정규논문 (Regular Paper)
방송공학회논문지 제19권 제5호, 2014년 9월 (JBE Vol. 19, No. 5, September 2014)
http://dx.doi.org/10.5909/JBE.2014.19.5.713
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

3D-HEVC 비디오 부호화 성능 분석

박 대 민[a)], 최 해 철[a)‡]

Performance Analysis of 3D-HEVC Video Coding

Daemin Park[a)], and Haechul Choi[a)‡]

요 약

차세대 영상 기술의 하나로 다양한 분야에서 주목받고 있는 고품질 다시점 및 3차원 콘텐츠들에 대한 비디오 부호화 기술의 연구 및 표준화가 활발히 진행되고 있다. 다시점 및 3차원 비디오 기술은 복수의 시점을 이용하여 사용자에게 실감나는 영상을 제공할 수 있다. 하지만, 많은 시점을 획득 및 전송하는 것은 한계가 있으므로, 소수의 시점을 이용하여 다양한 시점을 제공하는 연구가 주를 이룬다. 이러한 연구에는 전송 시점을 줄이는 대신 깊이 정보를 전송하여 전송된 시점으로부터 더욱 정확히 임의 시점을 생성하는 기술과 시점간 정보 중복성을 제거하기 위한 부호화 기술이 있다. 최근 국제 표준화 기구인 JCT-3V(Joint Collaborative Team on 3D Video Coding Extension Development)에서는 다시점 및 3차원 비디오 영상을 효율적으로 부호화할 수 있는 기술에 대하여 표준화가 진행되고 있다. 본 논문은 현재 JCT-3V에서 HEVC(High Efficiency Video Coding) 기반으로 표준화가 진행 중인 3D-HEVC 부호화 기술에 대해 살펴보고 그 부호화 및 복잡도 성능을 분석하였다. 이러한 성능 분석은 향후 부호화 성능 향상을 위한 알고리즘 개발 및 고속 부호화기 개발을 위한 부호화 툴의 선별 및 조정에 유용할 것으로 판단된다.

Abstract

Multi-view and 3D video technologies for a next generation video service are widely studied. These technologies can make users feel realistic experience as supporting various views. Because acquisition and transmission of a large number of views require a high cost, main challenges for multi-view and 3D video include view synthesis, video coding, and depth coding. Recently, JCT-3V (joint collaborative team on 3D video coding extension development) has being developed a new standard for multi-view and 3D video. In this paper, major tools adopted in this standard are introduced and evaluated in terms of coding efficiency and complexity. This performance analysis would be helpful for the development of a fast 3D video encoder as well as a new 3D video coding algorithm.

Keyword : 3D video, Multi-view video, video coding, performance analysis, HEVC

a) 한밭대학교 정보통신전문대학원 멀티미디어공학과 (Multimedia Engineering, Hanbat National University)
‡ Corresponding Author : 최해철(Haechul Choi)
E-mail: choihc@hanbat.ac.kr
Tel: +82-42-821-1149
※본 연구는 교육부와 한국연구재단의 지역혁신인력양성사업(2012H1B8A2025982) 및 미래창조과학부와 한국연구재단의 신진연구지원사업(NRF-2013R1A1A1010344)으로 수행된 연구결과임.
· Manuscript received July 22, 2014 Revised August 18, 2014 Accepted August 18, 2014

Ⅰ. 서 론

최근 카메라 및 디스플레이 기기의 혁신적인 발전으로 실감나는 영상에 대한 소비자의 요구가 증가하고 있다. 여러 대의 카메라를 이용하여 복수의 시점을 획득하고 이를 이용하여 획득된 시점 뿐 아니라 다양한 시점을 합성하여 제공하는 3DV(3D Video) 기술은 현실과 흡사한 환경을 만들어 사용자에게 실제 공간에 있는 것처럼 느껴지는 실감 영상을 제공한다[1]. 이와 같은 3차원 비디오 기술의 발전으로 제공되는 서비스들은 기존의 HD(High Definition) 화질보다 2배 이상의 화질을 지원하도록 하고 2D 영상보다 실감나는 느낌을 제공하는데 목적을 두고 있다[2]. 과거의 3D 영상기술은 단지 두 개의 시점을 이용한 스테레오스코픽(Stereoscopic) 영상이었으며, 최근 기술은 다양한 카메라 시점을 이용하여 영상을 획득하고 사용자가 원하는 시점에 대한 영상을 합성하거나, 다양한 시점을 이용하여 더욱 실감나고 편안한 입체영상을 제공할 수 있다. 하지만, 많은 수의 시점을 카메라를 이용하여 촬영하는 것은 한계가 있고 전송 측면에서도 많은 데이터양을 요구하기 때문에 영상의 깊이정보를 이용하여 실제 카메라의 수보다 많은 시점을 더욱 정확히 생성하고, 전송되는 시점의 수를 줄이는 방법을 이용한다[3].

상기 기술들에 대한 표준화를 위하여 국제 표준화 단체인 ISO/IEC JTC1/SC29/WG11 MPEG(Mov-ing Picture Experts Group)에서는 3DV에 대한 표준안의 요구사항을 정의하였다. 이는 데이터 포맷 요구사항, 압축 요구사항, 보간 요구사항 등을 포함한다. 첫 번째 데이터 포맷 요구사항으로 데이터에는 한 시점 이상의 데이터를 포함해야 하며, 다른 시점을 만들어 낼 수 있는 데이터와 카메라 파라미터에 대한 데이터를 포함해야 한다. 두 번째로 압축요구 사항에는 3DV 압축 데이터 포맷이 합성된 영상의 정확도에 비례하여 비트율을 조정할 수 있어야 하고, 기존에 사용된 부호화 표준과 역호환성을 유지할 수 있도록 스테레오 및 비트스트림을 모두 추출할 수 있어야 한다. 마지막 보간 요구사항으로 데이터 포맷에는 실시간으로 보간을 지원하기 때문에 복잡도가 낮아야하고, 3D 장면이 스크린에 따라 전면 및 후면에서 인식되어야 하므로 디스플레이에 맞는 깊이위치를 이동할 수 있어야 하며, 실시간 처리를 위한 스케일러블 렌더링 기능을 지원해야 한다[4]. 상기 요구사항이 제정된 이후로 3차원 비디오 부호화에 대한 제안요청서(Call For Proposal)[5]가 2011년 4월4일 96차 MPEG 회의에서 배포되었으며, 2011년 10월부터 JCT-3V(Joint Collaborative Team on 3D Video Coding Extension Development)에서 3차원 부호화 기술들에 대한 표준화가 본격적으로 진행되고 있다.

표 1. 3차원 비디오 부호화 표준화 일정[5]
Table 1. Standardization schedule of JCT-3V[5]

일정	내용
2011년 01월	제안 요청서 초안 배포
2011년 04월	최종 요청서 배포 및 기술 등록
2012년 04월	작업 초안(Working Draft) 작성
2014년 07월	MVC compatible extension including depth 표준안 제정
2014년 10월	AVC compatible video-plus-depth extension 표준안 제정
2015년 06월	HEVC 3D extensions 표준안 제정

상기 표 1은 3차원 비디오 부호화 표준화 작업 일정을 보여준다. ‘MVC compatible extension including depth’는 H.264/AVC multi-view video coding (MVC)의 확장판으로 2014년 7월에 최종 표준안이 제정될 예정이다. 이 표준안은 부호화시 블록 단위의 구문(syntax) 변경 없이 3차원 비디오 부호화기에서 깊이 영상을 부호화하기 위한 툴이 포함되었다. 또한 AVC(Advanced Video Coding)와 호환하는 3D 비디오 부호화의 표준은 2014년 10월에 ‘AVC compatible video-plus-depth extension’이라는 표준안[6]이 제정될 예정이다. 마지막으로 가장 최근의 2D 비디오 부호화 표준인 HEVC(High Efficiency Video Coding)와 호환하는 3D 비디오 부호화 표준은 2015년 6월까지 ‘HEVC 3D extension’이라는 이름으로 제정될 예정이다[3][6]. 이 HEVC 기반의 3D 비디오 부호화 표준화를 3D-HEVC로 명명하였다. 현재 JCT-3V에서는 3D-HEVC 비디오 부호화 알고리즘 개발을 위해 경쟁을 통한 표준화가 진행 중이다. 이러한 툴을 더 효율적으로 개발하기 위해서는 현재 표준으로 채택되어 있는 툴의 알고리즘을 정확히 파악하고, 전체적인

성능 및 개별 툴의 성능을 비교하고 분석하여야한다. 이에 따라, 본 논문은 현재 표준화에 채택된 부호화 툴에 대한 종합적인 분석을 수행하고, 그 실험 결과를 제시한다. 이를 위하여 3D-HEVC에서 사용되는 툴에 대하여 전체적인 성능 분석을 하고, 개별 툴에 대한 성능 및 일부 조합의 성능을 분석한다. 특히, 3D-HEVC에서 사용하는 깊이 영상 정보를 이용하는 툴과 텍스쳐 영상 정보를 부호화하는 툴을 구분하여 분석한다. 이러한 성능분석은 향후 부호화 효율을 향상시킬 수 있는 알고리즘 개발에 매우 유용할 것이며, 또한 고속 혹은 실시간 부호화기 개발에서도 부호화 효율 및 복잡도 측면에서의 툴의 선별 결정에 근거가 될 수 있을 것이다.

본 논문의 구성은 다음으로 이루어져 있다. 먼저 2장에서는 3D-HEVC 비디오 부호화에 대한 개요, 예측구조, 부호화 순서, 현재까지 HTM(HEVC Test Model Ver. 10.0 기준)[7]에 채택된 다시점 부호화 방법, 깊이정보 부호화 방법에 대하여 설명한다. 다음 3 장에서는 HTM에 채택된 개별 부호화 기술에 대한 부호화 및 복잡도 효율성에 대해 성능을 분석하며, 추가적으로 다양한 부호화 기술의 조합에 대한 성능 분석을 제시한다. 마지막으로 4장에서 결론을 맺고자 한다.

2. 3D-HEVC 비디오 부호화 개요

그림 1은 3D-HEVC에서 사용되고 있는 다시점 뷰(View)의 텍스쳐 영상 및 깊이 영상의 부호화 및 예측구조를 나타내고 있다. 다시점 뷰(View)에서 텍스쳐 영상과 깊이 영상의 부호화 순서는 기본 시점(View 0)에 있는 텍스쳐 영상을 시작으로 T0-D0-T1-D1-T2-D2 순으로 부호화가 진행되며, 여기서 T는 텍스쳐 영상을 말하고 D는 깊이 영상을 지칭하며, T와 D 뒤의 숫자는 시점의 순서를 정의한다. 여기서 기본 시점은 가장 먼저 부호화하는 시점이며 HEVC로 부호화한다. 비 기본 시점(Non-Base view)은 기본 시점을 제외한 나머지 시점을 지칭하며 3D-HEVC의 새로운 툴로 부호화할 수 있다. 상기 기술은 깊이 영상에 대한 부호화가 텍스쳐 영상을 부호화한 후 수행되므로 텍스쳐 우선 부호화(Texture first coding)라고 부른다. 텍스쳐 영상의 부호화가 먼저 이루어지기 때문에 텍스쳐 영상을 부호화할 때 동일 시점의 깊이 정보를 이용할 수는 없지만, 비 기본 시점들은 이전에 이미 부호화된 시점의 텍스쳐나 깊이 정보를 이용하여 부호화 효율을 향상 시킬 수 있다[7].

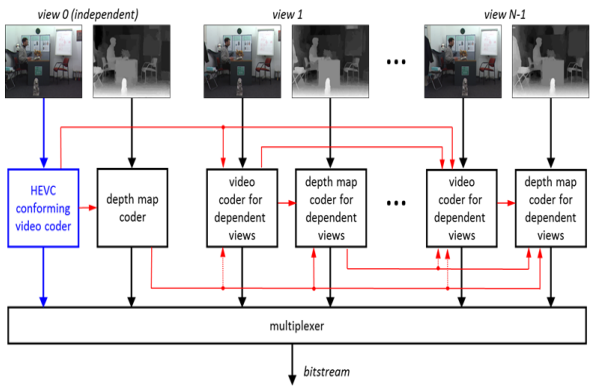

그림 1. 3D-HEVC 비디오 부호화 및 예측 구조[7]
Fig. 1. 3D-HEVC encoder and prediction structure[7]

2.1 시점간 예측

기본 시점은 디스패리티 보상 및 향상된 시점간 예측 부호화를 위하여 비 기본 시점의 참조 영상으로 사용될 수 있다. 디스패리티는 두 시점의 대응하는 블록들 간의 시점간 차이를 말하며, 비 기본 시점은 기본 시점이 이미 부호화되어 있기 때문에 디스패리티를 보상한 시점간 예측을 수행할 수 있다. 3D-HEVC의 시점간 예측구조에는 주요 부호화 기술로 기본 시점과 비 기본 시점간의 디스패리티 정보를 예측하는 기술과 시점간의 예측 향상을 위한 휘도 보상 기술, 잔차 신호 예측 기술 등이 있다[7][8].

2.1.1 IVMP(Inter-View Motion Parameter Prediction)

3D-HEVC 부호화에서는 디스패리티를 이용한 부호화 기법으로 DV-MCP(Disparity vector motion compensated prediction)와 DCP(Disparity compensated prediction)를 지원한다. MCP 부호화에서는 공간적 주변 블록에서 사용된 움직임 정보를 이용하여 현재 블록의 움직임 정보를 예측한다. 이와 달리 DV-MCP에서는 디스패리티를 이용하여 움직임 정보를 예측한다. DCP는 동일 시점의 시간적 이웃

블록을 이용하여 보상하는 기존의 MCP 기술 대신 디스패리티를 이용하여 이미 부호화된 다른 시점의 텍스쳐를 참조 블록으로 활용하는 기법이다. 이와 같이, 상기 두 가지 기술은 공통적으로 디스패리티 벡터를 필요로 한다. 따라서 3D-HEVC에서는 디스패리티 벡터를 추정하기 위해 시점간의 움직임 정보(즉, 디스패리티) 예측 기술(IVMP, Inter-View Motion Prediction)을 지원한다. 3D-HEVC에는 디스패리티 벡터를 추정하기 위한 IVMP와 관련하여 총 4 가지의 방법이 제안되었으나, 방법 1과 2는 공통 실험 조건에서 제외되었기 때문에 방법 3과 4에 대해서 설명한다.

방법 3은 텍스쳐 영상에서 디스패리티 벡터를 추정하는 방법이며, 방법 4 Do-NBDV(Depth oriented neighbouring block based disparity vector)는 깊이 영상에서 디스패리티 벡터를 추정하는 방법이다. 방법 3과 방법 4는 공통적으로 디스패리티 벡터를 먼저 추정하고, 텍스쳐 혹은 깊이 영상 부호화 과정에서 추정된 디스패리티 벡터를 활용한다. 방법 3에서 디스패리티 벡터는 CU(Coding Unit) 단계에서 도출하며 다음 4단계의 과정으로 수행된다. 4단계 중 어느 한 단계에서 디스패리티 벡터가 추정된 경우 더 이상의 단계는 수행하지 않는다. 1 단계로 그림 2와 같이 시간적 이웃 블록을 검색한다. 시간적 이웃 블록이 디스패리티 벡터를 포함하고 있으면 이를 현재 블록의 디스패리티 벡터로 추정한다. 만약 그렇지 않다면 2 단계로 그림 2에서의 공간적 이웃 블록의 디스패리티 벡터를 검색한다. 앞선 두 가지 단계에서 모두 디스테리티 벡터를 얻을 수 없을 경우 3 단계에서는 MCP로 부호화된 공간적 이웃 블록을 이용한다.

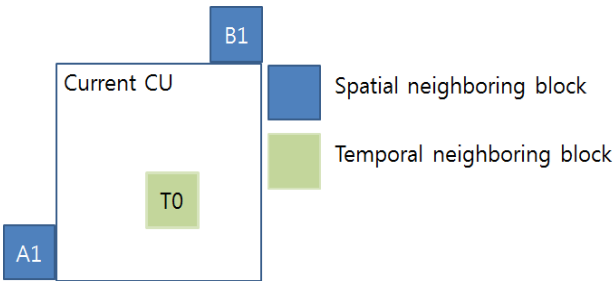

그림 2. 공간적 및 시간적 이웃 블록
Fig. 2. Location of spatial and temporal neighbour blocks

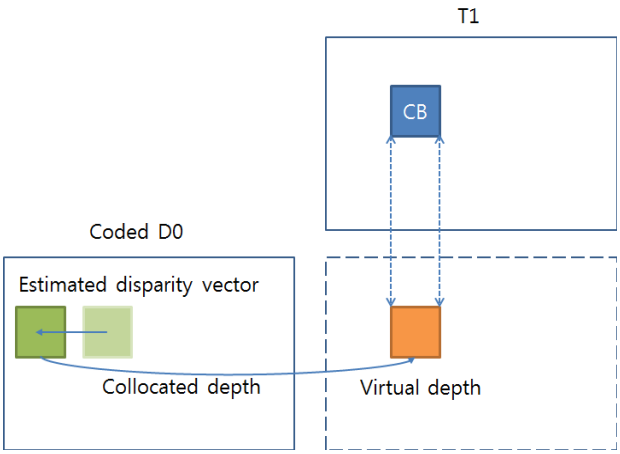

그림 3. 텍스쳐는 MCP로 부호화되었으며 움직임 정보는 IVMP로 부호화된 공간적 이웃 블록
Fig. 3. The inter-view predicted motion vector of a MCP coded block

그림 3과 같이 임의 이웃 블록이 MCP로 부호화되었으며 이 블록의 움직임 정보가 IVMP에 의해 예측되었다면, 이때 사용된 디스패리티 벡터를 현재 블록의 디스패리티로 추정한다. 상기 세 가지의 단계가 모두 불가능한 경우 마지막으로 제로 디스패리티 벡터를 현재 블록의 디스패리티 벡터로 결정한다[7][9].

방법 4 Do-NBDV는 이웃 시점의 깊이 정보를 이용하여 현재 블록의 깊이 정보를 예측(Virtual depth)하고, 이 Virtual depth를 기반으로 디스패리티를 추정한다. 자세히 살펴보면, 그림 4에 도시한 것처럼 우선 방법 3을 사용하여 디스패리티 벡터를 추정한 후, 참조 시점 깊이 영상에서 그 추정된 디스패리티 벡터가 가리키는 위치의 대응 블록을

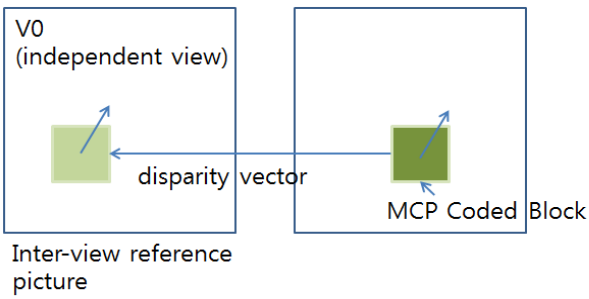

그림 4. 가상 깊이 블록의 유도과정
Fig. 4. Derivation of the virtual depth block

도출한다. 이 대응 블록의 깊이 정보를 현재 블록의 깊이 값으로 간주하며, 그 깊이 값을 이용하여 방법 3으로 찾은 디스패리티 벡터를 보정하여 더 정확한 디스패리티를 추정한다[7][9].

2.1.2 IVRP(Inter-View Residual Parameter Prediction)

IVRP(Inter-View Residual Parameter Prediction)는 시점간 잔차 신호 예측 기법으로 단 방향 예측에 이용된다. 그림 5와 같이, 현재 부호화하려는 블록 Dc가 동일 시점의 시간적 이웃 영상의 블록의 Dr로 예측되었고(즉, MCP로 예측), Dc의 디스패리티 벡터로 찾은 참조 시점의 대응 블록 Bc도 MCP로 예측되었을 때, Dr과 Dc 사이의 잔차 신호를 Dc에 대응하는 참조 시점의 블록 Bc의 잔차 신호로 예측하는 기법이다. Dc와 Bc가 동일한 움직임 객체로서 동일한 움직임을 갖는다면, 두 블록의 잔차 신호가 매우 유사할 가능성이 높기 때문에 효과적인 잔차 신호 예측을 수행할 수 있다[7][10].

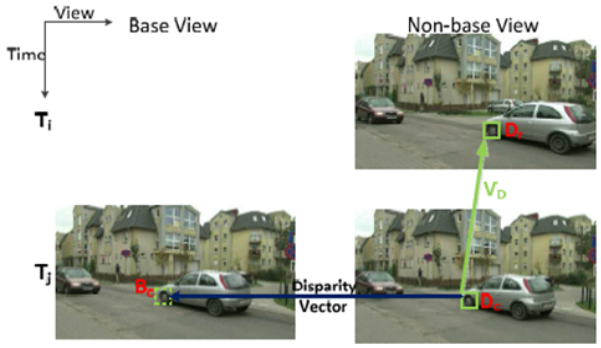

그림 5. 시점간 잔차신호 예측[11]
Fig. 5. Inter-view residual prediction[11]

2.1.3 ARP(Advanced Residual Prediction)

ARP는 상기의 IVRP의 잔차 신호 예측의 성능 향상을 위해 제안되었다. 그림 5에서 참조 시점의 블록 Bc와 부호화 대상 시점의 Dc가 서로 다른 움직임 정보를 갖는 경우 잔차 신호가 서로 상이할 가능성이 높으므로 IVRP의 성능이 떨어질 수 있다. 이러한 단점을 해결하기 위해서 ARP는 움직임 정보를 추가적으로 사용한다. 그림 6과 같이, 현재

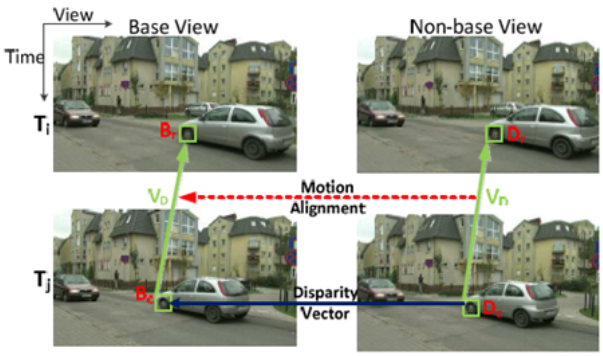

그림 6. 향상된 잔차 신호 예측(Advanced Residual Prediction)[11]
Fig. 6. Advanced Residual Prediction[11]

부호화하려는 블록 Dc의 움직임 벡터 VD를 블록 Bc에 적용하여 Br을 찾고, Bc와 Br 사이의 잔차 신호에 가중치를 곱하여 Dr과 Dc의 잔차신호 예측 값으로 이용한다[7][11].

2.1.4 IC(Illumination Compensation)

다시점(view)에서는 카메라의 성능 혹은 조명 차이로 인해 동일한 객체임에도 불구하고 시점간에 밝기 혹은 컬러 값이 차이날 수 있다. 이러한 차이를 줄이기 위해서 3D-HEVC에서는 IC(Illumination compensation), 즉 조명 보상 모델을 지원한다. 그림 7과 같이, CU 레벨에서 현재 부호화하려는 블록의 복원된 이웃 화소들과 현재 PU의 디스패리티 벡터로 찾은 참조 시점의 참조 블록 주위의 복원된 화소들을 이용하여 식 (1)을 최소화시키는 조명 보상을

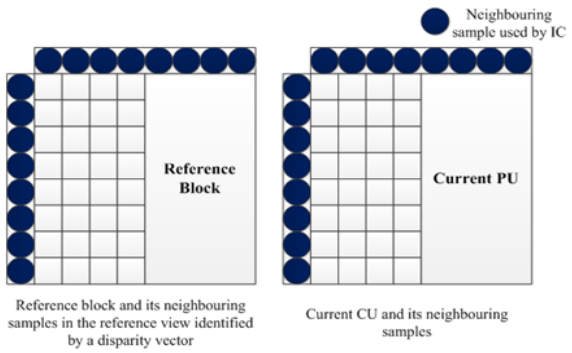

그림 7. 조명 보상에 이용되는 현재 블록 및 참조 블록의 이웃 화소[8]
Fig. 7. Neighbouring samples for the derivation of illumination compensation parameters[8]

위한 파라미터인 스케일링 팩터 a와 오프셋 b 값을 산출한다.

$$E(a,b)=\sum^{i}(Y_i-aX_i-b)^2+\lambda(a-1)^2 \quad (1)$$

여기서 X_i는 참조 블록의 이웃 샘플 값이고, Y_i는 현재 블록의 이웃샘플 값을 지칭한다. 이 조명 보상은 텍스쳐 영상에만 적용한다[7][12].

2.2 깊이 영상 부호화 기술

2.2.1 MPI(Motion Parameter Inheritance)

MPI(Motion Parameter Inheritance)는 텍스쳐 영상의 움직임 정보가 깊이 영상의 움직임 정보와 높은 중복성을 가지고 있는 점을 이용하여 깊이 영상의 움직임 정보를 부호화하는 기술이다. 3D-HEVC에서는 깊이 영상의 움직임 정보를 이미 부호화된 동일 시점 및 동일 시간의 텍스쳐 영상으로부터 상속하여 부호화할 것인지 기존 HEVC의 움직임 정보 부호화 방법으로 부호화 할 것인지를 결정시킬 수 있다. 이와 같이 텍스쳐 영상으로부터 움직임 정보를 상속하기 위해서는 각 CU 블록에 대하여 움직임 정보의 상속 여부를 알려주는 구문이 각 CU 블록 레벨에서 전송되어야 한다. 3D-HEVC에서는 움직임 정보뿐만 아니라 CU의 분할 정보도 상속할 수 있다. 따라서 깊이 영상의 CU가 움직임 정보와 블록 분할 정보를 모두 상속할 것인지 둘 중 하나만 상속할 것인지를 적응적으로 결정할 수 있다. CU의 분할 정보 상속과 관련하여 자세한 알고리즘은 다음 절에서 소개한다[7][14].

2.2.2 QTLPC(Quadtree limitation Predictive Coding)

3D-HEVC에서 QTLPC(Quadtree limitation Predictive Coding)는 깊이 영상의 CU 쿼드트리 분할 정보를 부호화하기 위해서 동일 시점 및 동일 시간의 텍스쳐 영상의 CU 분할 정보를 이용하는 기술이다. 동일 시점에서 텍스쳐 영상과 깊이 영상은 서로 높은 상관성을 갖는다. 따라서 텍스쳐 영상의 부호화과정에서 사용된 쿼드트리 정보를 깊이 영상의 분할 구조에 적용한다면, 깊이영상의 부호화시 이를 부호화하기 위한 정보량을 줄일 수 있다. 또한, 텍스쳐 영상에 비해 깊이 영상의 텍스쳐는 비교적 간단하므로, 텍스쳐 영상에서 사용한 CU의 최대 깊이 값을 제한시켜 깊이 영상의 CU 분할에 적용시킴으로써 텍스쳐 영상보다 간단한 깊이 영상의 CU 쿼드트리를 생성할 수 있다. QTLPC를 적용하면 깊이 영상의 PU 모양도 텍스쳐 영상의 PU 모양에 따라 제한적으로 사용한다. 그림 8은 텍스쳐의 PU 모양에 따라 허용되는 깊이 영상의 PU 모양을 도시하고 있다. 이와 같이, 텍스쳐 영상의 대응 PU가 2N×2N, 2N×N, N×2N인 경우 깊이 영상은 2N×2N의 PU만 가능하기 때문에 PU 모양 결정을 위한 부호화기의 복잡도를 상당히 감소시킬 수 있다[8][15].

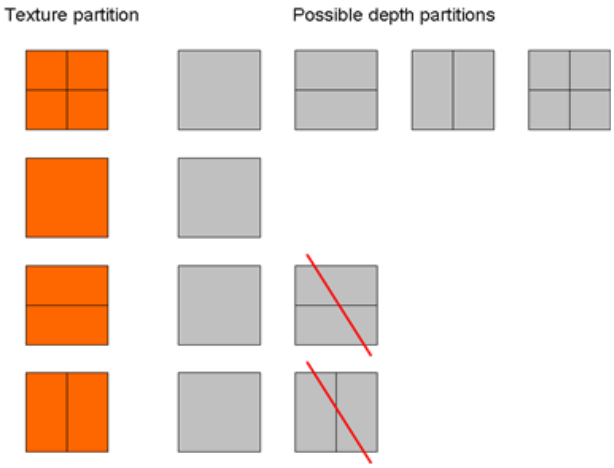

그림 8. 텍스쳐의 PU 분할에 따라 허용되는 깊이 PU의 분할[15]
Fig. 8. Texture partitions and corresponding possible depth partitions[15]

2.2.3 DMM(Depth Modeling Mode)

DMM은 깊이 영상에서 객체의 경계 구분이 명확한 점과 객체 내부의 값 변화가 적은 특징을 이용한 깊이 영상의 화면 내 부호화 기술이다. HEVC에서 지원하는 PU 분할 구조 및 35개의 화면 내 예측 모드에 추가적으로, 직선과 곡선을 이용하여 블록을 보다 자유롭게 분할하고 각 분할된 영역에 속하는 화소들은 DC 모드와 같이 하나의 평균값을 이용하여 예측하는 방법이다. 여기서 직선은 웨지렛(Wedgelet)으로 부르며 곡선은 컨투어(Contour)라고 부른다. 그림 9는 웨지렛과 컨투어를 이용한 분할 방식의 예를

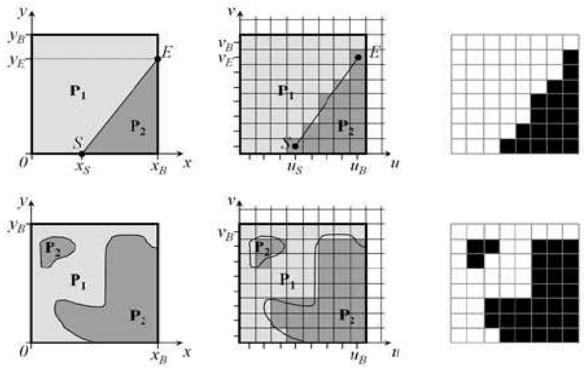

그림 9. 웨지렛(위) , 컨투어(아래)[13]
Fig. 9. Wedgelet(up), Contour(down)[13]

보여준다. 이러한 분할을 지원하기 위해서는 복호기에 웨지렛과 컨투어가 어떠한 모양인지를 알려주거나 복호기가 추정할 수 있어야 한다. 이에 따라 DMM은 이 웨지렛과 컨투어를 어떻게 설정했는지에 대한 정보를 부호화하는 방식에 따라 모드 1부터 모드 4까지 총 4 가지의 모드를 지원한다. 여기서 모드 1부터 모드 3은 웨지렛을 부호화하기 위한 모드이며, 모드 4는 컨투어를 부호화하기 위한 모드이다.

모드 1은 웨지렛을 직접 전송하는 방법으로 다른 참조 정보 없이 웨지렛의 출발점(그림 9의 S)과 끝점(그림 9의 E)의 위치를 미리 정의된 테이블을 이용하여 표현한 후 이를 직접 전송한다. 모드 2는 주변 블록의 기존 HEVC 화면 내 부호화 모드 혹은 DMM 모드의 방향성을 고려하여 그 연장선 형태로 현재 부호화하려는 블록을 분할하는 구조이다. 따라서 주변 블록을 참조하여 웨지렛의 출발점을 예측하고 끝점은 모드 1과 같이 부호화한다. 모드 3는 텍스쳐 블록의 휘도 성분이 깊이 영상의 휘도 성분과 비슷한 점을 이용하여 휘도 성분의 에지 정보를 기반으로 가장 근사한 웨지렛을 결정하고 이를 깊이 영상에 동일하게 적용하는 방법이다. 마지막으로 모드 4는 직선이 아닌 곡선을 예측하는 방법이다. 곡선에 대한 분할은 직선으로 분할할 수 있는 경우의 수보다 훨씬 더 많기 때문에 곡선의 정보를 별도로 부호화하기 어렵다. 따라서 현재 부호화하려는 깊이 영상의 블록과 동일한 위치의 텍스처 영상 블록인 CTLB(co-located texture luma block)의 휘도 평균을 구하고 이를 임계값으로 이용하여 CTLB를 분할한 후, 그 분할 정보를 깊이 영상에 동일하게 적용한다[7].

상기의 방법으로 웨지렛 혹은 컨투어로 분할된 영역은 각 영역별로 하나의 DC 값으로 예측한다. 즉, 그림 10과 같이 P1 영역에 속하는 모든 화소는 P1의 DC 값으로 예측하며, P2 영역에 속하는 모든 화소는 P2의 DC 값으로 예측한다. 이 DC 값은 그림 10과 같이 패턴에 따라 하나 또는 2개의 이웃 참조 샘플을 사용하여 예측 부호화한다[7].

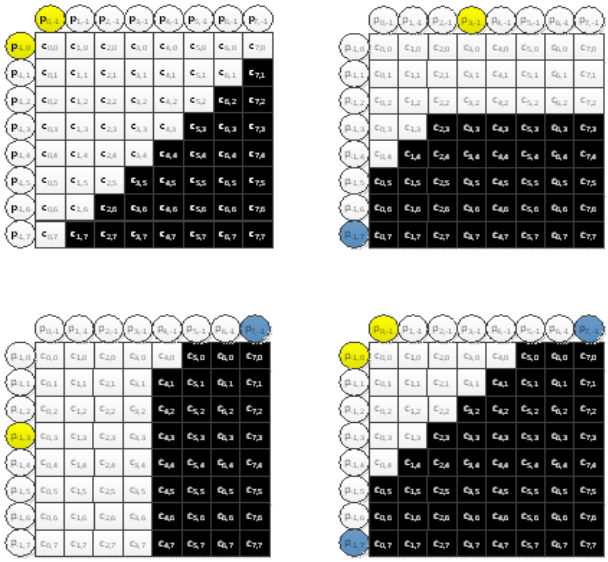

그림 10. 분할 패턴에 따른 참조 샘플 위치[7]
Fig. 10. Selection of reference samples for difference partition pattern cases[7]

3. 실험 결과 및 분석

본 논문에서는 앞서 기술한 텍스쳐 부호화에 사용되는 IVMP, ARP, IC와 깊이 부호화에서 사용하는 MPI, DMM, QTLPC에 대한 개별 및 조합 성능을 비교 분석한다. 이를 위해서, 기존 HTM에서 모든 툴을 이용하였을 때와 각각 툴을 사용하였을 때를 비교하고, 서로 다른 2 가지 이상의 툴을 조합 했을 때의 성능에 대하여 분석한다. 이러한 분석은 향후 부호화 성능 향상을 위한 알고리즘 개발 및 고속 부호화기 개발을 위해 유용한 정보를 제공할 수 있을 것으로 판단된다. 본 논문의 실험에서는 3D-HEVC의 참조 소프트웨어인 3DV-HTM 10.0을 이용하였다. 실험에 사용된

PC는 Intel Sandy Bridge 2.4GHz E5 2665, RAM은 128GB RAM이며. 테스트 영상으로는 3D-HEVC CTC(Common Test Condition)[16]에서 제공하는 영상 중 클래스 A(1024x768)의 Balloons, Kendo, Newpaper 영상과 클래스 C (1920x1088)의 GT_Fly, Poznan_Street 영상을 이용하였다. 실험조건 또한 CTC에 따라 텍스쳐 영상의 양자화계수(QP: quantization parameter)를 25, 30, 35, 40으로 설정하였으며, 깊이 영상의 양자화 계수는 텍스쳐 영상보다 높은 34, 39, 42, 45로 설정하였다. 깊이영상의 양자화 계수를 더 높게 설정하는 이유는 깊이 영상이 텍스쳐 영상에 비해 단순하여 양자화 계수가 텍스쳐 영상보다 높아도 부호화시에 큰 영향을 주지 않기 때문이다. 표 2는 본 실험에 이용된 테스트 영상에 대한 해상도, 시점, 총 프레임 수를 나타낸다.

표 2. 테스트 영상 정보
Table 2. Test sequences

테스트 영상	공간 해상도	시점 순서	프레임
GT_Fly	1920x1088	5-9-1	250
Poznan_Street	1920x1088	5-1-9	250
Newspaper	1024x768	4-2-6	300
Balloons	1024x768	3-1-5	300
Kendo	1024x768	3-1-5	300

첫 번째 실험으로 전체 툴을 사용할 때의 성능을 실험하였다. 표 3은 3D-HEVC에서 사용하는 기술을 모두 사용하는 경우와 모두 사용하지 않을 경우에 비해 얼마의 부호화 성능을 갖는지를 보여준다. 즉 모두 사용하지 않는 경우는 각 시점을 독립적으로 부호화(simulcast)하며 깊이 영상도 기존 HEVC로 단순히 독립적으로 부호화하는 경우이다. 따라서 이 실험을 통해 시점간 예측의 성능 및 깊이 영상 부호화 툴의 성능을 유추할 수 있다. 실험에서는 총 3 개의 시점을 사용하였으며, View 0는 기본 시점(base view)이며, 기존의 HEVC 호환성을 가지고 있기 때문에 성능 면에서 차이가 없으므로 표에 서술하지 않았다. 표 3에서와 같이 simulcast로 부호화하는 것보다 3D-HEVC는 평균 약 27%의 비트율을 감소시킬 수 있었다. 이와 같이 3D-HEVC는 매우 높은 부호화 효율을 갖고 있다는 것을 알 수 있다. 이는 시점간 중복성이 매우 높고 3D-HEVC의 부호화 툴들이 이 중복성을 효과적으로 제거했기 때문이다.

표 3. 각 시점 영상을 독립 부호화하는 경우 대비 3D-HEVC의 부호화 효율
Table 3. Coding efficiency of 3D-HEVC compared with simulcast

테스트 영상	부호화 효율(BD-rate)	
	View 1	View 2
GT_Fly	-30.4%	-29.8%
Poznan_Street	-18.5%	-19.2%
Newspaper	-21.7%	-20.4%
Balloons	-33.7%	-32.9%
Kendo	-30.9%	-33.1%
Average	-27.04%	-27.08%

그림 11과 그림 12는 표 3에서 부호화 성능이 가장 높은 영상과 가장 낮은 영상을 Y-PSNR 대비 bitrate로 나타낸 그래프이다. 가장 좋은 성능을 보인 영상은 Balloons의 view 1 시퀀스이고, 가장 낮은 성능을 보인 영상은 Poznan_Street의 view1 시퀀스이다. 그림에서와 같이 3D-HEVC는 모든 비트율 구간에서 꾸준하게 우수한 성능을 보이며, 특히 비트율이 높아질수록 더 우수한 성능을 나타낸다.

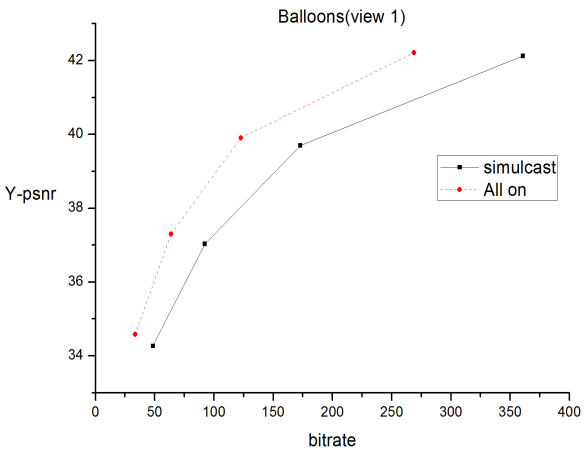

그림 11. 표 3에서 가장 높은 성능을 갖는 Balloons 영상의 율-왜곡 곡선
Fig. 11. R-D Curve for sequence Balloons that has the best coding efficiency in Table 3

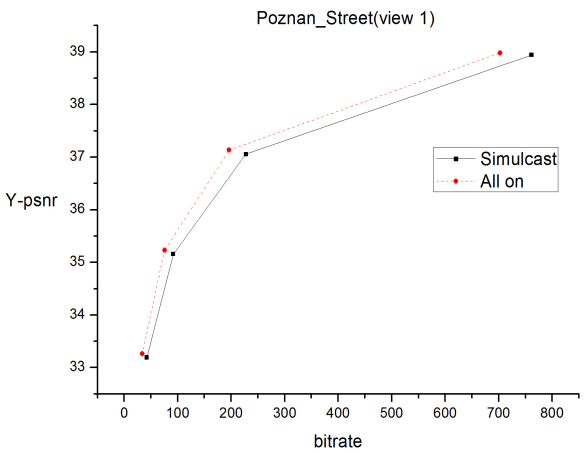

그림 12. 표 3에서 가장 낮은 성능을 갖는 Poznan_Street 영상의 율-왜곡 곡선
Fig. 12. R-D Curve for sequence Poznan_Street that has the worst coding efficiency in Table 3

두 번째 실험으로 시점간 예측 기술에 대한 성능을 평가하였다. 시점간 예측 기술만의 성능 측정을 위해서 깊이 영상을 부호화하기 위한 툴은 모두 제외하였다. 표 4는

표 4. 3D-HEVC 시점간 예측 기술의 부호화 효율
Table 4. Coding efficiency of inter-view prediction tools in 3D-HEVC compared with simulcast

테스트 영상		부호화 효율 (BD-rate)			
		IC	ARP	IVMP	IC+ ARP+ IVMP
GT_Fly	View 1	0.0%	-3.0%	-21.7%	-21.9%
	View 2	0.0%	-2.3%	-21.6%	-21.8%
Poznan_ Street	View 1	0.3%	-6.3%	-11.7%	-15.1%
	View 2	-0.9%	-6.5%	-11.6%	-16.0%
Newspap er	View 1	-0.8%	-8.0%	-14.4%	-19.6%
	View 2	-2.1%	-7.3%	-12.9%	-18.6%
Balloons	View 1	-3.1%	-13.8%	-24.0%	-31.5%
	View 2	-2.9%	-13.0%	-23.0%	-30.4%
Kendo	View 1	-4.7%	-11.8%	-18.5%	-28.6%
	View 2	-6.3%	-12.9%	-18.2%	-30.5%
Average	View 1	-1.66%	-8.58%	-18.06%	-23.34%
	View 2	-2.26%	-8.4%	-17.46%	-23.66%

II.2-1 시점간 예측에서 설명한 IVMP, ARP, IC 각 툴의 성능과 이 3 가지 툴을 같이 사용하였을 때의 성능을 보여준다. IC 기술에서 가장 좋은 성능을 보인 영상은 Kendo 영상으로 각 View1, 2에서 -4.7%와 -6.3%의 BD-rate를 보여주었고, 가장 낮은 성능을 보인 영상은 GT_Fly 영상으로 View1, 2에 성능 변화가 거의 없었다. Kendo 영상은 자연영상이기 때문에 IC 기술이 효율을 발휘하였지만, GT_Fly 영상은 스크린 콘텐츠로서 그래픽 영상이므로 시점간의 밝기 값 차이가 존재하지 않는다. 따라서 IC 기술로 인한 부호화 효율이 없었다. ARP 기술은 동일한 추세로 가장 좋은 영상은 Kendo 영상으로 View1, 2에 대해 각각 -13.8%, -13.0%의 성능을 보였고, 가장 낮은 성능을 보인 영상은 GT_Fly 영상으로 -3.0%와 -2.3%의 성능을 보였다. 이는 GT_Fly 영상이 그래픽 영상이기 때문에 잔차 신호의 양이 적어 잔차 신호 예측 기술의 성능이 낮은 것으로 보인다. 마지막으로 IVMP에서는 Balloons이 View1, 2에 대해 -24%, -23%의 성능으로 가장 좋았고, Poznan_Street이 View1, 2에 대해 -11.7%, -11.6%로 가장 낮은 성능을 보여주었다. 표 4에서 Balloons 영상이 IVMP를 이용할 시 가장 성능이 좋은 이유는 영상내의 객체 움직임이 매우 동적이여서 움직임 벡터가 비트스트림에서 차지하는 비율이 다른 영상에 비해 상당히 높기 때문이다. 조합 성능에서 보듯이 3 개의 툴의 성능이 단순히 더해지지는 않으나, 서로 보완적 역할을 수행하여 하나의 툴만 사용하는 것보다는 높은 성능을 나타내었다.

그림 13, 14는 표 3의 IC, ARP, IVMP 총 3 가지 툴을 같이 사용하였을 경우, 테스트 영상에 대한 율-왜곡 곡선을 보여준다. 가장 좋은 성능을 보여준 영상은 Balloons이고, 가장 낮은 성능을 보여준 영상은 Poznan_Street이다. 그림에서와 같이 IVMP가 모든 비트율에서 가장 두드러진 성능을 나타낸다.

다음으로 깊이 영상 부호화 툴에 대한 성능을 실험하였다. 깊이 영상 부호화의 효율 평가를 위해서, 시점간 부호화 기술은 모두 사용하고 깊이 영상 부호화 기술만 사용하지 않은 것을 비교 대상 기준으로 삼았다. 이는 3D-HEVC의 깊이 영상 부호화 기술 중 MPI는 시점간 부호화 기술을 사용하기 때문에 성능 비교 시 공정성을 높이기 위함이다.

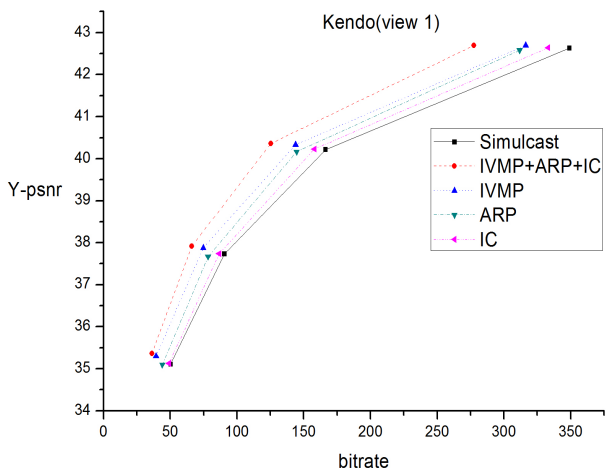

그림 13. 표 4에서 가장 높은 성능을 갖는 Kendo 영상의 율-왜곡 곡선
Fig. 13. R-D Curve for sequence Kendo that has the best coding efficiency in Table 4

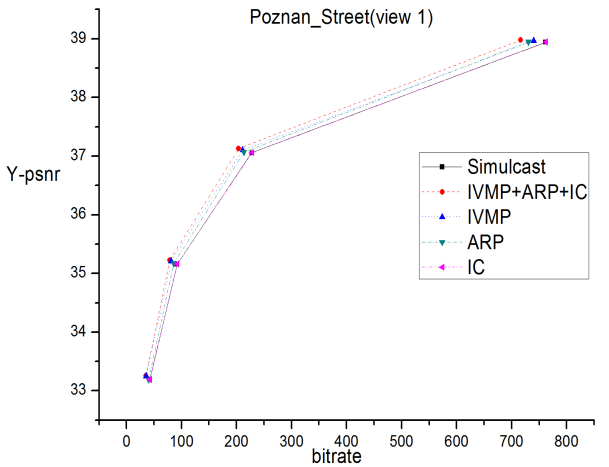

그림 14. 표 4에서 가장 낮은 성능을 갖는 Poznan_Street 영상의 율-왜곡 곡선
Fig. 14. R-D Curve for sequence Poznan_Street that has the worst coding efficiency in Table 4

II.2-2 깊이 영상 부호화 기술에서 설명한 3D-HEVC의 MPI, DMM, QTLPC에 대한 성능 분석 결과를 표 5에 나타내었다. 표 5에서 T는 텍스쳐 영상을 말하고, D는 깊이 영상을 가리킨다. 텍스쳐 영상의 첫 번째 시점은 HEVC 호환이고 성능차이가 없기 때문에 표에는 나타내지 않았다. 표에서와 같이 3 가지 툴 중 MPI가 평균적으로 가장 높은 부호화 효율을 나타내었다. 텍스쳐 영상과 깊이 영상은 동일한 객체를 포함하며 동일한 시간에 획득된 영상이다. 따라서 움직임 정보가 매우 유사하며 텍스쳐로부터 움직임 정보를 상속하는 MPI가 높은 성능을 나타낼 수 있다. 특히

표 5. 3D-HEVC 깊이 영상 부호화 기술 성능
Table 5. Coding efficiency of depth map coding tools when inter-view coding tools are turned on

테스트 영상		부호화 성능(BD-rate)		
		MPI	DMM	QTLPC
GT_Fly	View 1(T)	-6.7%	-8.5%	-4.5%
	View 2(T)	-8.6%	-7.1%	-3.9%
	View 0(D)	-13.1%	-8.6%	-1.4%
	View 1(D)	-8.5%	-5.4%	-3.7%
	View 2(D)	-9.4%	-3.6%	-3.0%
Poznan_ Street	View 1(T)	-4.4%	-4.3%	-10.0%
	View 2(T)	-3.9%	-3.9%	-9.4%
	View 0(D)	-10.6%	-1.7%	-0.3%
	View 1(D)	-11.2%	0.7%	-4.4%
	View 2(D)	-8.5%	-0.1%	-4.3%
Newspaper	View 1(T)	-2.5%	-2.5%	-2.5%
	View 2(T)	-2.0%	-2.1%	-2.0%
	View 0(D)	-7.1%	-2.2%	5.3%
	View 1(D)	-5.6%	-2.0%	1.4%
	View 2(D)	-4.8%	-1.6%	4.4%
Balloons	View 1(T)	-3.2%	-3.3%	-3.2%
	View 2(T)	-3.7%	-3.6%	-3.8%
	View 0(D)	-6.5%	-0.7%	3.3%
	View 1(D)	-7.0%	-0.9%	1.7%
	View 2(D)	-9.4%	-0.3%	1.9%
Kendo	View 1(T)	-3.4%	-3.5%	-3.3%
	View 2(T)	-4.0%	-3.8%	-4.1%
	View 0(D)	-3.2%	0.0%	3.6%
	View 1(D)	-5.8%	0.6%	3.7%
	View 2(D)	-5.5%	-0.6%	5.5%
Average	View 1(T)	-4.04%	-3.24%	-2.81%
	View 2(T)	-4.14%	-3.93%	-4.26%
	View 0(D)	-6.93%	-2.97%	0.33%
	View 1(D)	-6.36%	-1.00%	0.84%
	View 2(D)	-7.03%	-0.71%	1.70%

GT_Fly 영상에 대한 성능이 가장 높은 이유는 GT_Fly이 전체적으로 줌-인(zoom-in)을 하는 동적인 그래픽 영상이기 때문이다. 즉, 영상 전체적으로 움직임 정보가 많고, 이것이 MPI에 의해 효과적으로 부호화됐다는 것을 알 수 있다. 또한, DMM에 대한 성능 분석에서도 GT_Fly 영상이 가장 높은 성능을 보였다. GT_Fly 영상은 텍스쳐와 깊이 영상 모두 그래픽 영상으로 컴퓨터로 생성한 영상이다. 따라서 평활한 영역이 많으며 주변 화소와의 상관성이 매우 높은 특징이 있다. 또한 자연 영상에 비해 에지의 경계가 뚜렷하기 때문에 컨투어나 웨지렛으로 에지를 효과적으로 나타내는 DMM이 높은 성능을 나타내었다.

QTLPC는 깊이 영상 부호화 시 텍스쳐 영상으로부터 CU 및 PU의 분할 정보를 상속하는 기술로 GT_Fly와 Poznan_Street에서는 부호화 효율을 나타내지만, 나머지 영상에서는 오히려 부호화 손실이 발생한다. 이는 QTLPC가 깊이 영상에서 최대 CU 분할 깊이를 제한하며 PU의 모양도 제약을 가하기 때문이다. 하지만 QTLPC는 이러한 제약조건들로 인해 부호화 복잡도를 상당히 감소시킬 수 있다. 이에 대한 실험 결과는 아래에 기술한다.

그림 15, 16은 표 5에서 텍스쳐 영상에 대해 전체적으로 가장 높거나 낮은 부호화 효율을 보인 테스트 영상에 대한

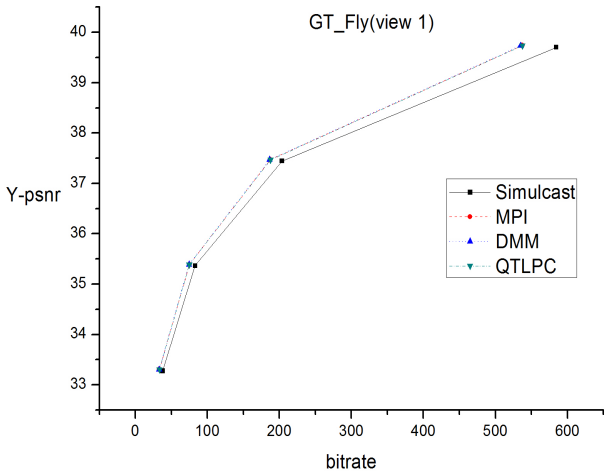

그림 15. 표 5에서 가장 높은 성능을 갖는 GT_Fly 텍스쳐 영상의 율-왜곡 곡선
Fig. 15. R-D Curve for sequence GT_Fly(Texture) that has the best coding efficiency in Table 5

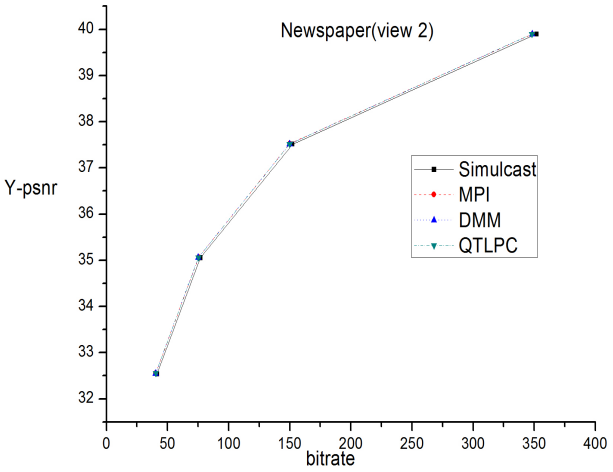

그림 16. 표 5에서 가장 낮은 성능을 갖는 Newspaper 텍스쳐 영상의 율-왜곡 곡선
Fig. 16. R-D Curve for sequence Newspaper(Texture) that has the worst coding efficiency in Table 5

표 6. 3D-HEVC QTLPC의 복잡도 성능
Table 6. complexity efficiency of QTLPC

테스트 영상	QTLPC의 부호화 시간
GT_Fly	74.1%
Poznan_Street	74.0%
Newspaper	87.9%
Balloons	80.2%
Kendo	87.2%
Average	80.68%

각 툴의 율-왜곡 곡선을 나타낸다. 가장 높은 성능을 보인 영상은 Poznan_street의 View 1이고, 가장 낮은 성능을 보인 영상은 Newspaper의 View 2이다. 그림에서와 같이 각 툴의 성능이 비트율에 따라 그 순위가 거의 고정적으로 유지되는 것을 알 수 있다. 그림 17, 18는 표 5에서 깊이 영상에 대해 전체적으로 가장 높거나 낮은 부호화 효율을 보인 테스트 영상에 대한 각 툴의 율-왜곡 곡선을 나타낸다. 가장 좋은 성능을 보인 영상은 GT_Fly의 View 0이고, 가장 낮은 성능을 보인 영상은 Kendo의 View 0이다. 그림에서와 같이, 깊이 영상 부호화 툴이 그림 15 및 그림 16의 텍스쳐

영상에서보다 깊이 영상에서 더 높은 부호화 효율을 보임을 확인할 수 있다. 그림 18에서 보이는 것처럼, 낮은 비트율에서는 DMM과 MPI의 성능이 차이가 나지만 높은 비트율로 갈수록 그 차이가 줄어드는 것을 알 수 있다. 낮은 비트율에서는 상대적으로 움직임 정보의 비중이 높기 때문에 MPI가 높은 성능을 발휘하는 것이며, 높은 비트율에서는 고화질 부호화를 위해서 DMM의 예측 방식이 효과를 발휘하기 때문이다.

앞서 기술한 것과 같이 QTLPC는 부호화 복잡도를 감소시킬 수 있다. 이에 대해 QTLPC의 복잡도를 전체 영상을 부호화하기 위한 실행 시간(execution time)으로 측정하였다. 모든 시점간 예측 부호화 툴만을 사용하며 모든 깊이 영상 부호화 툴을 사용하지 않는 기준에 대비한 QTLPC의 복잡도 성능을 표 6에 나타내었다. HEVC는 CU 분할을 결정하기 위해 상당한 부호화 복잡도를 요구한다. QTLPC는 이렇게 복잡한 CU분할 과정을 깊이 영상 부호화에서는 생략할 수 있으므로 실험 결과와 같이 평균 80.68%의 매우 높은 복잡도 감소 효과를 얻을 수 있다.

V. 결 론

본 논문에서는 현재 표준화가 진행 중인 3D-HEVC의 주요 기술들을 살펴보고 이에 따른 기술 성능 평가를 하였다. 성능 평가에 대한 의의는 현재 3D-HEVC에서 사용되는 각각의 기술들이 기존의 방법보다 어떠한 점에서 효율적이고, 테스트 영상의 특징에 따라 어떠한 기술들이 더 효과적인지를 기술하고자 하였다. 이러한 분석은 향후 부호화 성능 향상을 위한 알고리즘 개발뿐 아니라 고속 부호화기 개발을 위한 알고리즘 개발에도 유용할 수 있을 것으로 판단된다.

디스플레이 및 카메라의 발전으로 향후 3차원 비디오에 대한 수요 및 요구는 점차 증대될 것으로 기대된다. 따라서 3차원 비디오 부호화에 대하여 많은 관심과 연구가 이루어지고, 이는 더욱 실감나는 방송 기술을 이끌 수 있는 원동력이 될 수 있을 것이다.

참 고 문 헌 (References)

[1] A. Smolic and P. Kauff, “Interactive 3D video representation and coding technologies," Proc. of IEEE, Special Issue on Advances in Video Coding and Delivery, pp. 98-110, Jan. 2005.

[2] ISO/IEC JTC1/SC29/WG11, “Reference software of depth estimation and views synthesis for FTV/3DV,” M15836, Oct. 2008.

[3] Y. Ho, 3D Video Compression MPEG 3DV Standardization, IDEC Newsletter, pp. 10-13, May 2012.

[4] ISO/IEC JTC1/SC29/WG11, “Applications and requirements on 3D video coding,” N12035, Geneva, Switzerland, March 2011.

[5] ISO/IEC JTC1/SC29/WG11, “Call for proposals on 3D video coding technology," N12036, Geneva, Switzerland, March 2011.

[6] ISO/IEC JTC1/SC29/WG11, “MPEG time line," N14404, Valencia, Spain, April 2014.

[7] Y. Chen, G. Tech, K. Wegner, and S. Yea, “Test model 8 of 3D-HEVC and MV-HEVC," JCT3V-H1003, Valencia, Spain, April 2014.

[8] L. Zhang, G. Tech, K. Wegner, and S. Yea, “Test model 7 of 3D-HEVC and MV-HEVC," JCT3V-G1005, San Jose, US, Jan. 2014.

[9] J. An, Y.-W. Chen, J.-L. Lin, Y.-W. Huang, and S. Lei, “CE 5 related: Inter-view motion prediction for HEVC-based 3D video coding," JCT3V-A0049, Stockholm, Sweden, July 2012.

[10] G. Tech, K. Wegner, Y. Chen, and S. Yea, “3D-HEVC test model 2,” JCT3V-B1005, Shanghai, China, Oct. 2012.

[11] X. Li, L. Zhang, and Y. Chen, “Advanced residual prediction in 3D-HEVC,” Proc. of 20th IEEE International Conference Image Processing (ICIP), pp. 1747-1751, Sept. 2013.

[12] J. Lee, “Description of Core Experiment 2: Illumination compensation," JCT3V-A1102, Stockholm, Sweden, July 2012.

[13] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. H. Rhee, G. Tech, M. Winken, and T. Wiegand, “3D high-efficiency video coding for multi-view and depth data,” IEEE Trans. on Image Processing, Vol. 22, No. 9, pp. 3366-3378, Sept. 2013.

[14] J. Jung and K. Viswanathan, “Depth quadtree prediction for HTM," JCT3V-A0044, Stockholm, Sweden, July 2012.

[15] X. Zhang, K. Zhang, J. An, J.-L. Lin, and S. Lei, “CE2 related: A texture-partition-dependent depth partition for 3D-HEVC," JCT3V-G0055, San Jose, US, Jan. 2014.

[16] D. Rusanovskyy, K. Mueller, and A. Vetro, “Common test conditions of 3DV core experiments,” JCT3V-E1100, Vienna, Austria, Aug. 2013.

저 자 소 개

박 대 민

- 2013년 2월 : 한밭대학교 정보통신공학부 멀티미디어공학과 (공학사)
- 2013년 3월 ~ 현재 : 한밭대학교 정보통신전문대학원 멀티미디어공학과 (석사과정)
- 주관심분야 : 비디오 부호화, 게임 프로그래밍, 3D 애니메이션

최 해 철

- 1997년 2월 : 경북대학교 전자공학과 학사
- 1999년 2월 : 한국과학기술원 전기및전자공학과 (공학석사)
- 2004년 8월 : 한국과학기술원 전기및전자공학과 (공학박사)
- 2004년 9월 ~ 2010년 2월 : 한국전자통신연구원(ETRI) 방송미디어연구부 선임연구원
- 2010년 3월 ~ 현재 : 한밭대학교 정보통신공학과 부교수
- 주관심분야 : 비디오 부호화, 패턴 인식, 모바일 프로그래밍