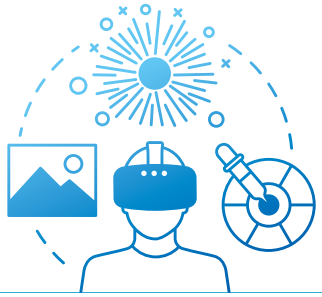
MPEG-I 3DoF+ 비디오 부호화 표준기술

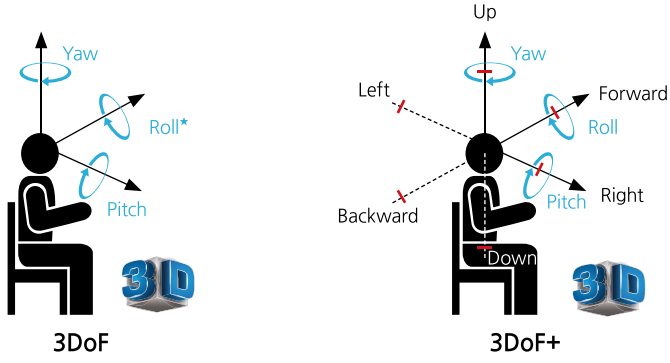


김현호 _ 한국항공대학교 항공전자정보공학과 석사과정
김재곤 _ 한국항공대학교 항공전자정보공학부 교수
이광순 _ ETRI 실감미디어연구실 책임연구원

1. 머리말

최근 가상현실(VR, Virtual Reality)에 대한 수요가 높아짐에 따라, 360도 비디오는 몰입형(immersive) 서비스를 제공하는 새로운 미디어로 인기를 얻고 있다. 이러한 몰입형 미디어를 제공하고 사용자의 몰입도를 더욱 향상시키기 위해 다양한 몰입형 미디어의 제작, 부호화, 전송 및 재현에 대한 기술개발 및 표준화가 활발히 이루어지고 있다. MPEG에서는 2016년 10월 중국 청두에서 열린 116차 회의에서 MPEG-I(MPEG-Immersive)라는 새로운 프로젝트(Coded Representation of Immersive Media, ISO/IEC 23090)를 만들고 몰입형 미디어를 위한 아키텍처, 전송 포맷, 오디오/비디오 압축, 메타데이터 등을 포함한 표준 패키지를 구성하고 이에 대한 표준화 작업을 시작하였다[1][2].

MPEG-I에서는 몰입형 미디어의 몰입도를 사용자 시점의 자유도에 따라서 3DoF(3 Degree of Freedom)에서부터 6DoF까지 단계별로 구분하고 있다. [그림 1]은 3DoF와 3DoF+를 나타낸 것이다. 3DoF는 360도 미디어에 해당하는 것으로 사용자의 위치는 고정되어 있지만 머리를 X, Y, Z의 3축을 중



[그림 1] 몰입형 미디어의 자유도: 3DoF 및 3DoF+

심으로 회전이 가능한 상태로 전방위의 미디어 소비가 가능하다. 이러한 사용자의 머리 회전만을 지원하는 360 비디오는 객체의 가려진 부분은 보지 못함으로써 몰입도가 제한된다. 또한 시청자가 고개를 살짝 기울이기만 해도, 객체에 가려져 영상에 포함되어 있지 않은 부분을 렌더링하기 힘들기 때문에 시각적으로 불편함이 발생할 수 있다. [그림 1]의 3DoF+는 3DoF에 비해 사용자의 X, Y, Z축에 대한 약간의 제한적인 병진 움직임을 허용한다. 즉, 사용자는 앉은 상태에서 머리, 어깨를 움직일 수 있으므로 기존의 전방위 비디오에 추가로 물체에 가려진 부분을 약간 볼 수 있는 움직임 시차(motion parallax)가 제공됨으로써 몰입도가 향상된 미디어 소비가 가능하게 된다. 이는 비디오의 깊이 정보를 이용하여 사용자의 움직임에 따라 해당 위치에서의 비디오를 합성함으로써 제한된 범위에서 임의의 시점에서의 뷰(view)를 재현하는 방식이다.

본고에서는 MPEG-I에서 진행 중인 3DoF+ 비디오 부호화에 대한 표준기술 및 표준화 동향을 소개한다. 2019년 4월 제126차 제네바 MPEG 회의에서 CfP(Call for Proposal) 응답으로 7개 기관(PUT/ETRI, Technicolor/Intel, Philips, Nokia, ZJU)이 CfP에 대응하는 5개의 기술제안서를 제출하였으며, 이들의 비교 검토를 바탕으로 시험모델인 TMIV(Test Model for Immersive Video)와 WD(Working Draft)를 발간하였다. 이후 CE(Core Experiments)를 통한 기술 검증 및 표준기술 채택의 표준화가 진행되고 있다.

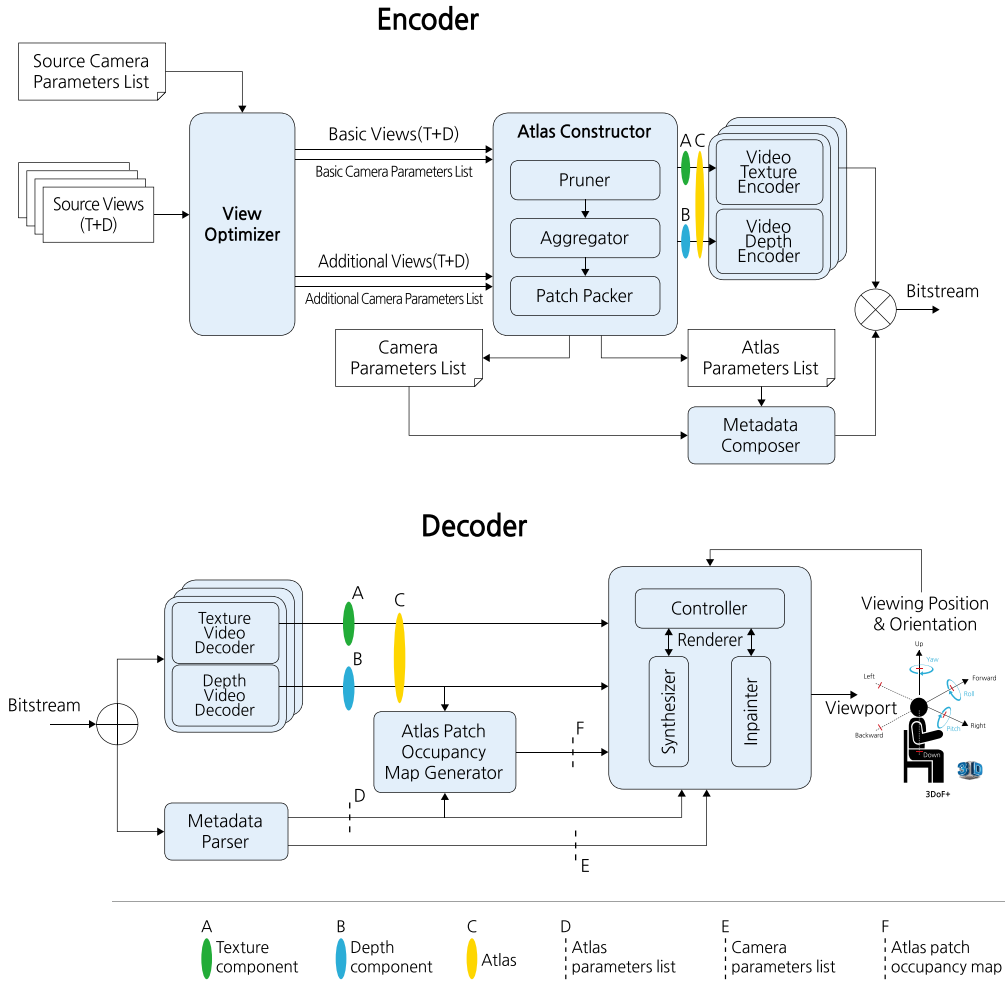
2. 3DoF+ 비디오 부호화

3DoF+ 비디오는 고정된 위치에서의 머리와 어깨의 움직임에 해당하는 제한된 범위에서의 획득 또는 합성한 다수의 뷰 비디오(텍스처(texture) 및 깊이(depth))로 구성된다. 이러한 대용량의 3DoF+ 비디오를 효율적으로 부호화하기 위해서 MPEG-I에서는 기본 코덱으로 기존의 HEVC를 사용하고 압축/전송되는 화소 데이터를 줄이는 전처리와 임의의 뷰를 합성하는 후처리 과정을 포함한다. 현재 MPEG-I에서는 제안기술들을 바탕으로 전처리 및 후처리를 포함한 실험모델인 TMIV를 개발하고 있으며, 본 장에서는 TMIV의 프로세스 흐름, 주요 알고리즘 및 성능 평가 방법에 대해 살펴본다.

2.1 부호화기/복호화기 구성

3DoF+ 비디오 부호화를 위한 전/후처리의 핵심은 제한된 공간에서의 다양한 방향 및 각도로 획득한 다수의 뷰 비디오 간에 존재하는 많은 중복성을 제거함으로써 압축 이전에 미리 압축/전송되는 화소 데이터를 최소화하는 것이다.

[그림 2]는 TMIV의 부호화기/복호화기(Encoder/Decoder) 구성도로 전/후처리 과정을 보여준다. 부호화기에서 다수의 소스 뷰 비디오(텍스처 및 깊이)와 해당 뷰의 카메라 파라미터가 입력된다. 입력된 소스 뷰 비디오는 View Optimizer에서 카메라 파라미터 정보를 기반으로 입력 비디오를 그대로 부호화할 기준 뷰(Basic View)와 기준 뷰와 중복되는 영역이 제거될 부가 뷰(Additional View)로 분류된다. 분류된 기준 뷰와 부가 뷰 비디오들은 Atlas Constructor로 입력되며, 각 뷰 간의 중복성을 제거하는 Pruner, 중복성이 제거되고 남은 영역을 일정 시간구간(Intra Period) 만큼 누적시키는 Aggregator, 그리고 이렇게 만들어진 각 영역들을 감싸는 사각형 영역으로 정의되는 각 패치(patch)들을 하나의 프레임에 패킹(packing)하는 Patch packer를 거쳐 최종적으로 부호화기의 입력이 될 아



<그림 2> TMIV의 부호화기 및 복호화기 구성도[4]

틀라스(Atlas)를 생성한다. 또한 해당 아틀라스의 구성 정보를 나타내는 아틀라스 메타데이터 리스트와 카메라 파라미터 리스트가 생성되어 메타데이터로 구성되어 전송되고 이들은 복호화기에서 가상시점을 복원할 때 사용하게 된다.

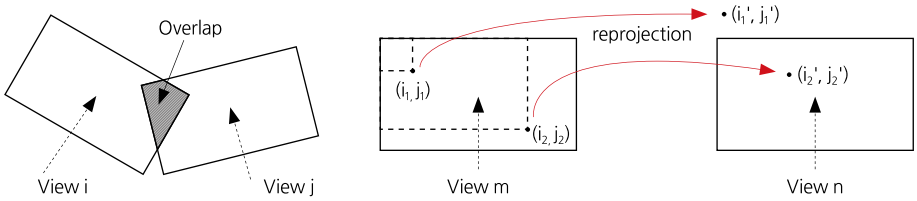
복호화기에서는 압축 전송된 기준 뷰와 아틀라스 비디오를 복호화하고 전송된 메타데이터의 아틀라스 구성 정보를 함께 사용하여 합성기(Renderer)에서 사용자가 원하는 시점의 뷰를 합성한다.

2.2 부호화기(Encoder)

2.2.1 View Optimizer

View Optimizer에서의 기준 뷰와 부가 뷰의 분류 과정은 먼저 몇 개의 기준 뷰가 필요한지를 계산하고 결정된 기준 뷰의 수만큼 기준 뷰를 선택하는 두 단계로 구성된다. 기준 뷰의 수를 결정하는 과정은 다음과 같다.

우선 다수의 입력 소스 뷰 비디오들 중에 뷰 방



[그림 3] View optimizer에서 뷰 간 겹침 정도 계산[4]

향의 각도 차이가 가장 큰 한 쌍의 뷰를 찾는다. 만약 여러 쌍이 찾아지면 쌍을 이루는 두 뷰의 시야각(FOV, Field of view) 합이 가장 큰 쌍을 선택한다. 다수의 쌍이 존재하면 추가로 뷰 간 사이의 거리가 가장 멀리 떨어진 뷰 쌍이 최종 선택된다. 다음으로, 최종 선택된 한 쌍의 뷰가 서로 얼마나 겹쳐지는(overlap)지를 계산한다. [그림 3]은 두 뷰의 겹침 정도를 계산하는 방법으로 한쪽 뷰를 재투영하여 다른 한쪽의 뷰를 합성해 본 뒤, 해당 뷰에서 합성 가능한 화소라면 겹쳐지는 것으로 판단한다. 이렇게 한쪽 뷰의 모든 화소에 대해서 겹침 여부를 확인하고 겹쳐진다고 판단되는 화소의 수가 전체 화소 수의 반 이상이라면 하나의 기준 뷰만이 필요하다고 결정한다. 만약 겹쳐지는 화소의 수가 반 이하라면 겹침 계산을 위해 사용한 한 쌍의 뷰를 포함하여 최소 2개 이상의 기준뷰가 필요하다고 결정된다.

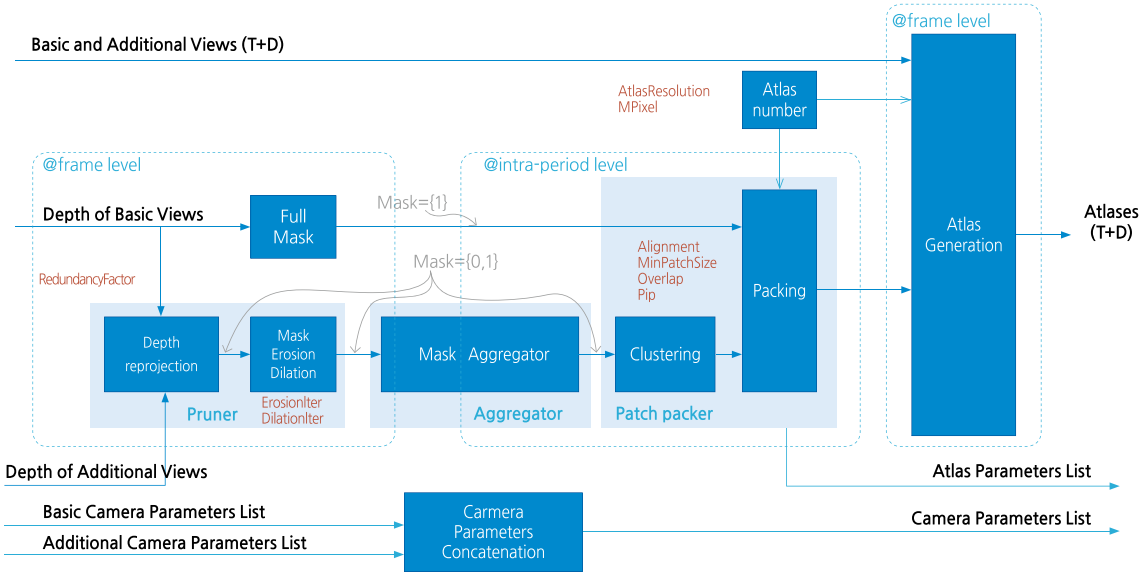
위 과정에서 기준 뷰의 수가 한 개로 결정되면 소스 뷰 비디오 중 모든 소스 뷰 비디오의 카메라 중심 위치와 가장 가까이에 있는 카메라의 뷰가 단일 기준 뷰로 선택된다. 만약 두 개 이상의 뷰가 필요하다고 결정되었다면, 겹침 계산에 사용된 한 쌍의 뷰가 먼저 선택된 뒤, 그 두 뷰와 방향의 각도가 가장 큰 하나의 뷰에 대해서 선택된 두 뷰와의 겹침을 계산하여 겹치는 영역이 반 이하라면 해당 뷰를 기준 뷰로 추가 선택한다. 이러한 과정을 반복하여 기준 뷰 선택을 완료한다.

2.2.2 Atlas Constructor

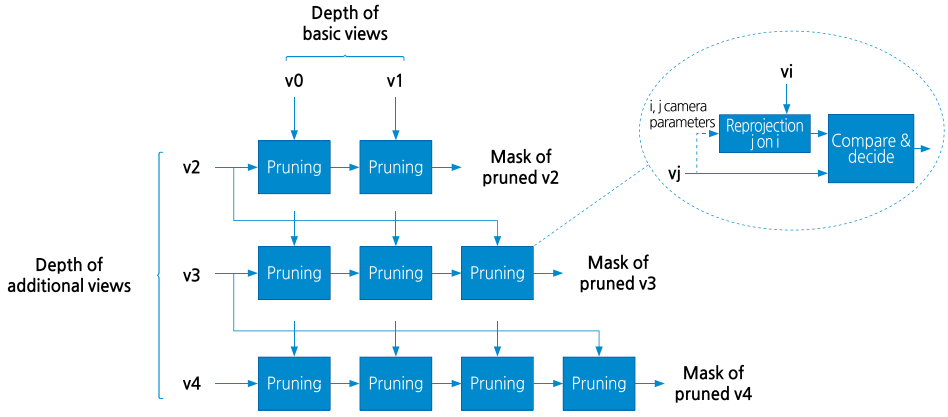
Atlas Constructor는 기준 뷰와 부가 뷰 및 각 뷰에 해당하는 카메라 파라미터를 입력으로 받아 가지치기(Pruning)의 결과로 아틀라스를 생성하고 아틀라스 구성정보 리스트 및 카메라 파라미터 리스트를 출력한다. Pruning 과정을 통해 부가 뷰들은 기준 뷰에 포함된 중복적인 영역은 제거된 뒤 남은 영역을 모아 아틀라스를 구성하게 되며, 기준뷰는 그 자체로 아틀라스로 구성되어 출력된다. [그림 4]와 같이 Atlas Constructor는 크게 Pruner, Aggregator, Patch Packer로 구성된다. 각 구성요소에서의 Pruning, Aggregation, Clustering의 처리는 깊이 정보만으로 진행되며 그 결과를 바탕으로 텍스처와 깊이에 대한 아틀라스가 생성된다. Pruning은 프레임 단위로, Aggregation과 Clustering은 일정 시간 구간(Intra Period) 단위로 진행이 되고 이를 바탕으로 최종 아틀라스는 프레임 단위로 생성이 된다.

1) Pruner

Pruner에서는 입력되는 각 부가 뷰를 Pruning하고 그 결과를 저장하는 이진 마스크(mask)를 생성한다. 이 마스크들은 각 뷰와 동일한 해상도로 '1' 값은 깊이영상의 해당 위치에서의 값이 유효한 값임을 나타내고 '0'은 기준 뷰와 중복되므로 제거되야 할 값임을 의미한다. 즉, [그림 5]에서 각 부가 뷰는 자신의 깊이영상을 재투영하여 기준 뷰의 깊이영상을 합



[그림 4] Atlas Constructor 구성도[4]

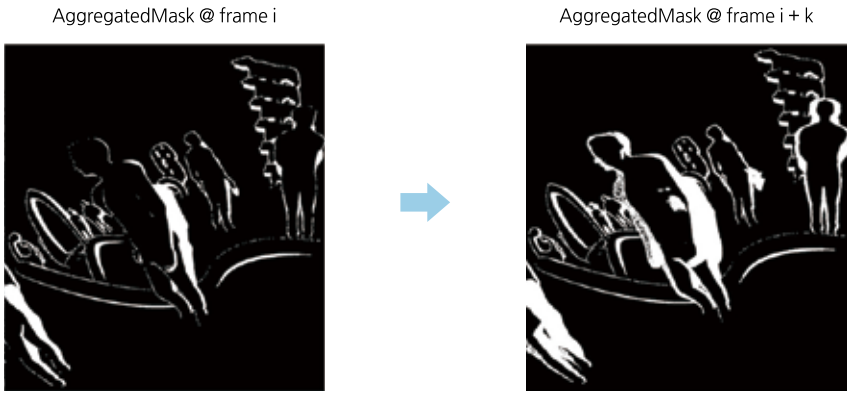


[그림 5] Pruning의 예(2 기준 뷰와 3 부가 뷰에서의 마스크 업데이트)[4]

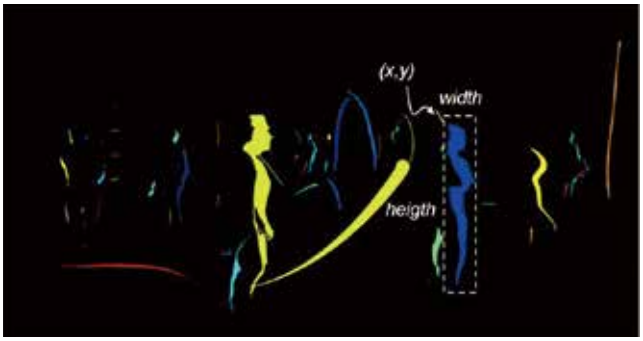
성해 보고 기준 뷰와 중복되는 화소의 위치는 0으로 나머지 부분은 1로 채워진다. 또한 동일한 과정으로 Pruning 처리가 완료된 부가 뷰와도 중복성 처리를 하여 최종 결과로 마스크를 업데이트 한다. 결과적으로, 부가 뷰의 마스크에는 중복되지 않은 최소한의 필요 화소들만이 남게 되고, 기본적인 모폴로지(morphology) 연산을 적용하여 합성 오류로 인한 구멍(hole)이나 잡음을 제거한다.

2) Aggregator

Aggregator에서는 Pruner에서 프레임 단위로 만들어진 각 뷰의 마스크를 입력으로 받아 이들을 일정 시간구간(Intra Period)으로 누적시킨다. 이는 이후 작업을 단순화하고 최종 아틀라스의 구성 정보를



[그림 6] 마스크 누적의 예시[4]



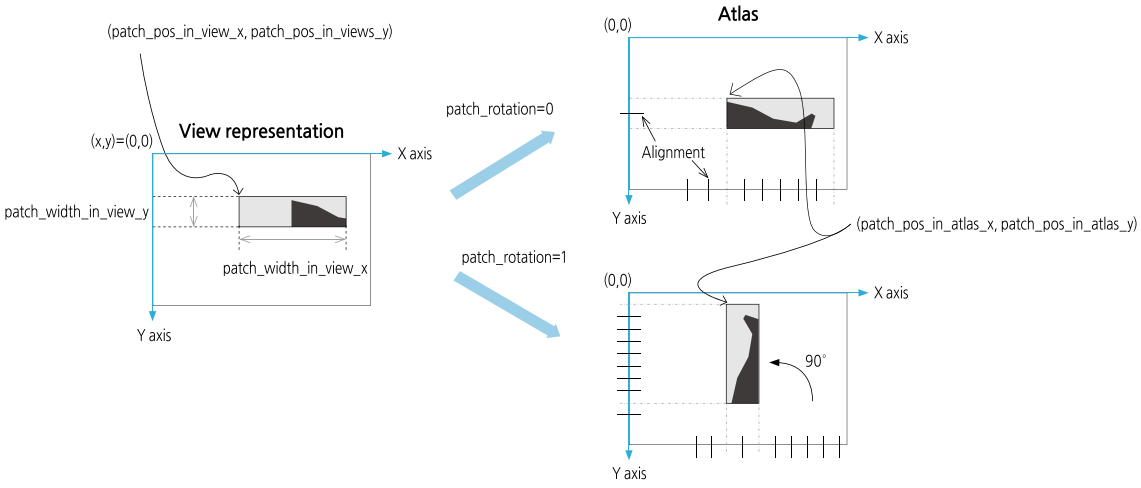
[그림 7] 임의 뷰의 Pruning 결과 생성된 패치 예[4]

줄이는 효과를 가져온다. [그림 6]은 한 뷰에서의 마스크가 일정 시간구간 누적된 결과를 보여준다.

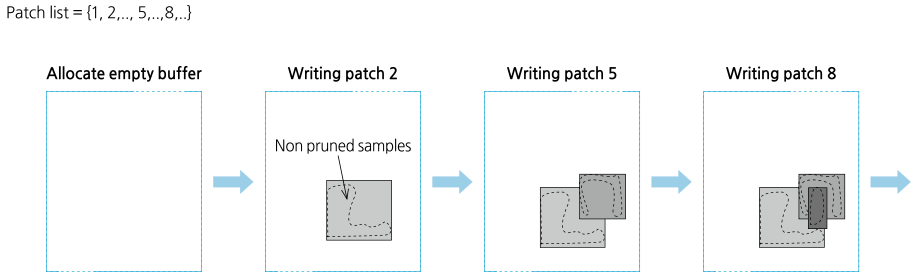
3) Patch Packer

Patch packer에서는 누적된 마스크를 사용하여 마스크 내에서 '1' 값을 가지는 화소의 영역을 포함하는 사각형 박스인 패치(patch)를 생성한다. 이를 위해서 먼저 '1' 값을 가지는 화소 영역을 추출하는 Clustering 과정을 거친다. 즉, '1' 주변 8화소를 탐색하여 '1'의 값을 가지는 동일한 영역에 포함하는 영역 확장(region growing) 방식을 사용한다. 이렇게 추출된 각 영역을 포함하는 사각형 패치는 좌상단 화소 위치와 너비, 높이 값을 파라미터로 가지며, 최종적으로 각 패치 크기의 내림차순으로 정렬되어 저장된다. [그림 7]은 한 뷰에서 패치가 생성된 예시를 보여주고 있으며, 다른 색으로 표시된 영역은 각각 개별적인 패치가 된다. 또한 사각형으로 정의된 패치는 유효영역뿐 만 아니라 마스크에서 '0'을 가지는 유효하지 않은 영역도 포함하고 있다.

이렇게 각 뷰마다 생성된 패치들을 효율적으로 전송하기 위해 아틀라스라는 하나의 프레임에 패킹한다. 이때, 아틀라스의 점유 공간을 최소화하여 압축 효율을 높일 수 있으며, 이를 위해서 패치들은 각자의 유효영역을 침범하지 않는 범위에서 서로 겹쳐질



[그림 8] 아틀라스에 패킹된 패치의 패킹 정보[4]



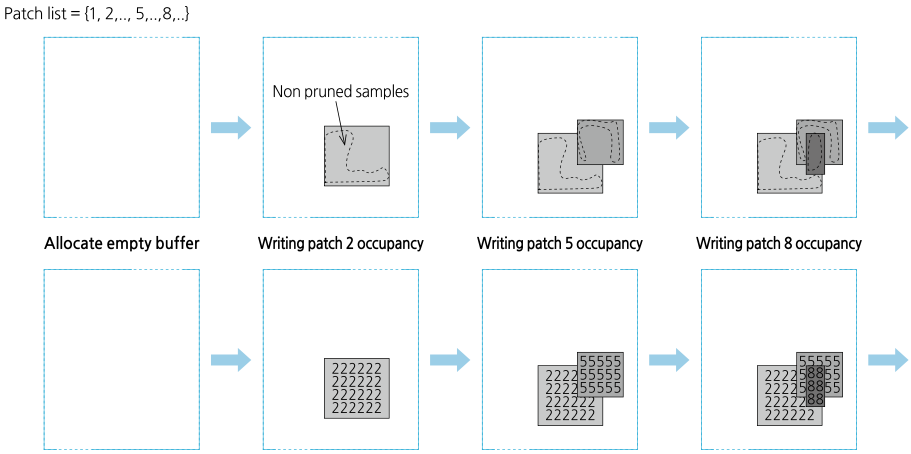
[그림 9] 패치의 아틀라스 배치 예[4]

수 있으며, 반시계방향의 90도 회전까지 허용된다. 각 패치들의 패킹 정보는 Atlas Parameters List로 저장되어 복호화기로 전송이 되어야 이를 토대로 각 소스 뷰를 복원할 수 있게 된다. 패킹 정보는 [그림 8]과 같이 각 패치가 어느 뷰로부터 생성되었는지를 나타내는 뷰의 번호, 해당 뷰에서 패치의 위치정보, 패치의 높이와 너비, 아틀라스에서의 패치 위치정보 그리고 패치의 회전 여부를 포함한다. 이렇게 생성된 패킹 정보를 토대로 후처리기에서는 영상을 복원할 수 있게 된다.

최종적으로, 마스크로만 이루어져 있는 아틀라스에 실제 뷰에서의 텍스처 및 깊이 값을 채워 넣음으로써 일련의 과정이 마무리된다. [그림 9]는 아틀라스에 패치를 순차적으로 채워 넣는 과정을 보여준다.

2.3 복호화기(Decoder)

TMIV 복호화기에서는 먼저 HEVC[5]로 부호화된 아틀라스를 복호화 한다. 복호화된 아틀라스를 사용하여 사용자가 원하는 시점의 뷰를 합성하기 위해 입력된 아틀라스 패킹 정보를 바탕으로 점유맵(Occupancy Map)을 생성한다. 이는 아틀라스에 패킹된 각 패치가 가지고 있는 유효영역만을 시그널링



[그림 10] 아틀라스 점유맵 생성 과정(패킹정보를 통한 아틀라스 내 유효영역 시그널링)[4]

해줌으로써, 이후 가상시점 합성 시 필요한 뷰의 유효화소들을 이용하여 합성을 진행할 수 있게 해주는 역할을 한다. [그림 10]은 점유맵 생성 과정을 나타낸 한 예시이며, 아틀라스 생성 시의 패치가 패킹되는 순서를 그대로 적용하여 패치가 실제 표현하고 있는 영역을 시그널링 해주게 된다.

2.4 성능평가

현재 MPEG-I 그룹에서는 3DoF+ 비디오의 압축 성능평가에 대한 다양한 방법을 사용하고 있다[6]. 먼저 기본적인 360 비디오는 ERP(Equi-Rectangular Projection)으로 표현되며, ERP의 투영 왜곡을 고려한 WS-PSNR(Weighted Sphere PSNR)로 화질을 측정한다. 또한, 주관적 화질평가를 위하여 MS-SSIM(Multi-scale Structural SIMilarity), VMAF(Multimethod Assessment Fusion), VIF(Visual Information Fidelity) 등 다양한 측도를 고려하고 있다.

3. 3DoF+ 비디오 부호화 표준화 계획

2019년 4월 제127차 예테보리 회의에서는 진행된 CE 결과를 검토하고 이를 바탕으로 WD2.0과 TMIV2.0을 발행하였다. CE 검토 결과 및 추가로 제안된 기술들을 기반으로 향후 표준화 진행을 위해 아래와 같이 3개의 CE를 설정하였고[7], 다음 회의에서 그 결과를 바탕으로 WD3.0과 TMIV3.0이 진행될 예정이다.

- **Core Experiments**
  - CE1: View optimization and reprojection (Intel, Interdigital, Nokia)
  - CE2: Pixel pruning(Philips, Nokia, ZJU, ETRI)
  - CE3: Atlas preparation(Interdigital, ZJU, KAU, ETRI, Intel, PUT)

3DoF+ 비디오 부호화 표준화는 CD(2020. 1월), DIS(2020. 7월), FDIS(2020. 10월)의 일정으로 진행될 예정이다[7].

4. 맺음말

몰입형 비디오를 위한 3DoF+ 비디오 부호화는 지난 4월 회의에서 공표된 CfP의 응답 기술제안서 평가를 바탕으로 본격적인 표준화가 시작되었으며 2020년 표준화 완료를 목표로 하고 있다. 국내 기관을 포함하여 세계 유수의 연구기관이 참여하여 비교적 활발한 표준화가 진행되고 있다. 3DoF+ 비디오 부호화는 HEVC를 기본 코덱으로 채택하고 있으며 비디오 부호화 자체보다는 부호화 전단계로 압축 전송될 화소 데이터를 최소화하기 위한 전처리 및 복호화 후의 합성/렌더링의 후처리 및 이를 위한 메타데이터에 대한 표준화에 집중하고 있다.

3DoF, 3DoF+, 6DoF 등 몰입형 비디오의 압축은 그 방대한 데이터량에 따라 핵심 코덱뿐 만 아니라 3DoF+ 압축에서 보듯이 전처리 및 후처리도 전체 압축 성능에 많은 영향을 줄 수 있는 중요한 이슈라고 할 수 있다. 이러한 측면에서 3DoF+ 비디오 부호화 기법은 향후 진행될 6DoF 비디오 부호화의 좋은 예시가 될 것으로 보인다.

[참고문헌]

[1] 'MPEG-I Use Cases for omnidirectional 6DoF, windowed 6DoF, and 6DoF', ISO/IEC JTC1/SC29/WG11, w16768, April 2017.

[2] M. Wien, J. M. Boyce, T. Stockhammer, and W.-H. Peng, 'Standardization Status of Immersive Video Coding', IEEE Jour. Emerg. Select. Topics Circuits Syst., vol. 9, no. 1, pp. 5-17, Mar. 2019.

[3] J. Boyce, R. Dore, V. Vadakital, 'Working Draft2 of ImmersiveVideo', ISO/IEC JTC1/SC29/WG11, w18576, July 2018.

[4] B. Salahieh, B. Kroon, J. Jung, M. Domański (Eds.), 'Test model 2 for Immersive Video', ISO/IEC JTC1/SC29/WG11, w18577, July 2019.

[5] HM reference software, [Online]. Available at: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[6] J. Jung, B. Kroon, J. Boyce, 'Common Test Conditions for Immersive Video', ISO/IEC JTC1/SC29/WG11, N18443, March 2019.

[7] B. Kroon, 'Report of the BOG on Immersive Video', ISO/IEC JTC1/SC29/WG11, m49836, July 2019.