

3차원 동영상 압축 표준의 텍스처 비디오 우선 부호화 방식을 위한 변위 벡터 추정 기법

강 제 원*

Disparity Vector Derivation Method for Texture-Video-First-Coding Modes of 3D Video Coding Standards

Je-Won Kang*

요 약

본 논문에서는 3차원 비디오 압축을 위한 표준으로서 3D-AVC와 3D-HEVC에 채택된 주위 블록의 변위 벡터 정보를 이용한 블록 기반의 변위 벡터 추정 기법 (Neighboring block-based disparity vector, NBDV)을 설명하고 성능을 평가하여 분석한다. NBDV에서는 시간적 공간적으로 인접한 주위 블록에서 부호화를 완료한 변위 움직임 벡터 (disparity motion vector)를 이용하여 변위 벡터로 변환한다. 변위 벡터는 인접 시점에서 현재 블록에 대응하는 블록을 지시하는 벡터로 시점간 통계적 연관성을 이용함으로써 3차원 동영상의 부호화 효율을 증대하는데 중요한 기능을 제공한다. 제안 기술은 저 복잡도와 동일 화질에서 약 20%의 부호화 효율 증가, 그리고 네트워크 내 미디어 게이트웨이의 효율적인 동작을 제공하는 텍스처 비디오 우선 부호화 방식을 지원하여 3차원 비디오 압축 표준 기술로 최종 채택되었다.

Key Words : High Efficiency Video Coding, 3D Video Coding, Disparity Vector

ABSTRACT

In 3D video compression, a disparity vector (DV) pointing a corresponding block position in an adjacent view is a key coding tool to exploit statistical correlation in multi-view videos. In this paper, neighboring block-based disparity vector (NBDV) is shown with detail algorithm descriptions and coding performance analysis. The proposed method derives a DV from disparity motion vector information, obtained from spatially and temporally neighboring blocks, and provides a significant coding gain about 20% BD-rate saving in a texture-video-first-coding scheme. The proposed DV derivation method is adopted into the recent 3D video coding standards such as 3D-AVC and 3D-HEVC as the state-of-the-art DV derivation method.

I. 서 론

최근 3차원 비디오의 취득과 디스플레이 기술의 발전에 따라 시청자에게 몰입감을 주는 3차원 TV 등의 실감 영상에 대한 수요가 늘어나고 있다^[1]. 기존의 3차원 입체 영상을 제공하는 기술로 서로 다른 두 위치에서

서 취득한 영상을 특수 안경을 사용하여 보여 주는 양안시점 (stereoscopic view) 기술이 있지만 시청자의 위치에 상관없이 하나의 시점만을 제공하므로 보다 실감나는 입체감을 주기에는 한계가 있다. 따라서 양안시점의 확장으로 둘 이상의 서로 다른 시점에서 동일한 피사체를 취득하는 다시점 비디오(multiview video)를

* First and Corresponding Author : Ewha Womans University Department of Electronics Engineering, jewonk@ewha.ac.kr, 정회원
 논문번호 : KICS2015-07-216, Received July 11, 2015; Revised September 15, 2014; Accepted October 13, 2015

이용한 3차원 비디오의 연구가 활발하게 연구되었다^[2-4]. 그러나 보다 자연스러운 3차원 비디오 서비스를 제공하고 무안경 방식의 3차원 디스플레이를 구현하기 위하여 수십에서 수백 시점을 제공하는 고해상도의 초다시점 비디오가 필요하다. 초다시점 비디오는 보다 좁은 시점간 간격으로 양안에 자연스러운 피사체의 시차를 제공하여 보다 넓은 입체 지역을 제공할 수 있다^[5].

다시점 비디오는 대체로 8~12 시점으로 구성하여 일반 비디오에 비하여 그 용량이 크므로 효율적인 압축 기술이 필요하다. 이러한 이유로 Moving Picture Expert Group (MPEG)에서는 기존의 H.264/AVC와 하방 호환이 되면서 다시점 비디오의 압축 효율을 증대하기 위한 다시점 비디오 압축에 관한 H.264/AVC 확장 표준화 (H.264/multiview video coding, H/264/MVC)를 진행하여 완료하였다^[6]. 일반 동영상 압축과 달리 다시점 동영상 압축은 시점간 통계적 연관성을 효율적으로 활용하는 것이 중요하다. H.264/MVC의 대표적인 부호화 기술로는 계층적 B 픽처 예측 구조 (hierarchical B prediction structure)를 이용한 시점간 변위 보상 부호화 기술 (disparity motion compensated prediction)이 있다^[7]. 그러나 H.264/MVC 압축 표준은 시점의 수가 증가함에 따라 전송 비트율도 선형 비례하여 증가하기 때문에 새로운 3차원 비디오 서비스 형태인 초다시점 비디오의 부호화에 직접 적용하기에는 어려움이 있다. 이러한 단점을 극복하기 위하여 ITU-T 산하 Video Coding Expert Group (VCEG)과 ISO/IEC 산하 MPEG의 공동 팀인 Joint Collaborative Team on 3D Video Coding (JCT-3V)에서는 기존의 다수의 컬러영상으로만 구성된 비디오 포맷 대신에 소수의 텍스처 비디오 (texture video)와 해당 시점의 보조 깊이 비디오 (depth video)를 한 쌍으로 구성하는 Multiview plus depth map (MVD) 방식을 새로운 3차원 비디오 압축 포맷으로 개발하였다^[8]. 깊이 비디오는 깊이 영상 (depth map or depth image)의 시퀀스로 영상의 화소 값은 카메라에서 피사체까지의 거리를 표현한다. MVD 방식을 이용하면 복호기 측에서는 필요로 하는 시점 보다 적은 수의 텍스처 비디오만을 수신하더라도 깊이 비디오와 깊이 영상 기반 렌더링 기술^[9]을 이용하여 각 시점 사이의 중간 시점을 가상으로 생성함으로써 초다시점 비디오를 제공할 수 있게 된다. 그림 1은 3개 시점의 텍스처 비디오와 깊이 비디오로 구성된 MVD 방식의 예시이다^[10].

JCT-3V 주도의 3차원 비디오 압축 표준(JCT-3V) 팀에서는 MVD 방식 비디오의 효율적인 압축을 위하

여 H.264/AVC와 HEVC^[11] 기반의 표준화를 진행하였다. 이 결과 MVC extension for inclusion of depth maps (MVC+D), AVC compatible video plus depth extension (3D-AVC), Multiview HEVC extension (MV-HEVC), 그리고 HEVC 3D extension (3D-HEVC)의 총 4가지 표준 트랙을 동시에 진행하였다. 이 중 MVC+D와 MV-HEVC는 기존 표준의 최소한의 변경만으로 MVD 비디오 형식을 지원하도록 설계하였다. 그러나 텍스처 비디오와 깊이 비디오 사이의 연관 정보를 압축에 이용할 수 없으므로 부호화 효율이 감소하는 단점이 있다. 따라서 보다 고효율의 압축 성능이 필요한 3차원 비디오 방송 시스템에 적용하기 위한 기술로 3D-AVC^[12]와 3D-HEVC^[13] 표준 기술이 개발되었다. 각 기술에서는 텍스처 비디오와 깊이 비디오를 효율적으로 압축하기 위하여 텍스처 비디오의 시점간 연관 정보를 활용하는 기법 이외에도 깊이 비디오의 시점간 연관 정보를 활용하는 기법과 텍스처 비디오와 깊이 비디오 사이의 연관 정보를 활용하는 기법 등의 새로운 기술들이 추가되었다. 시점간 연관정보를 이용하는 압축 기법에서는 앞서 시간 사이 연관 정보만을 사용하여 부호화를 완료한 독립 시점을 참조 영상으로 이용하여 의존 시점을 예측 부호화 한다. 예를 들어 그림 1에서 시점 3을 독립 시점으로 시점 1과 시점 5를 시점 3을 참조영상으로 이용하는 의존 시점으로 압축함을 보인다.

3차원 비디오 압축 기술에서는 그림 1에서와 같이 의존 시점 블록의 시점간 연관 정보를 이용하기 위하여 참조 시점 내 대응 위치를 찾는 것이 매우 중요하다. 변위 벡터 (disparity vector, DV)는 현재 블록에서 다른 시점의 참조 블록을 지시하는 벡터 정보로 3차원 동영상의 부호화 효율을 증대하는데 핵심 기능을 제공한다. 따라서 본 논문에서는 3D-AVC와 3D-HEVC에 표준 기술로 채택된 주위 블록의 변위 벡터 정보를 이용한 블록 기반의 변위 벡터 추정 기법 (Neighboring block-based disparity vector, NBDV)^[14,15]을 설명하고

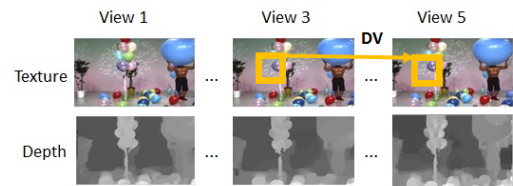


그림 1. 3개의 텍스처 비디오와 깊이 비디오로 구성된 MVD 포맷^[10]
 Fig. 1. MVD format comprising three texture videos and the corresponding depth videos^[10]

성능을 평가하여 분석한다. NBDV에서는 시간적 공간적 주위 블록의 정보를 이용하여 변위 벡터를 유도하고 현재 블록의 변위 벡터로 이용한다. NBDV는 3차원 비디오 압축 표준에서 시점간 예측을 수행하는 압축 기법을 위하여 참조 시점의 대응 블록을 찾기 위하여 사용하는 필수 기술로 저 복잡도와 높은 부호화 효율, 그리고 네트워크 내 미디어 게이트웨이의 효율적인 동작을 제공하는 텍스처 비디오 우선 부호화 방식(texture-video-first-coding, TVFC)을 지원하여 3DV 표준 기술로 최종 채택되었다.

본 논문의 구성은 II장에서 기존 변위 벡터의 추정 방식과 3차원 비디오 압축의 텍스처 비디오 우선 부호화 방식과 깊이 비디오 우선 부호화 방식의 차이를 설명하고 III장에서 NBDV의 세부 기술을 설명하며 IV장에서 성능 평가 및 분석을 수행한다. 마지막으로 V장에서 결론을 맺는다.

II. 관련 연구

2.1 3차원 비디오 압축의 기존 변위 벡터 추정 기법

3차원 비디오 압축 표준의 MVD 방식에서는 시점당 한 쌍의 텍스처 비디오와 깊이 비디오가 부호화 되므로 해당 시점의 깊이 비디오가 우선 부호화 되는 경우 복원한 깊이 정보로부터 텍스처 비디오의 변위 벡터 정보를 추정 가능하다. 취득한 텍스처 비디오는 사전에 카메라 캘리브레이션을 수행 후 정렬이 되어있다고 가정한다. 이 경우 의존 영상의 픽셀 위치 x_0 와 참조 영상의 픽셀 위치 x_1 사이의 변위 벡터 dv 는 다음과 같이 유도한다.

$$dv = x_0 - x_1 = \frac{f}{d_q} \times c \quad (1)$$

여기서 f 는 카메라의 초점 거리이고 c 는 카메라 사이 거리이며 d_q 는 양자화 잡음이 섞인 복원한 깊이 비디오의 픽셀 값으로 깊이 카메라에서 센싱한 값을 8비트 양자화 과정을 거쳐 표현한 값이다. 따라서 사전에 깊이 비디오의 부호화를 마쳐 깊이 정보가 사용 가능한 경우 해당 위치에서의 화소 값의 역에 비례하여 변위 벡터를 용이하게 추정 가능하다.

깊이 정보로부터 변위 벡터를 추정하는 기존의 연구가 있다. Rusanovskyy et al. 은 현재 부호화 블록 위치에 대응하는 깊이 영상의 블록 (예컨대 4×4 블록

또는 8×8 블록)을 선택하고 각 블록 내 최대 값을 갖는 화소를 대표 화소로 선택하여 해당 화소 값을 변위 벡터로 변환하는 방식을 제안하였다^[6]. 그러나 부호화하고자 하는 텍스처 비디오에 앞서 깊이 비디오의 압축 정보가 비트스트림에 먼저 포함되어있어야 한다. Yamamoto et al.은 시간적·공간적으로 변위 벡터의 연속성과 일관성을 유지하도록 에너지 함수를 정의하고 해당 목적 함수를 최소화 하도록 하는 깊이 영상의 화소를 선택하도록 하였다^[7]. Scharwarz et al.은 시점간 블록 매칭을 이용하여 변위 벡터를 유도한 뒤 (1) 식으로부터 의존 시점의 가상 깊이 영상 (virtual depth map)을 생성하고 이후 텍스처 비디오의 가상 깊이 영상을 워핑하여 생성하는 기법을 제안하였다^[8]. 각각의 가상 깊이 영상은 대응하는 텍스처 비디오 블록의 변위 벡터를 유도하는데 사용한다. 그러나 가상 깊이 영상의 생성에 필요한 복잡도가 증가하고 가상 깊이 영상을 추가적으로 저장하여야 하기 때문에 소요되는 메모리의 양이 급격히 증가하는 단점이 있었다. 특히 복호기 측에서 가상 깊이 영상을 생성하는 부분은 실시간 어플리케이션에 적합하지 않다.

변위 벡터를 깊이 정보로부터 추정하는 방식 이외에도 인접하는 시점의 텍스처 비디오로부터 변위 벡터를 추정하는 방식에 관한 연구가 있다. H.264/MVC 표준에서는 비디오 픽처 단위로 전역 변위 벡터를 전송한다. 추가로 16×16 크기의 매크로블록 (macroblock, MB) 단위로 보다 정교한 변위 벡터를 전역 벡터의 차분 값으로 부호화하여 전송하는 방식이 개발되었다^[9]. 그러나 증가하는 전송 정보로 부호화 효율이 떨어지는 단점이 있었다.

2.2 3차원 비디오의 텍스처 비디오 우선 부호화 방식 및 깊이 비디오 우선 부호화 방식

앞서 서술한 바와 같이 MVD 포맷은 컬러 영상과 동일 화면의 깊이 표현을 동시에 전송하는 방식이므로 한 쌍의 텍스처 비디오와 깊이 비디오를 부호화하여야 한다. 이 때 텍스처 비디오와 깊이 비디오의 부호화 순서에 따라 예측 부호화 방식이 다르게 된다. 특히 3차원 비디오 압축에 중요한 변위 벡터는 부호화 방식에 따라 다른 유도 알고리즘을 이용해야 할 것 이다.

그림 2^[10]는 3차원 비디오의 텍스처 비디오 및 깊이 비디오의 부호화를 위한 예측 구조를 보인다. 각 영상 아래 기호로 T는 텍스처 비디오를 D는 깊이 비디오를 의미하며 뒤의 숫자 (0, 1, 2)는 카메라의 시점을 표현하는 인덱스이다. 그림의 예에서 T0/D0는 시점간 예측을 수행하지 않고 시간 사이 예측 부호화만 수행하는

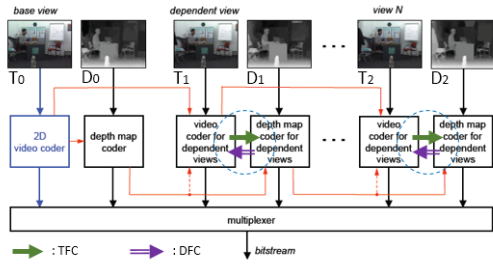


그림 2. 텍스처 비디오 우선 부호화 방식과 깊이 비디오 우선 부호화 방식^[19].
Fig. 2. Texture video first coding (TVFC) and Depth video first coding (DVFC)^[19].

독립 시점의 영상이고 T1/D1 및 T2/D2는 독립 시점 혹은 앞서 부호화한 의존 시점을 참조 영상으로 이용하여 부호화 하는 의존 시점의 영상이다. 그림2 에서 붉은색 화살표의 방향은 예측 부호화의 참조 영상을 표현한다. 그림에서 보듯이 3차원 비디오의 압축은 텍스처 비디오 우선 부호화 구조 (texture-video-first-coding, TVFC) 와 깊이 비디오 우선 부호화 구조 (depth video-first-coding, DVFC)를 이용할 수 있다. 텍스처 비디오 우선 부호화 방식은 독립 시점 및 의존 시점에서 텍스처 비디오를 먼저 부호화한다. 반대로 깊이 비디오 우선 부호화 방식은 T0-D0-D1-T1-... 순으로 독립 시점에서는 텍스처 비디오의 픽처를 먼저 부호화하고 의존 시점에서는 깊이 비디오의 픽처를 먼저 부호화한다.

텍스처 비디오 우선 부호화 방식 대비 깊이 비디오 우선 부호화 방식의 장점은 의존 시점 텍스처 비디오의 영상을 압축할 때 보다 많은 참조 영상을 사용할 수 있기 때문에 부호화 성능이 다소 증가하는 것이다. 예컨대 앞서 압축한 D1의 정보는 변위 벡터로 변환하여 T1의 압축에 사용이 가능하다. 그러나 깊이 비디오 우선 부호화 방식은 텍스처 비디오의 복호 과정에서 깊이 비디오가 반드시 필요하게 되는 단점이 있다. 깊이 비디오는 3차원 비디오 서비스를 위한 보조 영상이며 실제 컬러 영상이 사람의 시각에 인지되는 콘텐츠이다. 즉 스테레오 비디오나 다시점 비디오를 지원하는 기존 3차원 영상 기기는 실제 사용하지 않는 깊이 비디오를 추가로 전송받아야만 텍스처 비디오를 온전히 복호할 수 있으므로 깊이 비디오의 전송을 위한 네트워크 대역폭이 늘어나게 된다. 반면 텍스처 비디오 우선 부호화 방식은 미디어 게이트웨이에서 복호기 측의 요청에 따라 깊이 비디오의 해당 패킷을 버리고 텍스처 비디오에 해당하는 패킷만을 필터링함으로써 이러한 문제를 극복할 수 있다. 이러한 필터링은 비디오 압축 포맷

의 소수 플래그만 간단히 복호하여 검사하는 방식으로 가능하므로 전체 비트스트림을 복호할 필요가 없으므로 미디어 게이트웨이의 추가적인 복잡도가 크게 증가하지 않는다^[19]. 그러나 텍스처 비디오 우선 부호화 방식을 지원하기 위하여 텍스처 비디오에서 변위 벡터를 추정하는 기존의 방식은 깊이 비디오 우선 부호화 방식의 변위 벡터 대비 정확하지 않거나 변위 벡터 전송 정보의 증가로 전체적인 부호화 효율이 크게 감소하게 된다. 초기 3D-HEVC 및 3D-AVC 표준 단계에서는 텍스처 비디오 우선 부호화 방식 대비 깊이 비디오 우선 부호화 방식은 약 20%의 BD-rate 감소 효율을 제공하였다.

III. Neighboring Block-based Disparity Vector

본 장에서는 텍스처 비디오 기반의 변위벡터 추정 기법으로서 이웃 블록의 변위 벡터 정보를 이용하여 현재 블록의 변위 벡터를 추정하는 방식인 Neighboring block-based disparity vector (NBDV)를 설명한다. NBDV는 기존의 비디오 압축 신택스를 이용하여 변위 벡터 추정을 수행하기 때문에 별도의 비트율 증가가 없고 계산 량이 적어 현재 3차원 비디오 압축 표준 기술로 채택되었다^[10,14,15]. NBDV는 텍스처 비디오만을 이용하여 변위 벡터를 추정하므로 텍스처 비디오 우선 부호화 방식을 지원한다.

NBDV에서는 이전에 부호화가 완료된 공간상에 인접 하는 블록 그리고 시간상에 인접하는 블록의 움직임 벡터 정보를 이용하여 변위 벡터를 추정 한다. 3차원 동영상 압축에서는 두 가지 형태의 움직임 벡터, 즉 기존의 시간 상 예측을 수행하는 시간 움직임 벡터 (temporal motion vector, TMV)와 시점간 예측을 수행하는 변위 움직임 벡터 (disparity motion vector, DMV)가 있다. NBDV는 주위 블록에서 변위 움직임 벡터를 찾기 위하여 시간 및 공간 이웃 블록을 탐색한다. 시간 움직임 벡터와 변위 움직임 벡터는 (x, y) 벡터 양 이외에도 참조하는 영상이 속한 참조영상 리스트와 참조영상의 인덱스가 부가 정보로 부호화된다. 따라서 시간 움직임 벡터와 변위 움직임 벡터의 구분은 참조영상 리스트와 인덱스로부터 추출한 카메라 시점을 이용한다. 즉 움직임 벡터의 시점 번호가 현재 픽처의 번호와 상이하면 변위 움직임 벡터로 인식하고 동일하면 시간 움직임 벡터로 인식한다. 탐색하여 얻은 주위 블록의 시점 움직임 벡터는 바로 현재 블록의 변위 벡터로 사용하거나 둘 이상의 시점 움직임 벡터를

필터링하여 변위 벡터를 생성할 수 있다^[20]. 3DV 표준에서는 변위 움직임 벡터를 필터링 없이 바로 변위 벡터로 변환하는 방식이 최종 선택되었다.

NBDV의 변위 움직임 벡터 탐색을 위한 주위 블록 유닛의 위치 및 순서는 블록 기반의 비디오 압축 표준에 쉽게 적응적으로 사용이 가능하다. HEVC의 경우 블록 유닛은 코딩 유닛 (coding unit, CU)이다. NBDV는 시점간 예측 부호화에 주로 사용하므로 예측 유닛 (prediction unit, PU) 별로 적용이 가능하지만 복잡도를 고려하여 CU 단위로 통합되었다^[21]. 그림 3은 HEVC의 CU에서 제공하는 총 5개의 공간 이웃 블록 (붉은색)과 총 2개의 시간 이웃 블록 (푸른색)을 보인다. NBDV는 이 중에서 녹색으로 표현한 좌측과 상측의 공간 이웃 블록과 가운데의 시간 이웃 블록을 조사한다. 좌, 상측의 공간 이웃 블록과 가운데의 시간 이웃 블록은 HEVC의 머지 모드 및 움직임 예측 모드에 사용하는 총 5개의 공간 이웃 블록과 2개의 시간 이웃 블록과 중첩이 되도록 설계하여 하드웨어에 추가되는 메모리의 양을 최소화 하도록 설계하였다. 3D-AVC에서는 블록 유닛으로서 매크로블록을 사용하므로 매크로블록의 공간적으로 좌측과 상측, 그리고 우 상측의 주위 블록을 이용하며 하단의 시간적 이웃 블록을 사용한다.

NBDV 알고리즘에서 변위 움직임 벡터의 탐색은 먼저 공간상에 인접한 블록의 탐색을 순차적으로 수행하며 변위 움직임 벡터가 발견되면 해당 변위 움직임 벡터를 변위 벡터로 변환하고 그렇지 않으면 다음 블록의 위치로 이동하여 탐색을 계속 진행한다. 만약 공간 상의 블록 위치에서 찾지 못한 경우에는 시간상에 인접한 블록의 위치로 이동하여 탐색한다. 본 과정을 그림 4의 플로우차트에 보였다.

다음으로 시간상의 인접 블록을 탐색하기 위해서는 그림 5에서 보듯이 먼저 블록이 위치한 시간상의 참조

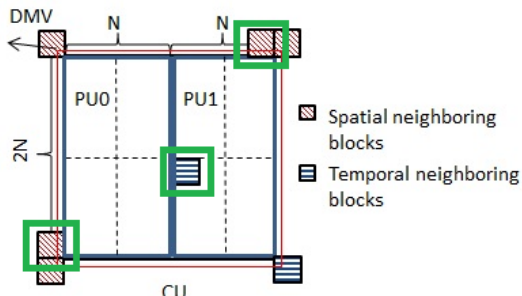


그림 3. NBDV의 공간/시간 이웃 블록의 위치^[10]
Fig. 3. spatial and temporal neighboring blocks used for NBDV^[10]

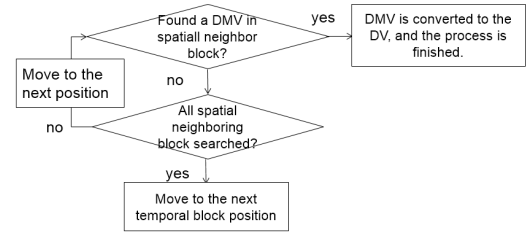


그림 4. 공간 상에 인접한 블록의 변위 벡터 탐색 과정
Fig. 4. DV search in spatially neighboring blocks.

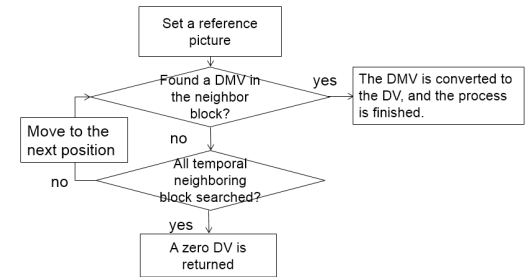


그림 5. 시간상에 인접한 블록의 변위 벡터 탐색 과정
Fig. 5. DV search in temporally neighboring blocks.

영상을 선택하여야 한다. 참조 영상을 정하는 방식은 우선 임의의 접근 픽처 (random access picture, RAP)를 선택하고 픽처의 시간 번호 (temporal id)가 낮은 순서로 탐색을 진행한다. 임의 접근 픽처의 경우 선택 가능한 부호화 모드가 시점간 예측 부호화 모드 혹은 화면내 예측 부호화 모드이기 때문에 보다 높은 확률로 변위 움직임 벡터를 찾을 수 있기 때문이다. 탐색 참조 영상을 선택한 이후에는 공간상의 이웃 블록과 유사하게 사전에 정의한 블록 위치 (예컨대 그림 3의 가운데 블록과 우하단 블록)에서 움직임 벡터의 변위 움직임 벡터/시간 움직임 벡터 여부를 판별하고 변위 움직임 벡터를 찾는 경우 변위 벡터로 변환한다. 탐색 블록의 위치와 순서는 여러 방식이 제안되었지만 복잡도와 성능의 균형을 고려하여 기본 코덱 대비 최소한의 메모리를 소요하면서 효율적인 부호화 성능을 제공하도록 실험적으로 결정하였다.

공간 및 시간상의 인접 블록에서 변위 움직임 벡터의 탐색을 실패하여 NBDV의 유도가 가능하지 않은 경우에는 영 벡터를 출력하지만 이 경우 변위 벡터의 부재로 해당 블록의 부호화 효율은 저하될 수 있다. 따라서 영 벡터의 수정을 위해서 여러 보조 기법이 추가되었다. 첫째로 유도 변위 벡터 (derived disparity vector, DDV)에서는 픽처 단위로 보조 변위 벡터를 전송한다^[22]. 유도 변위 벡터는 픽처의 부호화 시작마다

영 벡터로 초기화되고 픽처 내부의 매 CU가 부호화될 때마다 해당 변위 벡터로 업데이트 된다. NBDV가 영 벡터인 경우 DDV로 대체될 수 있다. 둘째로 깊이 영상 기반의 NBDV 수정 벡터 (depth-oriented NBDV, DoNBDV)를 사용한다^[23]. DDV와 DoNBDV는 약 0.5%의 비트율 감소 효율을 제공한다.

3DV 압축 표준에서 주로 사용하는 NBDV의 적용 예로는 시점간 움직임 예측 기법^[24], 잔차 신호 예측 기법^[25] 변위 움직임 벡터의 예측 보상 기법이 있다. 보다 자세한 내용은 3D-AVC^[12] 및 3D-HEVC^[13]의 표준 기술문서를 참고할 수 있다.

IV. 실험

본 장에서는 NBDV의 성능평가를 위하여 3D-AVC 참조 압축 코덱인 ATM software을 이용하여 실험을 하였다. 실험은 JCT-3DV의 공통 실험 조건에서 진행하였고 3대의 카메라 구성을 이용하여 1개의 독립 시점 비디오와 2개의 의존 시점 비디오를 구성하여 압축을 하였다. 실험 영상은 JCT-3V에서 표준화에 사용하는 3차원 동영상을 사용하였다. 보다 구체적으로 "Poznan Hall", "Poznan Street", "Undo Dancer", "GT fly", "Kendo", "Ballons", 그리고 "Newspaper"의 총 7

시퀀스 (각각 S01-S07로 표현)를 사용하였다^[26]. 각 시퀀스의 특성 및 취득에 사용한 카메라의 3차원 배열을 표 1에 요약하여 정리하였다. 표에서 보듯이 각 테스트 시퀀스는 다양한 어플리케이션 및 3차원 대상을 고려하여 생성하였다. 예컨대 S03과 S04는 컴퓨터에서 생성한 3차원 시퀀스로 원 깊이 정보를 활용할 수 있는 반면 나머지 시퀀스는 3차원 대상을 깊이 카메라로부터 취득하여 비교적 많은 잡음이 포함된 깊이 비디오를 포함한다. 또한 S01, S05, 그리고 S06은 다시점 비디오가 움직임을 가지고 대상을 추적하는 시퀀스이다. 또한 각 시퀀스는 카메라의 개수 및 카메라간 거리가 서로 다른 3차원 카메라 배열을 이용하여 취득하였다.

일반적인 동영상의 압축 효율을 평가하는 방식은 동일 화질 대비 감소한 비트율을 표현하는 BD-rate 감소 혹은 등가적으로 같은 비트율에서 화질 (PSNR)의 향상도를 평가하는 BD-PSNR을 사용한다. 3차원 동영상 압축의 부호화 효율 평가는 크게 텍스처 비디오의 화질과 비트율만 표현하는 방법인 "텍스처 비디오 부호화" (texture coding), 텍스처 비디오의 화질과 깊이 비디오 및 텍스처 비디오의 비트율을 표현하는 방법, "전체 비디오 부호화" (total coded PSNR), 그리고 복호한 깊이 비디오를 이용하여 중간 시점을 가상으로 생

표 1. Description of Test Sequence^[26]
Table 1. Description of Test Sequence^[26]

Seq. ID	Seq. Name	Resolution (width×height)	Camera Arrangements	Etc
S01	Poznan Hall2	1920×1088	9 cams with 13.8cm space	Moving cams
S02	Posnan Street	1920×1088	9 cams with 13.8cm space	Static cams
S03	Undo Dancer	1920×1088	Computer generated	Ground truth depth
S04	GT Fly	1920×1088	Computer generated	Ground truth depth
S05	Kendo	1024×768	7 cams with 5.0cm space	Moving cams
S06	Ballons	1024×768	7 cams with 5.0cm space	Moving cams
S07	Newspaper	1024×768	9 cams with 5.0cm space	Static cams

표 2. Coding gain in 3D-AVC, Texture-Video-First-Coding (NBDV Off) VS MVC+D
Table 2. Coding gain in 3D-AVC, Texture-Video-First-Coding (NBDV Off) VS MVC+D

	Texture Coding (BD-rate saving,%)	Depth Coding (BD-rate saving,%)	Total (Coded PSNR) (BD-rate saving,%)	Total (Synthesized) (BD-rate saving,%)
S01	-1.3%	+2.3%	-5.9%	-8.1%
S02	-1.2%	+11.8%	-3.2%	-7.9%
S03	-0.2%	+14.1%	-0.9%	-2.1%
S04	-0.1%	-14.8%	-1.8%	-2.7%
S05	-2.1%	+13.1%	-12.2%	-12.1%
S06	-2.3%	+6.9%	-11.3%	-11.1%
S07	-1.7%	-22.1%	-9.4%	-7.6%
Average	-0.8%	+1.6%	-6.4%	-7.3%

표 3. Coding gain in 3D-AVC, Texture-Video-First-Coding (NBDV On) VS MVC+D
 Table 3. Coding gain in 3D-AVC, Texture-Video-First-Coding (NBDV Off) VS MVC+D

	Texture Coding (BD-rate saving,%)	Depth Coding (BD-rate saving,%)	Total (Coded PSNR) (BD-rate saving,%)	Total (Synthesized) (BD-rate saving,%)
S01	-30.6%	-1.3%	-28.3%	-25.8%
S02	-12.3%	-2.1%	-11.7%	-9.9%
S03	-17.7%	+2.8%	-16.5%	-14.0%
S04	-23.3%	-26.5%	-21.3%	-21.0%
S05	-22.0%	-1.6%	-18.8%	-17.8%
S06	-26.6%	-2.9%	-24.5%	-21.9%
S07	-14.5%	10.8%	-18.6%	-12.1%
Average	-21.0%	-3.0%	-19.9%	-17.5%

표 4. 3D-AVC, Texture-Video-First-Coding (NBDV) VS 3D-AVC, Depth-Video-First-Coding.
 Table 4. 3D-AVC, Texture-Video-First-Coding (NBDV) VS 3D-AVC, Depth-Video-First-Coding.

	Texture Coding (BD-rate saving,%)	Depth Coding (BD-rate saving,%)	Total (Coded PSNR) (BD-rate saving,%)	Total (Synthesized) (BD-rate saving,%)
S01	0.4%	0.4%	0.3%	0.1%
S02	0.2%	-0.9%	-0.1%	0.0%
S03	0.4%	3.5%	0.6%	0.3%
S04	0.4%	-0.1%	0.4%	0.3%
S05	1.5%	-0.8%	1.2%	0.9%
S06	1.0%	-1.5%	0.6%	0.7%
S07	0.7%	-0.9%	0.4%	0.4%
Average	0.7%	-0.1%	0.5%	0.4%

성한 뒤 가상 시점의 화질과 텍스처 비디오 및 깊이 비디오의 비트율을 평가하는 방법 “전체 합성 비디오 부호화”(total, synthesized PSNR)가 있다. 본 논문에서는 3가지 방식을 모두 평가한다. 깊이 비디오의 부호화 효율은 단독으로 평가하지 않는다. 3차원 동영상 참조 코덱에는 NBDV가 기본적으로 활성화되어있기 때문에 실험에서는 NBDV의 기능을 비활성화 한 후 성능의 향상 여부를 보였다.

NBDV의 부호화 효율을 평가하기 위하여 다음과 같이 단계적으로 실험을 진행하였다. 먼저, MVC+D 대비 NBDV를 사용하지 않은 3D-AVC의 성능을 평가한다. 다음으로 MVC+D 대비 NBDV를 사용한 3D-AVC의 성능을 평가하여 NBDV의 3차원 비디오 부호화 효율을 명확하게 보인다. 다음으로 NBDV를 사용한 텍스처 비디오 우선 부호화 방식 대비 깊이 정보로부터 변위 벡터를 유도하는 깊이 비디오 우선 부호화 방식의 성능을 보인다. 각 결과 및 분석을 표 2, 표3, 그리고 표 4에 따라 보인다.

표 2는 3D-AVC의 NBDV를 사용하지 않는 조건에서 텍스처 비디오 우선 부호화 방식의 부호화 효율과 MVC+D의 부호화 효율을 비교한 것이다. 표 3은 3D-AVC의 NBDV를 이용한 텍스처 비디오 우선 부호

화 방식과 MVC+D와의 부호화 효율을 비교한 것이다. 따라서 두 표의 성능 차이는 NBDV의 사용 여부에 따른 것이다. 표 2와 표 3의 숫자는 동일화질 대비 비트율 감소 (BD-rate saving)로 음수 기호는 비트율의 감소로 부호화 효율이 증가한 것을 의미한다. 3D-AVC의 NBDV를 이용한 텍스처 비디오 우선 부호화 방식은 MVC+D 대비 약 “텍스처 비디오 부호화” 에서 약 21%의 부호화 효율이 증가하였고 “전체 합성 비디오 부호화”에서는 약 17.5%의 부호화 효율이 증가한 것을 알 수 있다. 깊이 비디오에서는 약 3%의 효율 증가를 보였지만 실제 사람 눈에 인식되는 부분은 텍스처 비디오이기 때문에 텍스처 비디오의 부호화 효율이 더욱 중요하다고 할 수 있다. 반면 NBDV를 사용하지 않은 3D-AVC의 기존 텍스처 비디오 우선 부호화 방식은 MVC+D 대비 약 1.25%의 부호화 효율로 NBDV가 약 19.75%의 부호화 효율을 제공하는 것을 알 수 있다. 그림 6은 복원한 텍스처 비디오의 주관적 화질 비교로 MVC+D (좌) 대비 3D-AVC, 텍스처 비디오 우선 부호화 방식 (우)가 더 시각적으로 좋은 화질을 제공하는 것을 볼 수 있다.

표 4는 3D-AVC의 텍스처 비디오 우선 부호화 방식과 깊이 비디오 우선 부호화 방식의 성능 비교이다. 표

에서 보듯이 두 부호화 방식의 차이는 0.7% 이내로 깊이 비디오 우선 부호화 방식이 텍스처 비디오 우선 부호화 방식에 비하여 근소하게 성능이 앞선다고 할 수 있다. 그러나 앞서 서술한 바와 같이 깊이 비디오 우선 부호화 방식은 미디어 게이트웨이의 깊이 비디오에 해당하는 비트스트림을 추출하는 경우 텍스처-깊이 영상 간의 의존으로 미디어 게이트웨이의 복잡도와 복호기 측의 복잡도가 증가하며 전체적인 네트워크의 대역폭을 보다 많이 필요로 하는 단점이 있다. 반면 텍스처 비디오 우선 부호화 방식은 약간의 부호화 효율 감소가 있었지만 전체적인 시스템 관점에서 3차원 비디오 서비스에 보다 적합한 특징을 제공한다.

다음으로 표 5는 3D-AVC, 텍스처 비디오 우선 부호화 방식의 복잡도 분석으로 하드웨어 구현 시 필요한 상대적인 메모리의 소요 증가량과 부호화 시간을 측정하여 보인다. 먼저 추가적인 메모리 소요량으로는 비디오 압축에서 메모리 소요량이 가장 큰 움직임 보상의 경우 최악의 경우에서 8x4와 4x8 두 블록의 서브블록에 대해서 각각 B 예측으로 움직임 보상을 수행하는 경우 약 117개의 화소를 메모리에 로드하여야 하고 2개의 참조 리스트를 사용해야 하며 8x4 혹은 4x8의 블록은 16x16 매크로블록에 총 8개의 블록이 포함되므로 매크로블록 당 총 1,872 바이트가 필요하다. 반면에 NBDV는 공간적 이웃 블록의 경우 H.264/AVC에서 이미 움직임 벡터를 체크하기 위하여 메모리에 로드하므로 추가적인 소요량이 필요 없고 총 2장의 참조영상에 대하여 2개의 시간적 이웃 블록이 추가적으로

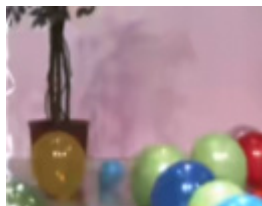
표 5. NBDV complexity analysis (TVFC: Texture-Video-First-Coding and DVFC: Depth-Video-First-Coding)
Table 5. NBDV complexity analysis (TVFC: Texture-Video-First-Coding and DVFC: Depth-Video-First-Coding)

Test configuration	Anchor	Compared	Increment (%)
Additional memory Access in the worst case, 3-View Scenario	H.264/AVC	3D-AVC (DVFC)	0.2%
		3D-AVC (TVFC)	0.2%
Additional encoding time, 3-View Scenario	3D-AVC TVFC, NBDV Off	3D-AVC (TVFC), NBDV On	5%

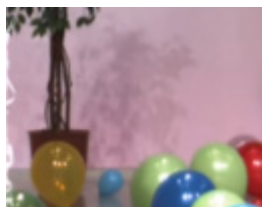
필요하므로 4 바이트가 필요하다. 이는 전체 움직임 보상 과정에서 약 0.2%에 해당하는 추가적인 소요량으로 NBDV의 메모리 소요량은 매우 적다고 간주할 수 있다. 깊이 영상 우선 부호화 방식의 경우에도 깊이 영상을 접근할 때 각 매크로블록 모서리의 총 4개 화소만을 접근하여 그 중 최대 값을 선택하여 매크로블록의 변위 벡터로 변환하므로 매크로블록 당 4바이트가 필요하다. 즉 마찬가지로 0.2%의 추가 메모리가 필요하다. 부호화 시간을 측정하면 텍스처 비디오 우선 부호화 방식은 NBDV의 on/off에 따라 약 5%의 차이가 있음을 알 수 있다.

V. 결론

본 논문에서는 차세대 3차원 비디오 압축 표준 기술로서 텍스처 비디오 우선 부호화 방식을 위한 주위 블록의 변위 벡터 정보를 이용한 블록 기반의 변위 벡터 추정 기법 (NBDV)을 제안하였다. 제안 기법에서는 시간적으로 그리고 공간적으로 인접한 주위 블록에서 부호화를 완료한 변위 움직임 벡터를 이용하여 변위 벡터로 변환함으로써 인접 시점간 통계적 연관성을 부호화에 이용하는 기능을 제공하여 동일 화질 대비 높은 부호화 효율을 제공하였다. 뿐만 아니라 텍스처 비디오의 정보만으로 변위 벡터를 추출함으로써 전체 3차원 비디오의 부호가 깊이 비디오의 부호에 의존하지 않아 미디어 게이트웨이 등에서 적은 대역폭만으로 3차원 비디오의 비트스트림을 전송할 수 있었다. 실험에 의하면 제안 기법은 기존 기법인 MVC+D 대비 “텍스처 비



(a)



(b)

그림 6. 주관적 화질 비교 : (a) 3D-AVC, 텍스처 비디오 우선 부호화 와 (b) MVC+D
Fig. 6. Subjective visual comparison : (a) 3D-AVC, Texture-First-Coding and (b) MVC+D

디오 부호화 성능 평가에서 약 21%, 전체 합성 비디오 부호화 성능 평가에서 약 17.5%의 부호화 효율을 제공하였다. 또한 제안 방식은 기본 코덱 대비 0.2%의 메모리 소요 및 5%의 부호화 측정 시간 증가의 적은 하드웨어 복잡도의 증가만으로도 구현이 가능하여 저 복잡도 높은 부호화 효율의 3차원 비디오 압축 기술을 제공하였다. 본 기술을 응용한 관련 어플리케이션으로는 3차원 비디오 영화, 방송 등 엔터테인먼트 뿐만 아니라 군용 의료용으로써 실감 미디어에 널리 사용될 수 있다.

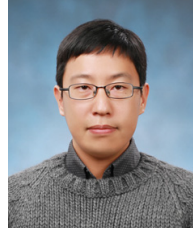
References

- [1] A. Smolic, K. Mueller, P. Merkle, P. Kauff, and T. Wiegand, "An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution," in *Proc. PCS 2009*, pp. 1-4, Chicago, IL, 2009.
- [2] G. Saygili, C. G. Gurler, and A. M. Tekalp, "Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3D video streaming," *IEEE Trans. Broadcasting*, vol. 57, no. 2, pp. 593-601, 2011.
- [3] P. Merkle, H. Brust, K. Dix, K. Müller, and T. Wiegand, "Stereo video compression for mobile 3D services," in *Proc. 3D TV Conf.*, 2009.
- [4] M. Tanimoto, "FTV standardization in MPEG," *3DTV-CON*, pp. 1-4, Budapest, Jul. 2014.
- [5] P. Carballeira, J. Gutierrez, F. Moran, J. Cabrera, and N. Garcia, "Subjective evaluation of super multiview video in consumer 3D displays," *QoMEX*, 2015.
- [6] *MVC Reference Software*, document N10897 MPEG, Dec. 2009.
- [7] P. Merkle, K. Müller, A. Smolic, and T. Weigand, "Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC," *ICME*, pp. 1717-1720, Toronto, Ont, Jul. 2006.
- [8] *Multi-View Video Plus Depth (MVD) Format for Advanced 3D Video Systems*, Document JVT-W100 Joint Video Team, Apr. 2007.
- [9] C. Fehn, "Depth-Image-Based Rendering (DIBR), compression and transmission for a new approach on 3D-TV," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI*, vol. 5291, May 2004.
- [10] J.-W. Kang, Y. Chen, Li. Zhang, and M. Karczewicz, "Low complexity neighboring block based disparity vector derivation in 3D-HEVC," in *Proc. ISCAS*, pp. 1921-1924, Melbourne, Jun. 2014.
- [11] *High Efficiency Video Coding*, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.
- [12] M. M. Hannuksela, Y. Chen, T. Suzuki, J.-R. Ohm, and G. Sullivan, *3D-AVC Draft Text 8*, in JCT-3V Document JCT3V-F1002, Nov. 2013.
- [13] G. Tech, K. Wegner, Y. Chen, and S. Yea, *3D-HEVC Draft Text 2*, in JCT-3V Document JCT3V-F1001, Nov. 2013.
- [14] J.-W. Kang, Y. Chen, L. Zhang, X. Zhao, and M. Karczewicz, "Neighboring block based disparity vector derivation for multiview compatible 3D-AVC," in *Proc. SPIE Electron.*, vol. 8856, San Diego, California, Aug. 2013.
- [15] J.-W. Kang, "Disparity vector derivation for texture first coding mode in 3D video coding," *IPIU Workshop*, 2015.
- [16] D. Rusanovskyy, M. M. Hannuksela, and W. Su, *Depth-based coding of MVD data for 3D video extension of H. 264/AVC*, 3D Research, Springer, Jun. 2013.
- [17] K. Yamamoto, M. Kitahara, H. Kimata, T. Yendo, T. Fujii, M. Tanimoto, S. Shimizu, K. Kamikura, and Y. Yashima, "Multiview video coding using view interpolation and color correction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1436-1449, Nov. 2007.
- [18] H. Schwarz and T. Wiegand, "Inter-view prediction of motion data in multiview video coding," in *Picture Coding Symp.*, pp. 101-104, Kraków, Poland, May 2012.
- [19] *Test Model 7 of 3D-HEVC and MV-HEVC*, ITU-T SG16 WP3 ISO/IEC JTC1/SC29/WG11, Doc. JCT3V-G1005.
- [20] J. Kang, Y. Chen, L. Zhang, and M. Karczewicz, *3D-CE2.h related: Enhanced*

disparity vector derivation, in JCT-3V Document JCT3V-C0050, Jan. 2013.

- [21] Kang, Y. Chen, L. Zhang, and M. Karczewicz, *CE2.h: CU-based Disparity Vector Derivation in 3D-HEVC*, in JCT-3V Document JCT3V-D0181, Mar. 2013.
- [22] L. Zhang, Y. Chen, and M. Karczewicz, *CE2.h related: Derived disparity vector for 3D-HEVC*, in JCT-3V, Document JCT3V-D0194, Apr. 2013.
- [23] Y.-L. Chang, C.-L. Wu, Y.-P. Tsai, and S. Lei, *3D-CE1.h: Depth-oriented neighboring block disparity vector (DoNBDV) with virtual depth retrieval*, in JCT-3V, Document JCT3V-C0131, Oct. 2012.
- [24] L. Zhang, Y. Chen, V. Thirumalai, J.-L. Lin, Y.-W. Chen, J. An, S. Lei, L. Guillo, T. Guionnet, and C. Guillemot, "Inter-view motion prediction in 3D-HEVC," in *Proc. ISCAS*, Melbourne, Australia, Jun. 2014.
- [25] X. Li, L. Zhang, and Y. Chen, "Advanced Residual Prediction in 3D-HEVC," in *Proc Int. Conf. Image Process.*, Melbourne, Australia, pp. 1747-1751, Sept. 2013.
- [26] D. Rusanovskyy, K. Müller, and A. Vetro, *Common Test Conditions of 3DV Core Experiments*, in JCT-3V, Document JCT3V-C1100, Jan. 2013.

강 제 원 (Je-Won Kang)



2006년 2월 : 서울대학교 전기
컴퓨터 공학부 졸업
2012년 8월 : Univ. of Southern
California 박사 졸업.
2014년 2월 : Qualcomm, Inc.
2014년 3월~현재 : 이화여자대
학교 조교수

<관심분야> 영상 동영상 압축, 기계학습, 컴퓨터 비
전