 성능을 향상시킬 수 있기 때문이다. PSF와 NRF의 PWVQ는 $R^{st} = 20\%$에서 크게 저하되는데, 이는 ROI 타일에 우선순위를 두기 때문에 비ROI 타일에 상대적으로 소홀해져 성능에 악영향을 미치기 때문이다.

[0134] 2. Zipf 파라미터에 따른 효과.

[0135] 도 9를 다양한 Zipf 파라미터에 따른 PWVQ의 변화를 보여준다.

[0136] 결과에 따르면 SVDA+SAA는 다른 벤치마크를 능가하며 6.6%에서 33.8%까지 PWVQ가 향상되었음을 알 수 있다. 이

격차는 θ 값이 높을수록 더욱 두드러지는데, 이는 θ 값이 증가하면 비디오 인기도의 영향이 줄어들어 SVDA의 FoV 기반 비디오 품질 예측의 정확성이 강조되기 때문이다.

[0137] 이 격차는 θ 값이 높을수록 더욱 두드러지는데, 이는 θ 값의 증가가 비디오 인기도의 영향을 감소시켜 SVDA의 FoV 기반 비디오 품질 예측의 정확도가 더욱 높아진다는 것을 나타낸다.

[0138] 다른 방식 중에서도 GFN은 Zipf 매개변수에 관계없이 일관되게 최고의 성능을 보여준다.

[0139] 3. FoV 분포에 따른 효과.

[0140] 도 10은 FoV 분포가 PWVQ에 미치는 영향을 살펴보기 위해 FoV 분포의 다양한 평균값 (μ^{FoV})에 따른 PWVQ의 변화를 보여준다. 이전 실험에서와 마찬가지로, SVDA+SAA는 다른 벤치마크보다 9.5%에서 22.7%까지 더 높은 PWVQ 값으로 우수한 성능을 보여준다. μ^{FoV} 가 감소함에 따라 ROI와 FoV 간의 상관관계가 감소하여 모든 방법에서 PWVQ가 감소한다. 이 감소 효과는 SVDA+SAA에서 가장 작으며, μ^{FoV}=0.3일 때 가장 큰 격차가 발생한다.

[0141] μ^{FoV} 가 감소함에 따라 RTF 및 NRF 방식의 성능은 저하된다. 이는 FoV 분포의 평균값이 감소하기 때문이며, 이는 ROI 타일에 집중되기보다는 FoV가 더 분산되어 있음을 나타낸다. 결과적으로 ROI 타일 저장의 이점이 감소하여 PWVQ가 상대적으로 더 많이 감소한다. 반대로 GFN 방식에서는 이러한 효과가 상대적으로 작다.

[0142] 또한, 도 11에 표시된 것처럼 표준편차(σ^{FoV})가 PWVQ에 미치는 영향도 조사했다. 다른 방식에 비해 9.9%에서 15.9% 범위에서 PWVQ를 증가시키면서 SVDA+SAA가 일관되게 최고의 성능을 보이는 것이 분명하다. μ^{FoV}의 효과와 비교할 때, FoV 분포의 표준편차가 모든 알고리즘의 성능에 미치는 영향은 무시할 수 있을 정도로 미미하다.

[0143] 4. 대역폭 분포에 따른 효과.

[0144] 대역폭 분포가 비디오 품질에 미치는 영향을 평가하기 위해, 도 12는 대역폭 분포의 평균값에 따른 평균 PWVQ의 변화를 보여준다. 이전 실험과 마찬가지로, SVDA+SAA는 다른 방식에 비해 9.9%에서 19.4% 더 높은 PWVQ를 생성하여 우수한 성능을 달성한다. 이 차이는 대역폭을 중간 수준(9Mbps)으로 설정했을 때 가장 두드러진다. 이러한 현상은 대역폭 분포의 평균이 높거나 낮으면 버전 선택의 유연성이 제한되고 결과적으로 DRL 사용의 이점이 감소하기 때문일 수 있다.

[0145] 도 13은 일 실시예에 있어서, 스토리지 관리 시스템의 구성을 설명하기 위한 블록도이고, 도 14는 일 실시예에 있어서, 비디오 스토리지 관리 방법을 설명하기 위한 흐름도이다.

[0146] 스토리지 관리 시스템(100)의 프로세서는 모델링부(1310) 및 버전 결정부(1320)를 포함할 수 있다. 이러한 프로세서의 구성요소들은 스토리지 관리 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소들은 도 14의 비디오 스토리지 관리 방법이 포함하는 단계들(1410 내지 1420)을 수행하는 스토리지 관리 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[0147] 프로세서는 비디오 스토리지 관리 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 스토리지 관리 시스템에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 스토리지 관리 시스템을 제어할 수 있다. 이때 프로세서는 모델링부(1310) 및 버전 결정부(1320) 각각에 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(1410 내지 1420)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[0148] 단계(1410)에서 모델링부(1310)는 스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링할 수 있다. 이때, 심층 강화 학습 환경은 결투 DQN(dueling deep Q-network, DQN)을 포함하고, 지정된 스토리지 용량 제한 제약 조건 내에서 각 타일의 버전 조합을 결정하는 행동 공간, 에이전트가 대역폭과 타일 접근율을 학습할 수 있는 관찰 공간, 행동에 대한 기대 비디오 품질을 도출하는 보상

모델로 구성될 수 있다.

[0149] 단계(1420)에서 버전 결정부(1320)는 모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정할 수 있다. 버전 결정부(1320)는 세그먼트에 할당된 스토리지 크기에 기초하여 비디오 품질을 최대화하는 세그먼트에 저장할 비트율 버전을 결정하는 세그먼트 버전 결정 문제를 정의할 수 있다. 버전 결정부(1320)는 서로 다른 관심 영역을 가진 훈련 세그먼트에 대해 대역폭 제한 및 FOV 분포를 기반으로 비디오 품질을 최대화하는 세그먼트에 할당된 스토리지 크기에 대한 값을 탐색하도록 학습하고, 학습이 종료됨에 따라 구축된 학습 모델을 이용하여 세그먼트에 저장할 비트율 버전을 결정할 수 있다. 버전 결정부(1320)는 스토리지 용량 제한 조건 내에서 각 세그먼트에 대해 스토리지 크기 인덱스를 선택하여 인기도 가중 비디오 품질(popularity-weighted video quality, PWVQ)을 최대화하는 인기도 인식 스토리지 할당 문제(Popularity-aware storage allocation problem, PSAP)를 정의할 수 있다. 버전 결정부(1320)는 각 세그먼트에 대해 스토리지 크기 인덱스의 값을 초기화하고, 인기도 가중 비디오 품질의 값이 최대화될 때까지 스토리지 크기 인덱스의 값을 반복적으로 수정할 수 있다.

[0150] 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.

[0151] 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 장치에 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.

[0152] 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.

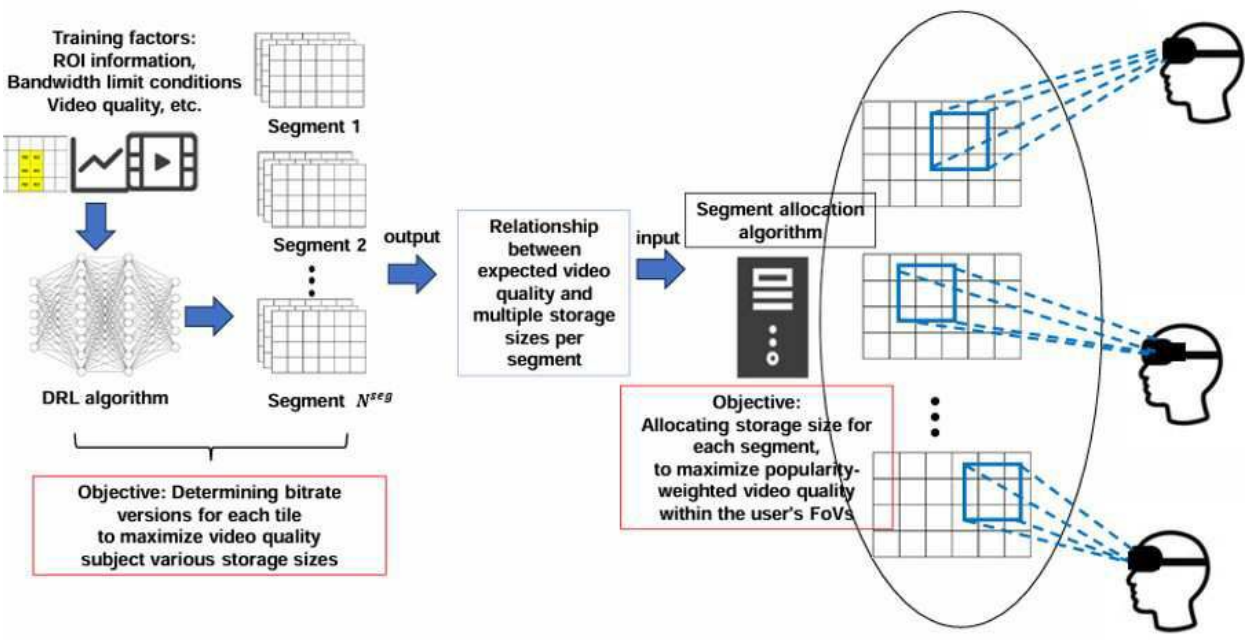
[0153] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0154] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한

다.

도면

도면1


Training factors:
ROI information,
Bandwidth limit conditions
Video quality, etc.
Segment 1
Segment 2
Segment N^{seg}
DRL algorithm
output
Relationship between expected video quality and multiple storage sizes per segment
input
Segment allocation algorithm
Objective: Determining bitrate versions for each tile to maximize video quality subject various storage sizes
Objective: Allocating storage size for each segment, to maximize popularity-weighted video quality within the user's FoVs

도면2

Algorithm 1: Quality determination algorithm (QDA).

Input: S_i^{ROI}, $\mathcal{N}(\mu^{\mathrm{FoV}}, \sigma^{\mathrm{FoV}})$, $\mathcal{N}(\mu^{\mathrm{bw}}, \sigma^{\mathrm{bw}})$, $A_{i,t}^{\vec{\mathrm{cur}}}$

Output: $Q_{i,n}^{\mathrm{exp}}(A_{i,t}^{\vec{\mathrm{cur}}})$

Sets of tile indices for request m, $(m = 1, ..., N^{\mathrm{req}})$: $S_{i,m}^{\mathrm{FoV}}$ and $S_{i,m}^{\mathrm{Non-FoV}}$;

Set for bandwidth limit values, S^{bw}: $\{B_1^{\mathrm{lim}}, ..., B_{N^{\mathrm{req}}}^{\mathrm{lim}}\}$;

Variables: Z_j, $(j = 1, ..., N^{\mathrm{tile}})$, B^{tmp} ;

for $m = 1$ *to* N^{req} **do**

  $\forall j \in S_{i,m}^{\mathrm{Non-FoV}}$, $Z_j \leftarrow 1$;

  $\forall j \in S_{i,m}^{\mathrm{FoV}}$, $Z_j \leftarrow N^{\mathrm{ver}}$;

  $B^{\mathrm{tmp}} \leftarrow \sum_{j=1}^{N^{\mathrm{tile}}} b_{i,j,Z_j}$;

  while $L \neq \phi$ **do**

    if $B^{\mathrm{tmp}} \leq B_m^{\mathrm{lim}}$ **then**

      break;

    end

    Find the lowest value $H_{i,j^*,k^*} \in L$;

    $L \leftarrow L - \{H_{i,j^*,k^*}\}$;

    $B^{\mathrm{tmp}} \leftarrow B^{\mathrm{tmp}} + b_{i,j^*,k^*} - b_{i,j^*,Z_{j^*}}$;

    $Z_{j^*} \leftarrow k^*$;

  end

  Compute the mean μ_m^{req} and the standard deviation σ_m^{req} of Q_{i,j,Z_j}, for all $j \in S_{i,m}^{\mathrm{FoV}}$;

end

$Q_{i,n}^{\mathrm{exp}}(A_{i,t}^{\vec{\mathrm{cur}}}) \leftarrow \frac{\sum_{m=1}^{N^{\mathrm{req}}} \mu_m^{\mathrm{req}} - \sigma_m^{\mathrm{req}}}{N^{\mathrm{req}}}$;

return $Q_{i,n}^{\mathrm{exp}}(A_{i,t}^{\vec{\mathrm{cur}}})$;

도면3

Algorithm 2: SVDA: training phase.

Input: $\mathcal{N}(\mu^{\mathrm{FoV}}, \sigma^{\mathrm{FoV}})$, $\mathcal{N}(\mu^{\mathrm{bw}}, \sigma^{\mathrm{bw}})$
Output: Dueling DQN training model: $\mathcal{T}$

for $i = 1$ *to* N^{sample} **do**
  Build S_i^{ROI} by randomly selecting N^{ROI} tiles ;
  Generate S^{bw} based on $\mathcal{N}(\mu^{\mathrm{bw}}, \sigma^{\mathrm{bw}})$;
  Generate $S_{i,m}^{\mathrm{FoV}}$ and $S_{i,m}^{\mathrm{Non-FoV}}$, $(m = 1, ..., N^{\mathrm{req}})$ based on $\mathcal{N}(\mu^{\mathrm{bw}}, \sigma^{\mathrm{bw}})$;
  Randomly choose storage index I, $(I = 0, \ldots, N^{\mathrm{st_unit}})$
  for $m = 1$ *to* N^{episode} **do**
    Build an observation space of S_0^{obj};
    $S^{\mathrm{tmp}} \leftarrow 0$;
    for $t = 1$ *to* $n(A)$ **do**
      Based on dueling DQN, action $a_{i,t}$ is selected;
      $\vec{A_{i,t}^{\mathrm{cur}}} \leftarrow \vec{A_{i,t}^{\mathrm{cur}}} \bigcup \{a_{i,t}\}$;
      if $S^{\mathrm{tmp}} > S_{i,I}^{\mathrm{seg}}$ **then**
        Break;
      end
      else
        Update S^{tmp} by adding storage size of $a_{i,t}$;
      end
      Run the QDA to find the value of $Q_{i,n}^{\mathrm{exp}}(\vec{A_{i,t}^{\mathrm{cur}}})$;
      Calculate $r_{i,t}$ using Equation (7) and then pass this value to the agent;
      Update observation space S_t^{obj};
    end
  end
end
return *Dueling DQN training model:* $\mathcal{T}$;

도면4

Algorithm 3: SVDA: decision phase.

Input: Dueling DQN training model: $\mathcal{T}$
Output: $\vec{V_{i,n}}$, $Q_{i,n}^{\mathrm{exp}}(\vec{V_{i,n}})$, $(i = 1, ..., N^{\mathrm{seg}})$, $(n = 1, ..., N^{\mathrm{st_unit}})$

Load dueling DQN training model of $\mathcal{T}$;
for $i = 1$ *to* N^{seg} **do**
  Take $S_{i,j,k}$, $Q_{i,j,k}$, and S_i^{ROI} as input parameters of segment i ;
  Calculate $S_{i,n}^{\mathrm{seg}}$ using Equation (3) ;
  for $n = 1$ *to* $n = N^{\mathrm{st_unit}}$ **do**
    Establish $\vec{V_{i,n}}$ for segment i and storage index n using model $\mathcal{T}$;
    Run QDA using an input parameter of $\vec{V_{i,n}}$ to find $Q_{i,n}^{\mathrm{exp}}(\vec{V_{i,n}})$;
  end
end
return $\forall\ i, n,\ \vec{V_{i,n}},\ Q_{i,n}^{\mathrm{exp}}(\vec{V_{i,n}})$;

도면5

Algorithm 4: Storage Allocation Algorithm (SAA)

Input: $Q_{i,n}^{\exp}(\vec{V_{i,n}})$, $S_{i,n}^{\text{seg}}$, p_i, $(i = 1, ..., N^{\text{seg}}$ and $n = 1, ..., N^{st_unit})$

Output: Y_i, $(i = 1, ..., N^{\text{seg}})$

List of $G_{i,n}$ values: L;
$\forall i$, $Y_i \leftarrow N^{st_unit}$;
$S^{\text{tmp}} = \sum_{i=1}^{N^{\text{seg}}} S_{i,Y_i}$;
while $L \neq \phi$ **do**
  if $S^{\text{tmp}} <= S^{\text{limit}}$ **then**
    break;
  end
  Find the lowest value $G_{i,n}$ for which $i = i^*$, $n = n^*$, and remove G_{i^*,n^*} from L;
  $S^{\text{tmp}} \leftarrow S^{\text{tmp}} - S_{i,Y_i}^{\text{seg}} + S_{i^*,n^*}^{\text{seg}}$;
  $Y_{i^*} \leftarrow n^*$;
  Update all values of $G_{i^*,n}$ for segment i^* in L using Equation (8);
end
return Y_i, $(i = 1, ..., N^{\text{seg}})$

도면6

MEAN VMAF VALUES OF EACH TILE DEPENDING ON THE BITRATE.

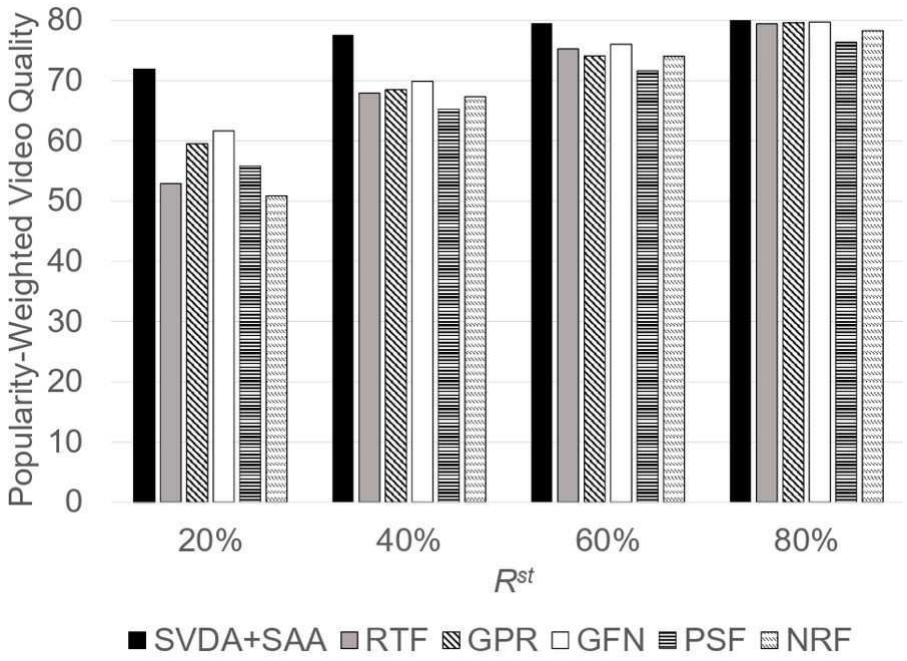
Bitrate (Mbps)	0.2	0.4	0.6	0.8	1	1.5	2
Mean VMAF	29	63	78	85	89	93	100

도면7

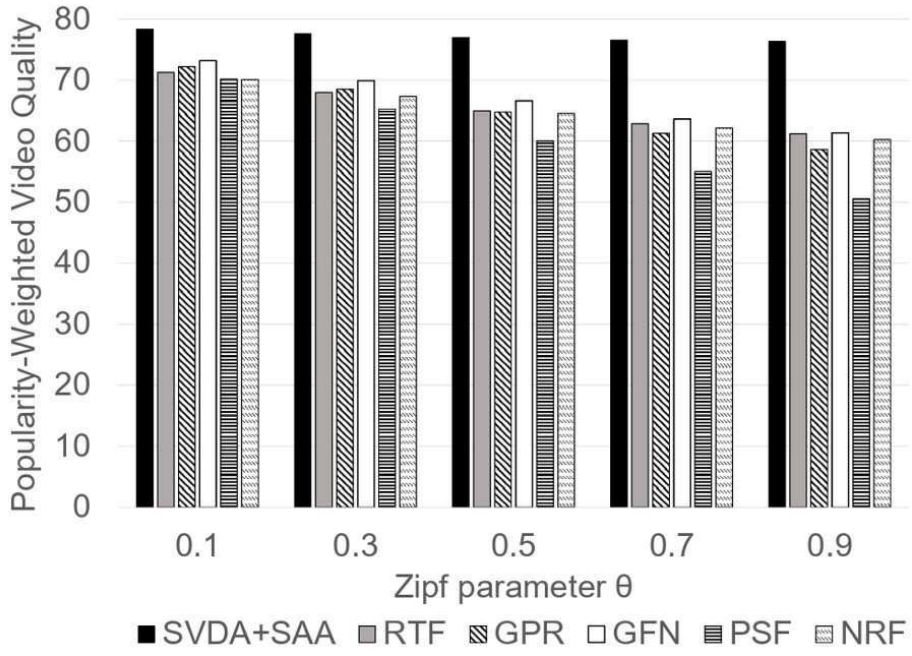
HYPERPARAMETERS USED IN SIMULATIONS.

Hyperparameter	Description	Values
Learning rate	The rate at which parameters are updated based on the loss	0.001
Epsilon value	Probability of choosing the random action instead of the action with the highest Q-value	0.4
Batch size	Number of experiences randomly sampled from the replay buffer	300
Discount factor	Relative weight value assigned to future rewards compared with immediate rewards	0.9
Replay buffer size	Size of replay buffer, utilized for storing past experiences	2000

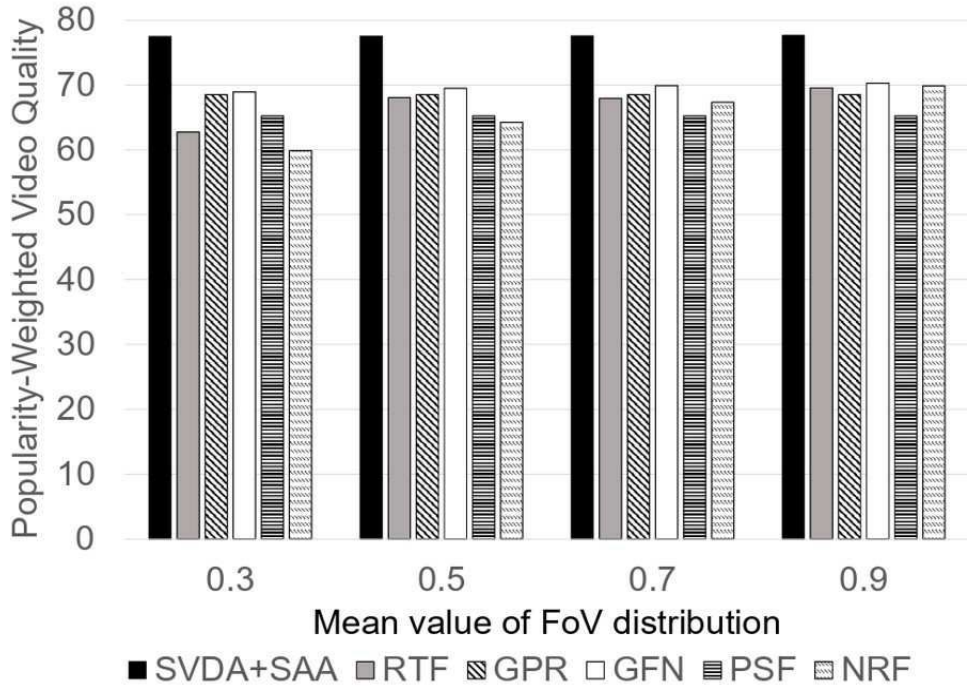
도면8


Popularity-Weighted Video Quality
80
70
60
50
40
30
20
10
0
20%
40%
60%
80%
Rst
SVDA+SAA
RTF
GPR
GFN
PSF
NRF

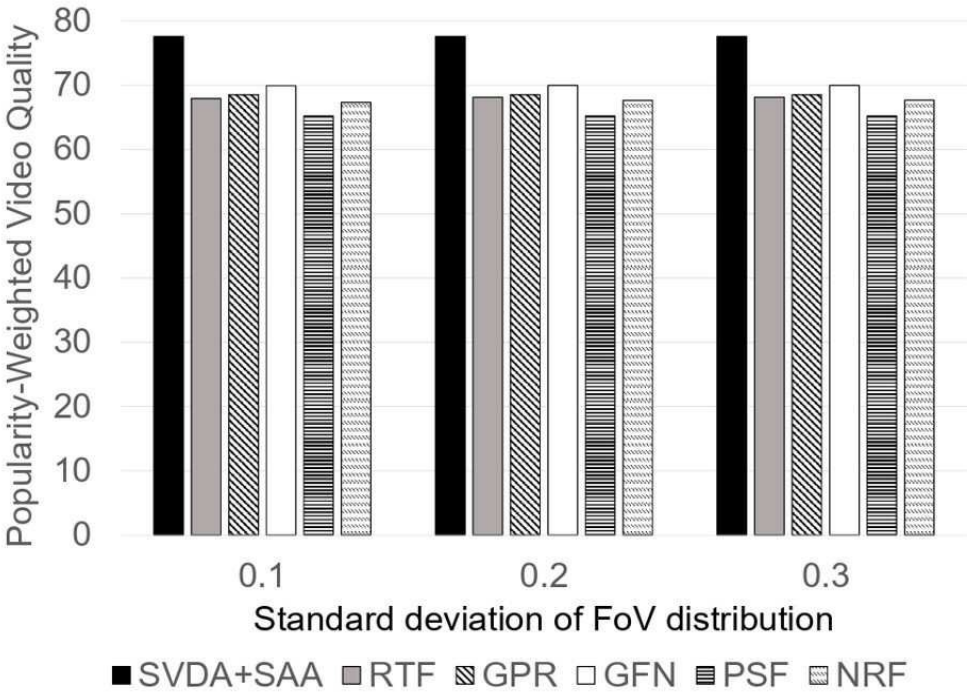
도면9


Popularity-Weighted Video Quality
80
70
60
50
40
30
20
10
0
0.1
0.3
0.5
0.7
0.9
Zipf parameter θ
SVDA+SAA
RTF
GPR
GFN
PSF
NRF

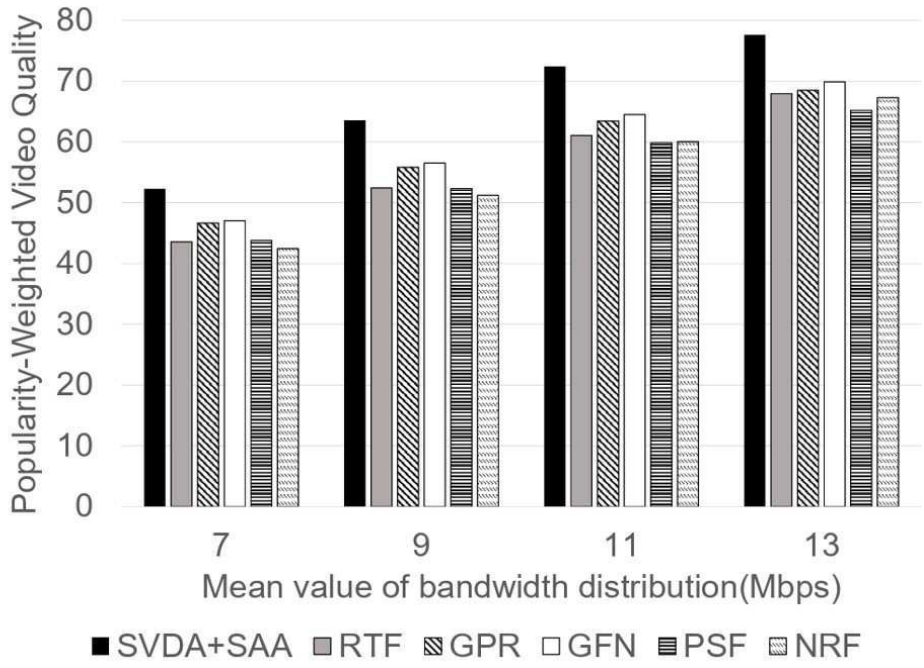
도면10


Popularity-Weighted Video Quality
80
70
60
50
40
30
20
10
0
0.3
0.5
0.7
0.9
Mean value of FoV distribution
SVDA+SAA
RTF
GPR
GFN
PSF
NRF

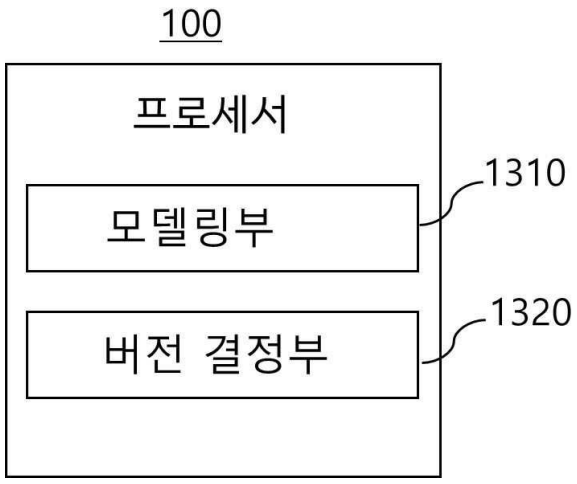
도면11


Popularity-Weighted Video Quality
80
70
60
50
40
30
20
10
0
0.1
0.2
0.3
Standard deviation of FoV distribution
SVDA+SAA
RTF
GPR
GFN
PSF
NRF

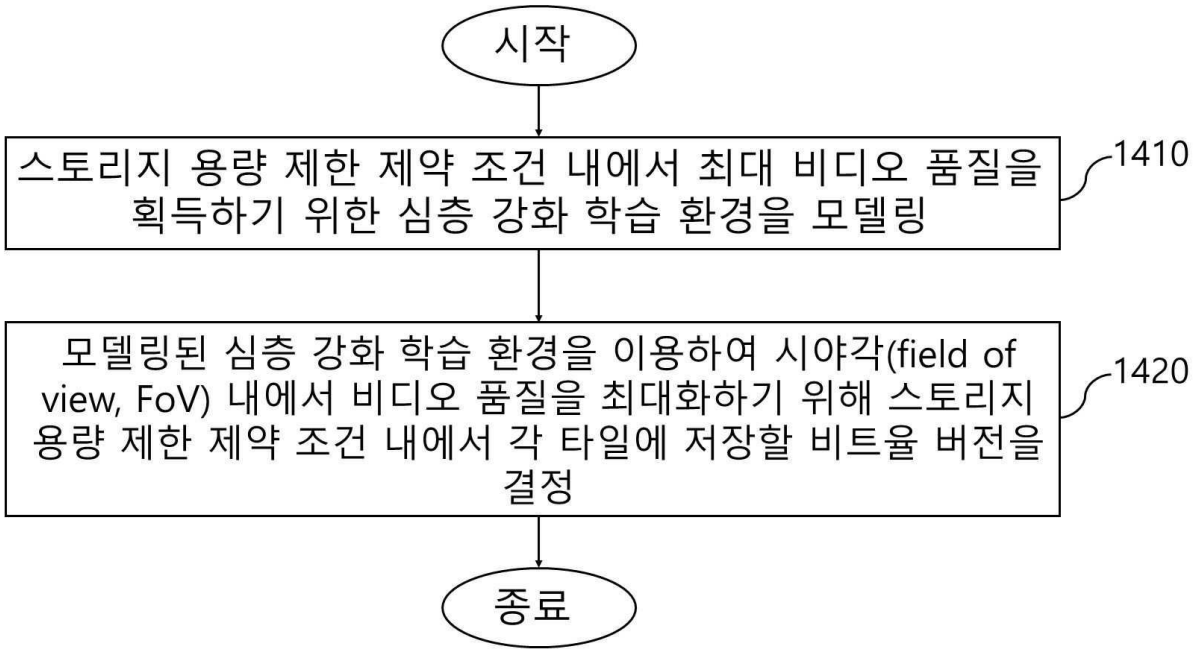
도면12


Popularity-Weighted Video Quality
80
70
60
50
40
30
20
10
0
7
9
11
13
Mean value of bandwidth distribution(Mbps)
SVDA+SAA
RTF
GPR
GFN
PSF
NRF

도면13


100
프로세서
모델링부
1310
버전 결정부
1320

도면14


시작
스토리지 용량 제한 제약 조건 내에서 최대 비디오 품질을 획득하기 위한 심층 강화 학습 환경을 모델링
1410
모델링된 심층 강화 학습 환경을 이용하여 시야각(field of view, FoV) 내에서 비디오 품질을 최대화하기 위해 스토리지 용량 제한 제약 조건 내에서 각 타일에 저장할 비트율 버전을 결정
1420
종료