

시점 기반 가상 현실 영상 복호화 시스템을 위한 복호기 적응적 타일 클러스터링 알고리즘

박준호, 정종범, 정세훈, 류은석

성균관대학교 컴퓨터교육과

wnshg2933@skku.edu, uof4949@skku.edu, tpgns97@skku.edu, esryu@skku.edu

Decoder Adaptive Tile Clustering Algorithm for Viewport-Dependent Virtual Reality Video Decoding System

Jun-Ho Park Jong-Beom Jeong Se-Hoon Jeong Eun-Seok Ryu

Department of Computer Education, Sungkyunkwan University (SKKU)

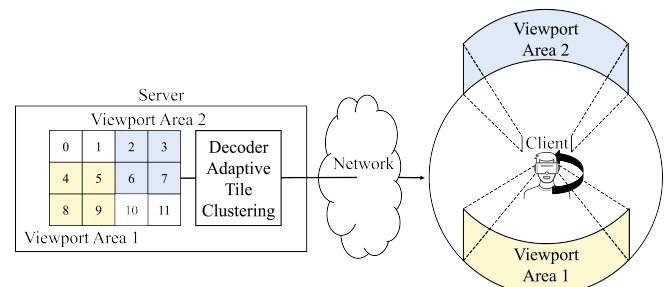
요약

몰입형 고품질 가상 현실 영상 스트리밍을 위한 360도 영상 부호화 및 전송 기술 중 하나로 사용자 시점 기반 타일 스트리밍 기법이 활발히 연구되고 있다. 360도 영상은 용량이 크기 때문에 개별 타일 기반 스트리밍 방법을 사용해 사용자 시점만 보내는 것이 효율적이다. 본 논문은 시점 기반 가상 현실 영상 복호화 시스템을 위한 복호기 적응적 타일 클러스터링 알고리즘을 제안한다. 제안하는 방법은 클라이언트의 복호기가 최대로 복호화 가능한 해상도를 탐색한 후, 사용자 시점 데이터와 복호기 적응적 타일 클러스터링 알고리즘을 이용해 클러스터화할 복수 개의 사용자 시점 타일들의 목록을 생성한 후, 타일 병합기를 이용해 타일들을 병합하여 클러스터 비트스트림을 생성한다. 이후 클라이언트는 병합된 클러스터 비트스트림들을 복호화한 후 사용자 시점을 생성한다. 제안하는 방법을 이용하면 클라이언트의 복호기 환경에 제약받지 않는 복호화가 가능하며, 제안하는 방법 중 하나인 4K_clustering 방법의 경우 8%의 복호화 속도 개선 효과를 얻을 수 있어 몰입형 고품질 가상 현실 영상을 위한 실시간 타일 스트리밍이 가능하다.

1. 서론 및 배경연구

최근 메타버스 관련 콘텐츠들의 고품질 적 요구에 따라 그에 관한 지 연구로써 VR 관련 연구가 활발히 수행되고 있다. 그중에서 특히 통신 기술의 발전에 힘입어, 360도 영상 스트리밍 기술에 관한 연구가 활발히 수행되고 있다. 360도 영상을 시청함에 있어서 사용자가 어지러움 (VR sickness) 을 느끼지 않으려면 360도 영상은 12K 이상의 해상도를 가져야 하며 20ms 이하의 반응지연속도 조건을 만족해야 한다[1]. 하지만 12K의 해상도를 가지는 360도 영상을 그대로 전송하게 되면 대역폭을 너무 크게 차지한다는 문제점과 그 크기 때문에 복호화 속도가 느려 20ms 이하의 반응지연속도 조건을 만족하기 어렵다는 문제점이 있다. 이러한 문제를 해결하고자 타일 스트리밍 기반으로 사용자 시점 타일들을 전송하는 연구가 활발히 진행되고 있다 [2][3]. 타일 스트리밍은 사용자가 시청하고 있는 영역은 고품질로 전송하고 나머지 영역은 저품질 수준으로 전달한다. 타일 스트리밍에서 사용자가 시청하고 있는 영역은 motion-constrained tile set (MCTS) 을 사용하여 압축된 비트스트림에서 추출되어 전송된다[4]. 기존 연구에서 제안하는 사용자 시점 360도 영상 타일 스트리밍 방법에는 두 가지가 있다. 하나는 사용자 시점 타일들을 독립적으로 전송하는 방법이다[5]. 다른 하나는 고해상도 영상(e.g. 16K)에서 사용자 시점 타일만 남겨서 전송하는 방법이다[6]. 전자의 방법 같은 경우엔 수신한 사용자 시점 타일 수 만큼 복호기를 호출하고 복호화를 진행해야 하므로 복호화 시간이 많이 소요된다는 단점이 있다. 후자의 방법 같은 경우엔 클라이언트의 복호기별로 최대로 복호화 가능한 해상도

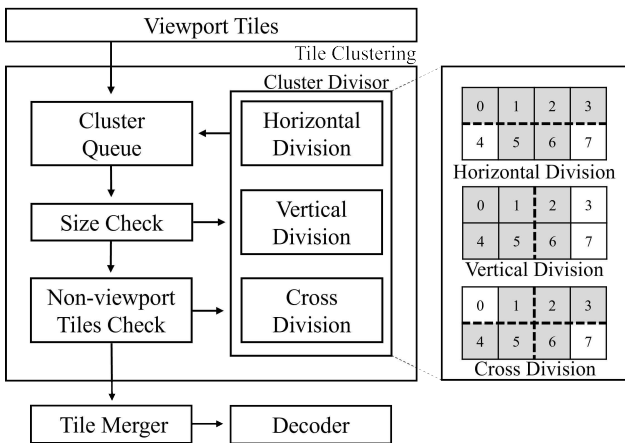
가 상이하기 때문에 클라이언트의 복호기 환경에 따라서 360도 영상의 복호화가 불가능할 수도 있다는 문제점이 있다. 그래서 본 논문은 시점 기반 가상 현실 영상 복호화 시에 클라이언트의 복호기 환경에 맞춘 복호화가 가능한 복호기 적응적 타일 클러스터링 알고리즘을 제안한다. 그림 1은 제안하는 방법의 동작 흐름을 나타낸다. 사용자 시점 데이터를 통해 추출된 고화질의 MCTS 비트스트림에서 복호기 적응적 타일 클러스터링 알고리즘을 이용해 병합된 클러스터 비트스트림을 생성하고 복호기를 이용해 복호화하여 사용자 시점을 생성한다. 본 논문의 구성은 다음과 같다. 2절에선 복호기 적응적 타일 클러스터링 알고리즘과 타일 병합기를 이용한 타일 병합 및 복호화에 대해 소개하고 3절에선 제안하는 방식을 사용했을 때의 복호화 시간과 사용자 시점 타일들을 각각 복호화했을 때의 복호화 시간과 비교하여 실험 및 결과를 제시하며 4절에서 본 논문의 결론을 맺는다.



(그림 1) 제안하는 방법을 이용한 시점 기반 가상 현실 영상 복호화 시스템

2. 복호기 적응적 타일 클러스터링 알고리즘을 이용한 복호화 시스템

본 절은 복호기 적응적 타일 클러스터링 알고리즘과 타일 병합기 그리고 복호기를 이용한 시점 기반 가상 현실 영상 복호화 시스템의 동작 과정에 관해 기술한다. 먼저, 서버는 클라이언트의 복호기가 최대로 복호화 가능한 최대 해상도를 탐색한 후, 사용자 시점 데이터와 복호기 적응적 타일 클러스터링을 이용하여 병합되어야 할 타일들의 목록을 생성한다. 생성된 병합할 타일 목록과 타일 병합기를 이용해 개별 타일 병합을 수행해 병합된 클러스터 비트스트림을 생성한 후, 복호기를 이용해 복호화를 진행한다. 그림2는 해당 시스템의 상세한 구조를 나타낸다.



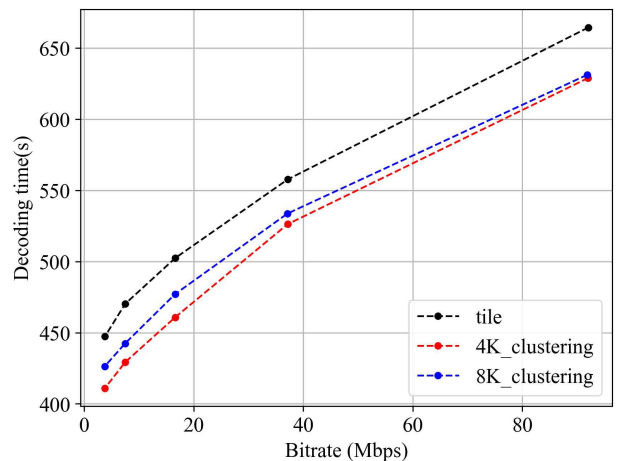
(그림 2) 복호기 적응적 타일 클러스터링 복호화 시스템 구조도

복호기 적응적 타일 클러스터링 알고리즘의 목적은 클라이언트의 복호기 환경과 사용자 시점 데이터를 이용해 병합해야 할 타일들의 목록을 만드는 것에 있다. 복호기 적응적 타일 클러스터링은 다음과 같은 순서로 진행된다. 먼저 클라이언트에 정보를 요청해 클라이언트의 복호기가 최대로 복호화 가능한 해상도 정보를 수신한 후, 해당 정보를 Size Check 모듈에 적용한다. 그 후, 사용자 시점 타일들 (viewport tiles)를 2차원 배열에 기록한 후, breadth first search (BFS)를 통해 인접한 사용자 시점 타일들을 묶어 군집 정보를 추출한다. 그 후, 추출된 군집 정보를 Cluster Queue에 삽입하며, 이후로 Cluster Queue의 모든 원소가 소거될 때까지 타일 클러스터링이 진행된다. Cluster Queue에서 클러스터 정보를 인출 한 뒤, Size Check을 진행한다. Size Check 모듈은 해당 클러스터가 클라이언트가 최대로 복호화 가능한 해상도를 초과하는지 검사하며, 만약 초과한다면 클러스터 분할 모듈로 클러스터 정보가 전달되어 클러스터 분할이 진행된다. 만약 초과하지 않는다면 다음 단계인 Non-viewport Tile Check 모듈로 클러스터 정보를 전달하며, 클러스터 내에 사용자 시점 외 타일이 존재하는지 검사한다. 만약 사용자 시점 외 타일이 클러스터 내에 존재한다면 클러스터 분할 모듈로 클러스터 정보가 전달되어 클러스터 분할이 진행되고, 사용자 시점 외 타일이 포함되어 있지 않다면, 타일 병합기로 클러스터 정보가 전달되어 타일 병합이 진행된다. 클러스터 분할은 총 3가지 분할 기법 중 하나를 수행하게 되는데, 각각 수평 분할(horizontal division), 수직 분할(vertical division), 십자 분할(cross division)이다. 수평 분할은 클러스터를 가로로 절반을 분할 하며, 수직 분할은 클러스터를 세로로 절반을 분할 하고, 십자 분할은 클러스터를 십자 모양으로 절반을 분할 한다.

만약 클러스터의 가로 길이가 세로 길이 보다 더 길다면 수직 분할, 수평 분할, 십자 분할 순으로 분할이 진행되고 세로 길이가 가로 길이보다 길다면 수평 분할, 수직 분할, 십자 분할 순으로 분할이 진행된다. 분할된 클러스터 중 한 클러스터라도 사용자 시점 외 타일이 없다면 클러스터 분할을 더 이상 진행하지 않고 분할된 클러스터 정보를 Cluster Queue에 삽입한다. 해당 과정을 Cluster Queue의 원소가 모두 소거될 때까지 진행하면, 타일 병합기에는 최종적으로 병합해야 할 타일들의 정보가 쌓이게 된다. 타일 병합기는 해당 정보를 이용해 타일 병합을 수행하여 클러스터 비트스트림을 생성한 후, 생성된 클러스터 비트스트림들을 복호기가 전달받아 복호화를 수행하게 된다.

3. 실험 및 결과

본 절은 제안한 복호기 적응적 타일 클러스터링 알고리즘을 이용한 시점 기반 가상 현실 영상 복호화 실험 결과에 대해 기술한다. 먼저 클라이언트가 복호화 가능한 최대 크기는 4K와 8K로 설정하였다. 4K는 보통의 모바일 환경에서 복호화 가능한 최대 크기이다[7]. 8K는 NVENC 기반 데스크톱 클라이언트 환경에서 복호화 가능한 최대 크기이다[8]. 실험 환경은 운영체제 Windows 10, CPU i7-8700K, RAM 32GB, 복호기 HM 16.20[9]이다. 실험에 사용된 테스트 시퀀스는 Drone과 LakePark이고 두 테스트 시퀀스 모두 16K의 해상도와 300 프레임을 가진다. 두 테스트 시퀀스의 부호화에 사용된 양자화 매개변수 (quantization parameter, QP) 값은 22, 27, 32, 37, 42이며 각 영상은 사용자 시점 32프레임 단위로 분할 되어 부호화되었다. 대조군은 사용자 시점 타일들을 병합하지 않고 각각 복호화했을 때로, 즉 개별 타일 기반 방법을 사용하였을 때로 설정하였고, 각 타일의 크기는 1280 x 1280의 해상도를 가진다. 실험군은 제안하는 방법을 적용하여 복호화했을 때의 시간으로 설정하였다. 실험군에 사용되는 클러스터들을 생성할때 사용되는 타일 병합기는 고속 타일 병합기[10]를 사용하였다. 표 1은 실험 결과이며, 그림 3, 4는 각각 Drone과 LakePark 시퀀스에서의 비트레이트별 복호화 속도 그래프이다. 그림 5는 두 테스트 시퀀스에서의 비트레이트별 복호화 속도 개선 그래프이다.

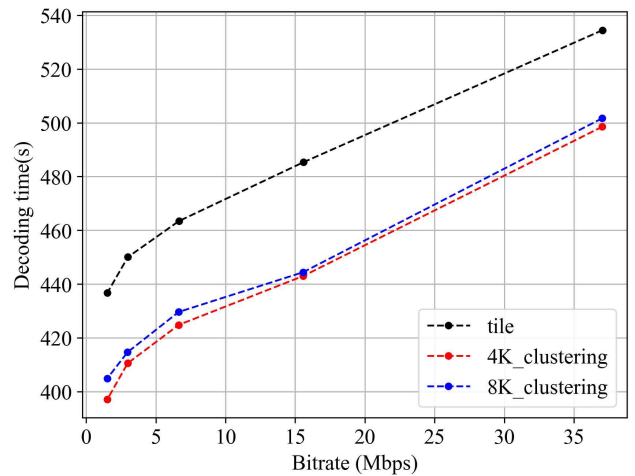


(그림 3) Drone 시퀀스에서의 비트레이트별 복호화 속도 (tile은 기존 방법인 개별 타일 기반 복호화를 수행한 결과. 4K_clustering, 8K_clustering은 클라이언트가 최대로 복호화 가능한 크기로 4K, 8K 해상도로 설정한 후, 제안하는 방법을 이용해 복호화를 수행한 결과.)

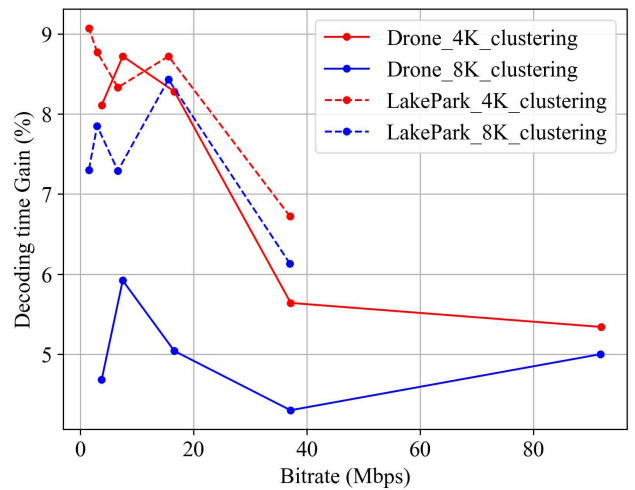
(표 1) 개별 타일 대비 타일 클러스터 기반 복호화 실험 결과

sequence	method	QP	bitrate(Mbps)	time(s)	gain(%)
Drone	tile	22	92.01	664.32	
		27	37.16	557.64	
		32	16.62	502.38	
		37	7.51	470.29	
		42	3.78	447.27	
	4K	22	92.00	628.87	5.34%
		27	37.15	526.19	5.64%
		32	16.61	460.76	8.29%
		37	7.50	429.29	8.72%
		42	3.78	410.99	8.11%
	8K	22	91.84	631.13	5.74%
		27	37.12	533.66	5.19%
		32	16.60	477.05	5.37%
		37	7.49	442.47	4.77%
		42	3.77	426.36	4.29%
LakePark	tile	22	37.04	534.41	
		27	15.58	485.30	
		32	6.65	463.43	
		37	2.99	450.04	
		42	1.52	436.79	
	4K	22	37.03	498.51	6.72%
		27	15.57	443.00	8.72%
		32	6.64	424.83	8.33%
		37	2.98	410.58	8.77%
		42	1.52	397.18	9.07%
	8K	22	37.02	501.64	6.88%
		27	15.56	444.41	8.75%
		32	6.64	429.65	8.11%
		37	2.97	414.69	6.96%
		42	1.51	404.92	6.26%

QP 22로 부호화된 영상을 복호화했을 때, 즉 그림 3, 4에서 두 테스트 시퀀스의 비트레이트가 가장 높은 지점을 보게 되면, 두 테스트 시퀀스 모두 16K의 해상도를 가짐에도 Drone의 비트레이트가 LakePark에 비해 두 배 이상 높음을 확인할 수 있다. 이를 통해 Drone이 LakePark보다 영상 복잡도가 높음을 알 수 있다. 그림 3, 4를 통해 제시하는 방법인 4K_clustering 방법과 8K_clustering 방법으로 복호화를 수행한 시간이 기존의 방법인 개별 타일(tile) 기반 방법보다 더 빠른 것을 확인할 수 있으며 모든 경우에서 4K_clustering 방법이 8K_clustering 방법보다 빠름을 알 수 있다. 그림 5를 보게 되면, 테스트 시퀀스 별로 개선비 증감 양상이 비슷함을 확인할 수 있다. 여기서, LakePark 시퀀스를 보게 되면, 비트레이트와 상관없이 두 방법 간의 개선비 차이가 어느 정도 균등하며, 그 크기가 그렇게 크지 않음을 확인할 수 있다. 하지만 Drone 시퀀스를 보게 되면 비트레이트가 커질수록 두 방법 간 개선비 차이가 점점 작아지고 비트레이트가 작은 경우 4K_clustering 방법의 개선비가 8K_clustering 방법 보다 약 두 배 높음을 확인할 수 있다. 실험 결과를 통해 모든 경우에서 4K_clustering 방법이 8K_clustering 방법보다 빠름을 확인할 수 있었고 특히 Drone과 같이 영상 복잡도가 높은 영상에서 낮은 비트레이트로 복호화를 수행



(그림 4) LakePark 시퀀스에서의 비트레이트별 복호화 속도



(그림 5) 비트레이트별 복호화 속도 개선비

할 때 4K_clustering 방법과 8K_clustering 방법간 복호화 속도 개선비 차이가 약 두 배 가까이 차이 남을 확인할 수 있다.

실험 결과를 보게 되면, 모든 경우에서 4K_clustering 방법이 개별 타일 기반 방법과 8K_clustering 방법보다 빠르다. 즉, 4K_clustering 방법이 개별 타일 기반 방법보다 빠르니 복호화를 수행하는 각 영상의 크기가 작을수록 복호화 속도가 빨라지는 것이 아님을 확인할 수 있다. 또한, 클라이언트가 복호화 가능한 최대 해상도가 커질수록 제안하는 방법을 이용해 생성한 클러스터의 수는 적어진다. 즉, 4K_clustering 방법이 8K_clustering 방법보다 빠르니 생성된 클러스터의 수가 적을수록, 즉 복호기를 호출하는 횟수가 적을수록 복호화 속도가 빨라지는 것이 아님을 확인할 수 있다. 그렇기에, 개별 타일 기반 방법과 8K_clustering 방법 사이에 복호기의 호출 횟수도 적당히 적으면서 각 영상의 크기도 적당히 작아 가장 빠른 복호화 속도를 낼 수 있는 최적의 분할 지점이 있을 것으로 생각된다. 해당 실험에선 4K_clustering 방법이 그 지점에 가장 가까우며 추후 실험으로 최적의 분할 방법에 관해 연구가 필요할 것으로 생각된다.

4. 결론

본 논문은 시점 기반 가상 현실 영상 복호화 시스템을 위한 복호기 적응적 타일 클러스터링 알고리즘을 제안한다. 제안하는 방법은, 사용자 시점 데이터를 클라이언트의 복호기 환경과 복호기 적응적 타일 클러스터링 알고리즘을 이용하여 병합해야 할 타일들의 목록을 생성한 후, 타일 병합기를 이용해 병합된 클러스터 비트스트림을 생성한다. 타일 기반의 스트리밍 방식은 실제 사용자가 보고 있는 데이터만 전송하며, 9개의 타일이 있는 경우, 1개의 타일만을 전송할 때는 최소 75%에서 97%, 4개의 타일을 전송할 때는 47%만큼의 대역폭을 절약할 수 있는 것으로 알려져 있다[11]. QP 22로 부호화된 원본 *Drone* 시퀀스의 비트레이트는 219.76Mbps, *LakePark*는 92.88Mbps이고 해당 실험에서 측정된 *Drone* 시퀀스의 비트레이트는 92.01Mbps, *LakePark*시퀀스는 37.04Mbps이다. 즉, 제안하는 방법을 적용하면 *Drone* 시퀀스는 58%, *LakePark*는 60%의 대역폭을 절약할 수 있다. 또한, 제안하는 방법을 사용하게 되면 클라이언트 복호기 환경에 제약받지 않는 복호화가 가능하며, 복호화 속도는 기존의 개별 타일 기반 방법과 비교하여 4K_clustering 방법의 경우 평균적으로 약 8%, 8K_clustering 방법의 경우 약 5% 개선을 얻을 수 있다. 추후 연구로는 최적의 클러스터 분할 방법에 관해 연구할 계획이다.

ACKNOWLEDGEMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00920-002, (세부 2) 중대형 공간용 초고해상도 비정형 플레옵틱 영상 저장/압축/전송 기술 개발)

참고문헌

- [1] Mary-Luc Champel, Thomas Stockhammer, Thierry Fautier, Emmanuel Thomas, Rob Koenen. 2016. Quality Requirements for VR. 116th MPEG meeting of ISO/IEC JTC1/SC29/ WG11, MPEG 116/m39532
- [2] J. W. Son, D. M. Jang and E. S. Ryu, "Implementing 360 video tiled streaming system," in Proc. of the 9th ACM Multimedia Systems Conf. ACM, Amsterdam, Netherlands, pp. 521-524, 2018.
- [3] J. W. Son and E. S. Ryu, "Tile-based 360-degree video streaming for mobile virtual reality in cyber physical system", Computers & Electrical Engineering, Volume 72, 2018, Pages 361-368, ISSN 0045-7906,
- [4] R. Skupin, Y. Sanchez, K. Sühring, T. Schierl, E. S. Ryu et al., "Temporal MCTS coding constraints implementation," in 122th MPEG Meeting of ISO/IEC JTC1/SC29/ WG11, San Diego, CA, USA, m42423, 2018.
- [5] Jeong, J. B., Lee, S., Kim, I., & Ryu, E. S. (2021). Performance Analysis of Viewport-dependent Tiled Streaming on 16K Ultra High-quality 360-degree Video. Journal of Internet Computing and Services, 22(3), 1-8. Son, J., Jang, D., & Ryu, E. S. (2018, June). Implementing 360 video tiled streaming system. In Proceedings of the 9th ACM Multimedia Systems Conference (pp. 521-524).
- [6] Jeong, J. B., Lee, S., Kim, I., & Ryu, E. S. (2021, January). Implementing Viewport Tile Extractor for Viewport-Adaptive 360-Degree Video Tiled Streaming. In 2021 International Conference on Information Networking (ICOIN) (pp. 8-12). IEEE.
- [7] Growing support of HEVC or H.265 video on mobile devices. (2018, August 23). Retrieved October 30, 2021, from ScientiaMobile : <https://www.scientiamobile.com/growing-support-of-hevc-or-h-265-video-on-mobile-devices/>
- [8] NVDEC Video Decoder API Programming Guide. (2021, October 26). Retrieved October 30 2021, from nvidia : <https://docs.nvidia.com/video-technologies/video-codec-sdk/nvdec-video-decoder-api-prog-guide/>
- [9] HM-16.20. (2018, September 14). Retrieved October 21 2021, from Video Coding Standardization GitLab : <https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tree/HM-16.20>
- [10] Jong-Beom Jeong, Soonbin Lee, Inae Kim, Eun-Seok Ryu, "Implementing Fast Tile Merger for Viewport-dependent 360-degree Video Streaming", Korean Society for Internet Information (KSII) Spring Conference, Apr. 23-24, 2021
- [11] J. W. Son, D. M. Jang, E. S. Ryu, 2018. Implementing Motion-Constrained Tile and Viewport Extraction for VR Streaming. In Proceedings of the 28th ACM Network and Operating System Support for Digital Audio and Video, pp. 61-66.