일반논문 (Regular Paper)
방송공학회논문지 제22권 제2호, 2017년 3월 (JBE Vol. 22, No. 2, March 2017)
https://doi.org/10.5909/JBE.2017.22.2.240
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

MV-HEVC에서 빠른 변위 움직임 벡터 탐색 방법

이 재 영[a)], 한 종 기[a)‡]

Fast Disparity Motion Vector Searching Method for the MV-HEVC

Jae-Yung Lee[a)] and Jong-Ki Han[a)‡]

요 약

기존의 High Efficiency Video Coding(HEVC)기반의 다시점 비디오 부호화 기법은 종속 시점의 픽쳐를 부호화 할 때, disparity compensation prediction(DCP)을 위해 추가적인 참조 영상을 사용하기 때문에 높은 부호화 복잡도를 가지고 있다. 본 논문에서는 MV-HEVC에서 변위 움직임 벡터 탐색 복잡도를 줄이기 위한 빠른 변위 움직임 벡터 탐색 방법을 제안한다. 제안하는 방법은 어파인 변환을 이용한 초기 탐색 지점 결정 방법과 적응적인 탐색 영역 크기 결정 방법을 포함한다. 실험 결과는 제안하는 방법이 부호화 효율의 저하가 거의 없이 변위 움직임 탐색 복잡도를 90.78%까지 줄일 수 있는 것을 보여준다. 또한, 실험 결과는 제안하는 방법이 다른 복잡도 감소 기술들 보다 효과적으로 복잡도를 줄이는 것을 보여준다.

Abstract

Multi-view video codec based on the High Efficiency Video Coding (MV-HEVC) has high encoding complexity because it exploits an additional reference picture for disparity compensation prediction (DCP) when the picture of dependent view is encoded. In this paper, we propose an efficient method to reduce the complexity of disparity motion vector search for the MV-HEVC. The proposed method includes the initial search point decision method using affine transform and the adaptive search range decision method. The simulation results show that the proposed method reduces the complexity of disparity motion vector search up to 90.78% with negligible coding efficiency degradation. Also the results show that the proposed method outperforms other conventional techniques reducing complexity.

Keyword : Multi-view HEVC (MV-HEVC), Disparity Searching, Multiview Video Coding (MVC), Search Range, Affine Transform

a) 세종대학교 전자정보통신공학과(Sejong University)
‡ Corresponding Author : 한종기(Jong-Ki Han)
E-mail: hjk@sejong.edu
Tel: +82-3408-3739
ORCID: http://orcid.org/0000-0002-5036-7199
※ 이 논문은 2015년도 미래창조과학부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2015R1A2A2A01006193).
· Manuscript received February 3, 2017; Revised March 15, 2017; Accepted March 17, 2017.

Ⅰ. 서 론

초고해상도와 실감형 미디어에 대한 요구가 증가하면서 지난 2013년 1월 VCEG(Video Coding Expert Group)와 MPEG(Moving Picture Expert Group)이 구성한 Joint Collaborative Team on Video Coding (JCT-VC)에서 초고

Copyright ⓒ 2017 Korean Institute of Broadcast and Media Engineers. All rights reserved.
"This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (http://creativecommons.org/licenses/by-nc-nd/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered."

해상도 동영상 압축을 위한 High Efficiency Video Coding (HEVC)[1][2] 표준화 완료하였다. 초고해상도 동영상뿐만 아니라 고해상도 다시점 영상 서비스의 수요가 증가함에 따라 VCEG과 MPEG은 Joint Collaborative Team on 3D Video(JCT-3V)를 구성하여 HEVC의 확장 표준인 MV-HEVC[3][4]를 2014년 10월에 표준화 완료하였으며, 블록 단위 부호화 알고리즘에 대한 확장을 허용하고 깊이 정보의 부호화까지 포함하는 3D-HEVC[5][6]를 2015년 2월에 표준화 완료하였다.

MV-HEVC는 블록 단위 부호화 알고리즘의 추가 및 변경 없이 Disparity Compensation Prediction(DCP)와 관련된 하이레벨 신택스를 추가하는 프레임 호환 방식으로 설계되어 있다. 프레임 호환 방식은 블록 단위 부호화 알고리즘을 변경하지 않기 때문에 다시점 영상에 대해 극대화된 부호화 효율을 기대할 수 없지만 HEVC 코덱의 기존 설계를 변경하지 않고 다시점 영상을 지원하기 때문에 구현 복잡도가 낮다. 반면, 3D-HEVC는 블록 단위 부호화 알고리즘에 대해 확장을 허용하고 깊이 정보를 부호화 할 수 있기 때문에 다시점 영상에 대해 효과적인 부호화 효율을 기대할 수 있다. 그러나 3D-HEVC는 HEVC 코덱의 설계를 변경하였기 때문에 구현을 위해서는 코덱을 재설계하여야 한다.

HEVC 코덱을 사용하여 다시점 영상의 각 시점을 독립적으로 부호화할 때, 발생하는 부호화 복잡도는 MV-HEVC를 이용하여 다시점 영상의 각 시점을 부호화 할 때, 발생하는 부호화 복잡도 보다 낮다. HEVC는 화면 간 예측을 위해 Motion Compensation Prediction(MCP)만을 사용하지만, MV-HEVC는 화면 간 예측을 위해 MCP와 DCP를 모두 사용 할 수 있다. 이때, DCP는 종속 시점의 영상을 부/복호화 할 때, 독립 시점의 복호된 영상을 참조하여 시점 간 예측을 수행하는 것을 의미한다. 앞서 설명하였듯이 MV-HEVC는 프레임 호환 방식으로 HEVC에서 확장 설계되었기 때문에 높은 부호화 복잡도를 가지게 된다. 이러한 문제를 해결하기 위해 많은 연구[7][8][9]들이 진행되어 왔다. [7]은 참조 시점의 영상에서의 변위 움직임 벡터 분포를 이용하여 적응적으로 변위 움직임 벡터 탐색 영역의 크기를 결정하는 방법을 사용하여 빠른 변위 움직임 벡터 탐색 방법을 제안하고 있으며, [8]은 참조 시점의 영상에서 현재 블록과 동일 위치의 블록의 예측 모드 및 분할 모드와 계층적 부호화 구조에서 Temporal layer 계층에 따라 변위 움직임 벡터 탐색 영역의 크기를 결정하는 방법과 현재 픽쳐 이전에 복호된 동일 시점의 영상의 변위 움직임 벡터를 사용하여 초기 탐색 지점을 결정하는 방법을 사용하여 빠른 변위 움직임 벡터 탐색 방법을 제안하였다.

본 논문에서는 MV-HEVC에서 변위 움직임 벡터 탐색 복잡도를 줄일 수 있는 빠른 변위 움직임 벡터 탐색 방법을 제안한다. 제안하는 방법에서는 어파인 변환을 이용하여 두 시점 간 변위 움직임 벡터의 관계를 구하고 이를 이용하여 초기 변위 움직임 벡터 탐색 지점을 결정하는 방법과 RAP(Random Access Point) 픽쳐 위치의 변위 움직임 벡터들과 어파인 변환을 통해 결정된 초기 변위 움직임 벡터 탐색 지점을 이용하여 변위 움직임 벡터 탐색 영역의 크기를 결정하는 방법을 설명한다. 본 논문의 구성은 다음과 같이 구성된다. 먼저, Ⅱ장에서는 MV-HEVC에서 DCP를 위해 변위 움직임 벡터를 탐색하는 과정과 변위 움직임 벡터 탐색 복잡도를 분석한다. 그리고 Ⅲ장에서는 MV-HEVC에서 빠른 변위 움직임 벡터 탐색을 위해 본 논문에 제안하는 방법에 대해 설명하고, Ⅳ장에서는 실험 결과를 통하여 제안된 방법이 기존 MV-HEVC와 동일한 성능을 유지하면서 변위 움직임 탐색 복잡도를 줄일 수 있음을 보여준다. 마지막으로 Ⅴ장에서는 결론을 기술한다.

Ⅱ. HEVC기반 다시점 영상 부호화

앞서 언급하였듯이, MV-HEVC는 HEVC와 블록 단위에서 동일한 압축 알고리즘을 사용하며, HEVC에서 다시점 영상 부호화 지원을 위한 하이레벨 신택스를 추가한 압축 표준이다. 그러므로 MV-HEVC에서 2개 시점에 대한 다시점 영상 부호화를 지원하는 방법은 그림 1과 같다. 그림 1과 같이 2개의 시점에 대해 다시점 영상 부호화를 적용하는 경우, 독립 시점(View 0)과 종속 시점(View 1)으로 나뉠 수 있으며, 독립 시점의 비디오 데이터는 HEVC 코덱을 이용하여 부호화 한다. 반면, 종속시점의 비디오 데이터는 독립 시점의 비디오 데이터를 참조하여 부호화 한다. 종속 시점에서 독립 시점의 영상을 참조하는 방법은 종속 시점의

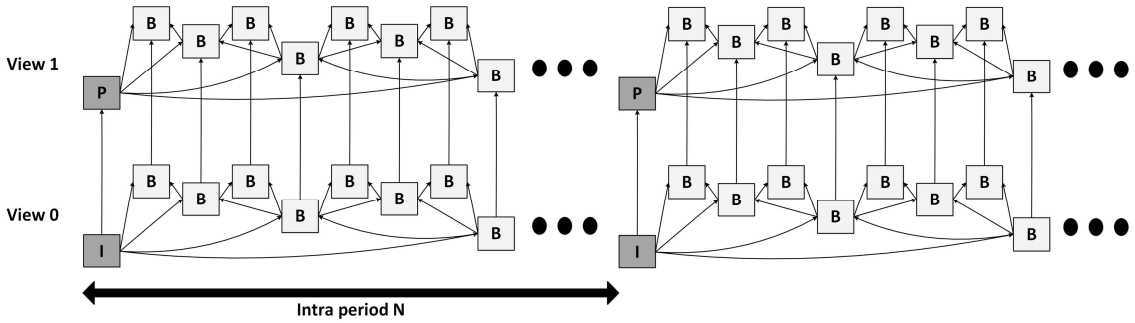


그림 1. 다시점 영상 부호화의 예측 구조
Fig. 1. Prediction structure of multiview video coding

한 영상을 부호화 할 때, 부호화되는 영상과 동일한 시간에 대응하는 독립 시점의 영상을 종속 시점 영상의 참조 영상 리스트에 추가적인 참조 영상으로 사용하는 것이다. 따라서, 독립 시점의 비디오 데이터는 다른 시점의 복호 여부와 상관없이 독립적으로 복호가능한 반면에 종속 시점의 비디오 데이터는 독립 시점의 비디오 데이터가 복호된 이후에 복호가 가능하다. 종속 시점의 영상 부호화에 있어서, 참조 영상의 종류를 2개로 구분할 수 있다. 구분되는 참조 영상 종류는 부호화되는 영상과 동일한 시점이나 다른 시간에 있는 참조 영상인 시간적 참조 영상과 부호화되는 영상과 동일한 시간이나 다른 시점에 있는 참조 영상인 시점 간 참조 영상이다. 현재 부호화되는 영상에서 시간적 참조 영상을 이용하여 화면 간 예측을 사용하는 경우, 움직임 벡터(Motion Vector, MV)를 사용하여 MCP를 하며, 시점 간 참조 영상을 이용하여 화면 간 예측을 사용하는 경우, 변위 움직임 벡터(Disparity Motion Vector, DV)를 사용하여 DCP를 한다.

1. HEVC기반 다시점 영상 부호화에서 움직임 탐색

MV-HEVC에서는 그림 2와 같이 현재 블록이 종속 시점의 영상에 존재하는 경우, 시간적 참조 영상에서 시간적 움직임 벡터를 찾고, 시점 간 참조 영상에서 변위 움직임 벡터를 찾는다. 이때, 시점 간 참조 영상에서 변위 움직임 벡터

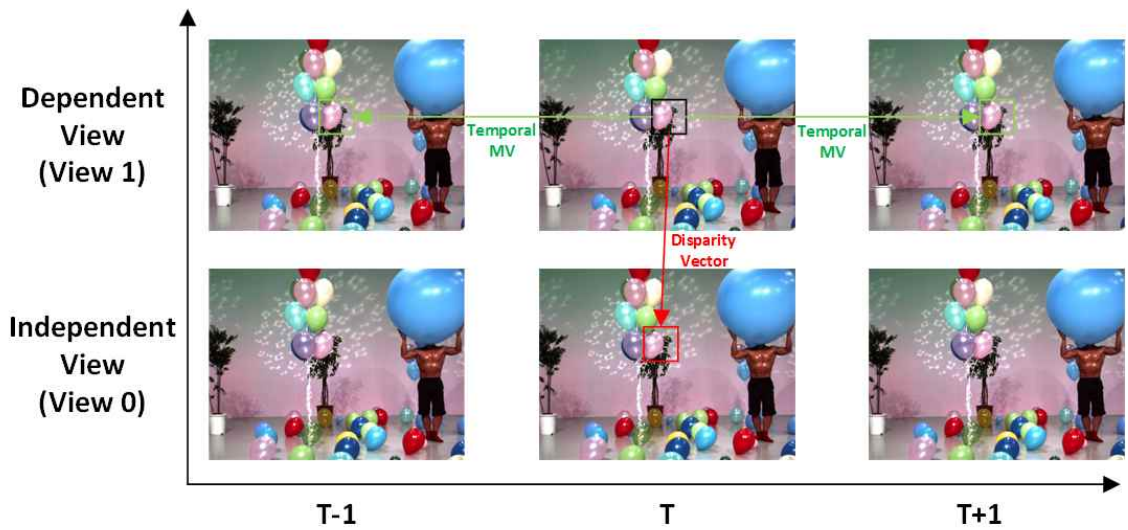


그림 2. 다시점 영상 부호화에서 시간적 움직임 벡터와 변위 움직임 벡터
Fig. 2. Temporal motion and disparity motion vector for multiview video coding

탐색은 시간적 참조 영상에서 시간적 움직임 벡터 탐색과 동일한 방법을 사용한다. 따라서, 시점 간 참조 영상에서 변위 움직임 벡터 탐색에 요구되는 복잡도는 시간적 참조 영상에서 시간적 움직임 벡터 탐색의 복잡도와 거의 동일하다.

그림 3은 MV-HEVC에서 변위 움직임 벡터 탐색을 위한 초기 탐색 지점을 보여주고 있다. 그림 3에서 P_C 는 현재 부호화되는 블록의 위치를 의미하며, P_{init} 은 정수 단위 변위 움직임 벡터 탐색을 위한 초기 탐색 지점을 의미한다. 기존 방법인 MV-HEVC에서는 변위 움직임 벡터 탐색을 위한 초기 탐색 지점으로 현재 블록의 AMVP(Ad- vanced Motion Vector Prediction)중 하나를 선택하여 P_{init} 을 결정한다. P_{init}의 결정 방법은 현재 블록의 위치와 두 개의 AMVP에 의해 구해진 참조 블록들과 원본 영상의 SAD (Sum of Absolute Difference)값들 중 낮은 SAD를 가지는 AMVP을 선택하여 P_{init}을 결정한다.

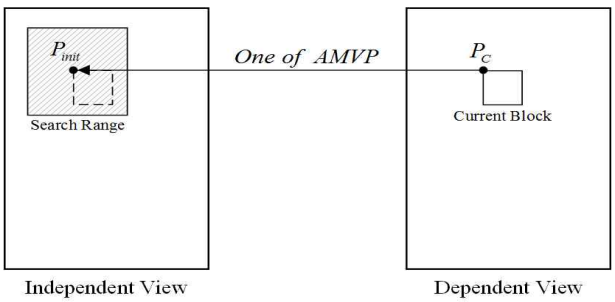


그림 3 . MV-HEVC에서 변위 움직임 벡터 탐색을 위한 초기 탐색 지점
Fig. 3. Initial search point for disparity motion vector estimation in the MV-HEVC

변위 움직임 벡터의 탐색 과정은 결정된 먼저 P_{init}의 위치를 중심으로 정수 단위 탐색을 수행하고, 가장 낮은 움직임 비용(Motion Cost)을 가지는 정수 단위 탐색 위치에서 부화소 단위 탐색을 수행하여 변위 움직임 벡터를 결정한다. MV-HEVC에서 최종 변위 움직임 벡터의 해상도는 1/4 화소 단위이다.

2. HEVC기반 다시점 영상 부호화기의 복잡도 분석

표 1은 2개의 시점에 대한 다시점 영상을 부호화하고 시간적 참조 영상으로 각 참조 방향 마다 이전에 복호된 같은 시점의 영상 2장을 사용하는 MV-HEVC에서 종속 시점 영상의 전체 움직임 탐색 시간대비 시간적 움직임 벡터 탐색 복잡도와 변위 움직임 벡터 탐색 복잡도의 비율을 보여주고 있다. 표 1에서 보여주듯이 종속 시점의 영상 부호화에 있어서 전체 움직임 탐색 복잡도 대비 시간적 움직임 벡터 탐색 복잡도 비율은 평균적으로 77.59%이며, 전체 움직임 탐색 복잡도 대비 변위 움직임 벡터 탐색 복잡도 비율은 평균적으로 22.41%이다. 각 참조 방향 마다 2장의 시간적 참조 영상을 사용하기 때문에 1장의 시간적 참조 영상에 대한 시간적 움직임 벡터 탐색 복잡도는 대략 19.40%이며, 이는 1장의 시점 간 참조 영상에 대한 변위 움직임 벡터 탐색 복잡도와 거의 동일함을 보여주고 있다.

표 1 MV-HEVC에서 시간적 움직임 벡터 탐색 복잡도와 변위 움직임 벡터 탐색 복잡도 비교
Table 1 Comparison between temporal motion estimation complexity and disparity motion vector estimation complexity

Sequence	Complexity	
	Temporal Motion Estimation (%)	disparity motion vector Estimation (%)
Balloons	77.10%	22.90%
Kendo	77.20%	22.80%
PozanaHall2	77.78%	22.22%
PozanaStreet	77.71%	22.29%
GhostTownFly	77.72%	22.28%
Shark	77.83%	22.17%
UndoDancer	77.77%	22.23%
Average	77.59%	22.41%

표 1의 결과를 통해, MV-HEVC에서 변위 움직임 벡터 탐색에 요구되는 복잡도가 높은 것을 알 수 있다. 따라서, MV-HEVC에서는 변위 움직임 벡터 탐색 복잡도를 줄이기 위한 방법이 요구되고 있다. 이러한 문제를 해결하기 위해 영상의 깊이 정보 또는 Camera Calibration 정보를 이용한 빠른 변위 움직임 벡터 예측 및 탐색 방법에 대한 많은 연구들이[10~15] 이전에 진행되어 왔다. 그러나 보통의 MV-HEVC에서는 깊이 정보 및 Camera Calibration 정보를 사

용할 수 없기 때문에 이웃한 시점간의 관계를 쉽게 구할 수 없다. 따라서, 본 논문에서는 깊이 정보 및 Camera Calibration 정보를 이용하지 않고 변위 움직임 벡터 탐색 복잡도를 줄이는 방법에 대해 다음 장에서 설명하고자 한다.

Ⅲ. 제안하는 방법

본 장에서는 변위 움직임 벡터 탐색 복잡도를 줄이기 위한 빠른 변위 움직임 벡터 탐색 방법을 설명한다. 제안하는 변위 움직임 벡터 탐색 방법은 먼저, 어파인 변환을 사용하여 시점 간 변위 움직임 벡터의 관계를 구하여 초기 탐색 지점을 결정하고, 어파인 변환 통해 결정된 초기 탐색 지점과 변위 움직임 벡터를 이용하여 정수단위 탐색 영역의 크기를 조절한다.

1. Affine Motion Model을 이용한 초기 탐색 지점

변위 움직임 벡터 탐색 복잡도를 줄이기 위해서는 먼저 초기 탐색 지점의 결정이 중요하다. 본 절에서는 어파인 변환을 이용하여 초기 탐색지점을 지정하는 방법에 대해 설명하고자 한다. 먼저, 이웃하는 두 시점 간 변위 움직임 벡터는 그림 4와 같이 종속 시점의 영상 내 동차좌표로 표현된 한 점 $P_D=(x,y,1)^\top$ 와 P_D에 대응하는 독립 시점의 영상 내 한 점 $P_I=(x^{'},y^{'},1)^\top$ 간의 위치 차이를 나타내며, 두 점 P_D 와 P_I 의 관계는 아래와 같이 표현될 수 있다.

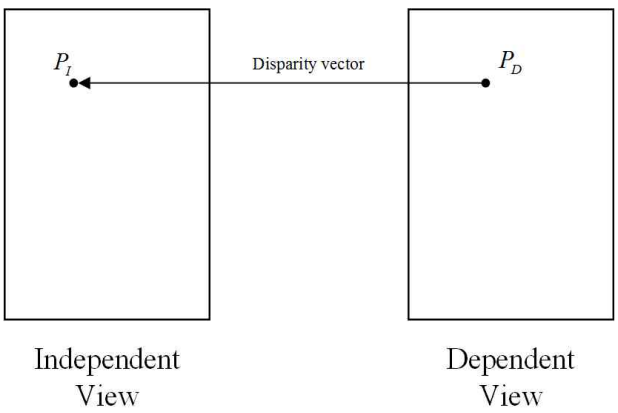


그림 4 . 두 시점 사이의 변위 움직임 벡터
Fig.4 . The Disparity motion vector between two neighboring views

$$P_I=HP_D \tag{1}$$

여기에서, 식(1)과 같이 독립 시점의 영상 내 한 점 P_I 와 대응하는 종속 시점의 영상 내 한 점 P_D 의 관계는 어파인 변환 행렬 H 로 나타낼 수 있으며, H 는 아래와 같이 표현될 수 있다.

$$H=\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

그러므로 식(1)과 식(2)를 이용하여 종속 시점의 영상 내 한 점 P_D는 아래와 같이 표현될 수 있다.

$$\begin{aligned} x^{'} &= ax+by+e \\ y^{'} &= cx+dy+f \end{aligned} \tag{3}$$

식(3)은 종속 시점의 영상 내 한 점 P_D 가 독립 시점의 영상 내 한 점 P_I 에 대응하는 관계를 나타내고 있으며, 이는 어파인 변환이라고 할 수 있다. 어파인 변환 행렬 H 의 파라미터를 구하기 위해서는 두 시점 간의 대응하는 매칭점을 알고 있어야 한다. 그러나 영상을 부호화하기 전에 매칭점을 계산하는 것은 사전 프로세스로 추가적인 모듈과 계산 복잡도가 요구된다. 따라서, 본 논문에서 제안하는 방법에서는 어파인 변환 행렬 H 의 파라미터를 구하기 위해 다음과 같은 방법을 사용한다. 먼저, 그림 5와 같이 RAP(Random Access Point) 영상을 참조 영상으로 가지는 종속 시점의 영상을 Key picture라 할 때, Key picture는 화면 간 예측을 위한 참조 영상으로 RAP 영상만을 가질 수 있으므로 Key picture에서 발생하는 모든 움직임 벡터는 변위 움직임 벡터이다. 또한, Key picture는 계층적 B-picture 구조에서 가장 낮은 QP를 사용한다. 그러므로 Key picture 복호 후 구할 수 있는 변위 움직임 벡터의 양과 정확도는 어파인 변환을 계산하는데 충분하다.

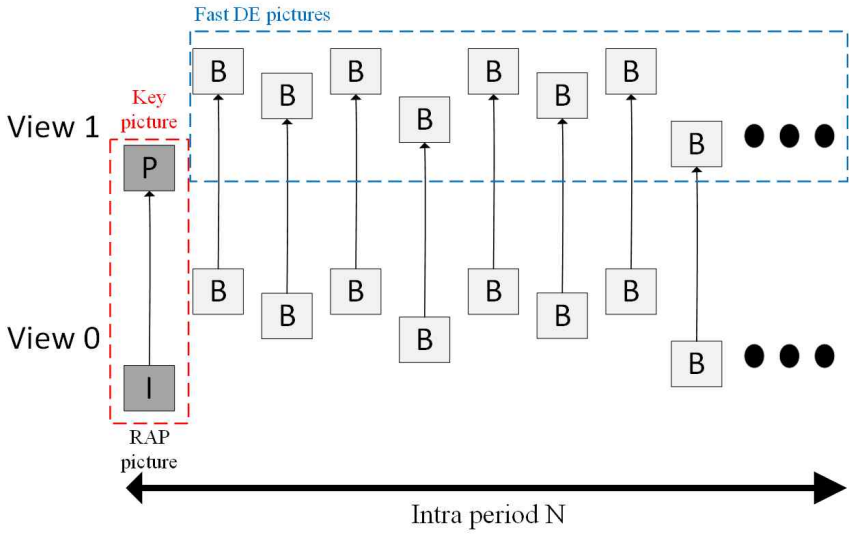


그림 5. 어파인 변환을 이용한 초기 탐색 지점 결정 방법
Fig.5. Initial search point decision using the affine transform

앞서 설명했듯이 어파인 변환 행렬 H 의 파라미터를 구하기 위해 그림 6과 같이 Key picture에서 k 번째 시점 간 예측 모드인 블록의 위치 $P_{D,k}=(x_k,y_k)$ 와 변위 움직임 벡터 DV_k 에 따라 참조 영상으로 사용되는 독립 시점의 영상 내 참조 블록의 위치 $P_{I,k}=(x_k^{'},y_k^{'})$ 를 이용하여 변환 오차 e를 구할 수 있으며, 변환 오차 e는 식(3)을 이용하여 $P_{I,k}$ 와 $HP_{D,k}$ 의 거리의 제곱으로 정의될 수 있다. 그러므로 변환 오차 e는 아래와 같이 표현될 수 있다.

$$e=\sum_{k=0}^{L-1}\left\{\left[x_k^{'}-(ax_k+by_k+e)\right]^2+\left[y_k^{'}-(cx_k+dy_k+f)\right]^2\right\} \quad (4)$$

식(4)에서 L은 Key picture에서 시점 간 예측 모드인 블록들의 개수를 의미한다. LMS(Least Mean Square) 방법에 기반하여 변환 오차 e를 a, b, c, d, e, f에 대해 편미분하면, 어파인 변환 행렬의 파라미터를 구할 수 있다. 또한, 앞서 설명했듯이 Key picture에서 발생한 변위 움직임 벡터의 양과 정확성은 Fast DE 픽쳐에서 발생한 변위 움직임 벡터보다 좋고, 보통의 다시점 영상의 경우, 화면을 획득하는 물리적 카메라의 위치는 변하지 않으므로 Key picture에서 구한 어파인 변환 행렬 H 를 다음 Key picture 직전 종속 시점의 영상인 Fast DE 픽쳐 부호화의 변위 움직임 벡터 탐색에 사용한다.

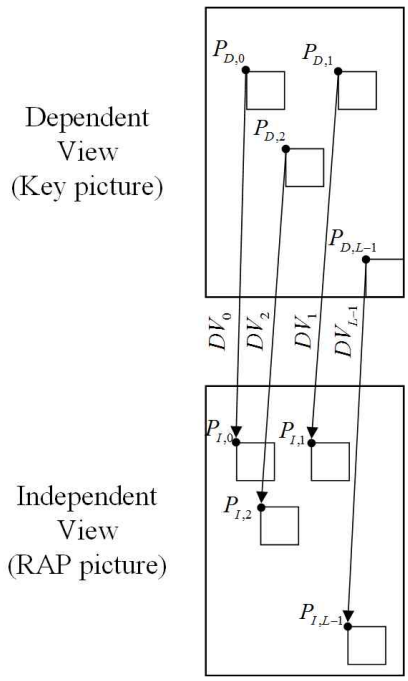


그림 6. Key 픽쳐에서의 변위 움직임 벡터
Fig.6 Disparity motion vectors in the key picture

Key picture에서 구해진 어파인 변환 행렬 H를 식(3)에 적용하여 제안하는 방법에서의 초기 탐색 지점 $P_{init}^{'} = (x_{init}^{'}, y_{init}^{'})$를 아래와 같이 구할 수 있다.

$$x_{init}^{'} = ax_c + by_c + e$$
$$y_{init}^{'} = cx_c + dy_c + f \quad (5)$$

$P_{init}^{'}$ 은 그림 7과 같이 Fast DE 픽쳐에서 변위 움직임 벡터 탐색을 위한 초기 탐색 지점으로 이용된다. 또한, MV-HEVC에서는 움직임 벡터의 1/4 단위 해상도를 사용하고 있으므로 어파인 변환 행렬의 파라미터 또한 1/4 단위 해상도의 변위 움직임 벡터들을 이용하여 구하고 있다. 따라서, $P_{init}^{'}$ 의 위치는 1/4 단위 해상도를 가지고 있다.

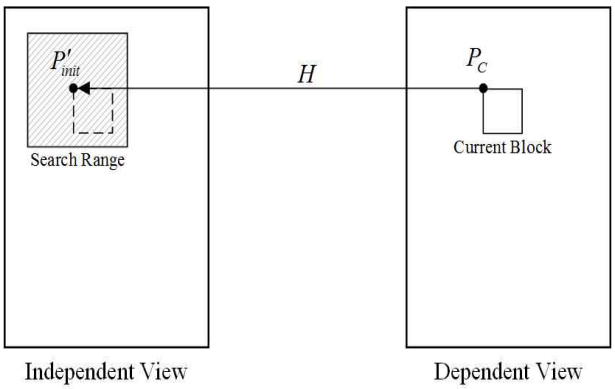


그림 7. 제안하는 방법에서 변위 움직임 벡터 탐색을 위한 초기 탐색 지점
Fig. 7. Initial search point for disparity motion vector estimation in the proposed method

2. 적응적인 탐색 영역 크기 조절

본 절에서는 빠른 변위 움직임 벡터 탐색을 위해 적응적인 탐색 영역 크기 결정 방법을 제안한다. 탐색 영역 크기는 시간적 움직임 벡터 탐색 복잡도와 변위 움직임 벡터 탐색 복잡도에 큰 영향을 미친다. 따라서, 고정된 크기의 변위 움직임 벡터 탐색 영역 대신 가변 크기의 변위 움직임 벡터 탐색 영역을 사용하여 복잡도를 크게 줄일 수 있다. 제안하는 방법에서의 탐색 영역 크기 결정 방법은 그림 8과 같이 Key picture에서 발생한 변위 움직임 벡터를 이용하여 시점 간 예측을 사용하는 각 블록의 초기 탐색 지점 $P_{init,k}^{'} = (x_{init,k}^{'}, y_{init,k}^{'})$와 변위 움직임 벡터 탐색을 통해 결정된 참조 블록의 위치 $P_{R,k} = (x_{R,k}, y_{R,k})$ 의 차이를 이용하여 결정할 수 있다.

그림 8에서 $P_{D,k}$ 는 변위 움직임 벡터를 이용하여 시점 간 예측을 사용하는 k번째 블록의 위치를 의미하며, d_k는 $P_{D,k}$ 로부터 예측된 초기 변위 움직임 벡터 탐색 지점 $P_{init,k}^{'}$ 와 k번째 블록의 변위 움직임 벡터가 지정하는 참조 블록의 위치 $P_{R,k}$ 간의 거리를 의미한다. 따라서, d_k

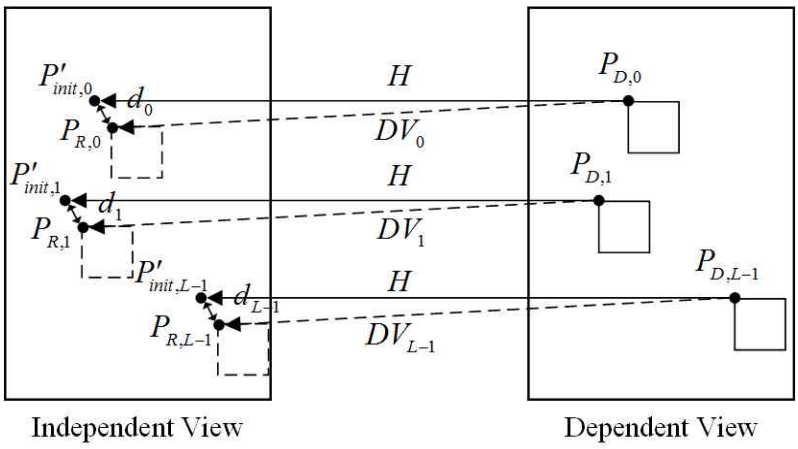


그림 8. 변위 움직임 벡터 탐색 영역 크기 결정
Fig. 8. Search range decision method for disparity motion vector estimation

는 아래와 같이 나타낼 수 있다.

$$d_k = \sqrt{(x_{R,k} - x^{'init,k})^2 + (y_{R,k} - y^{'init,k})^2} \quad (6)$$

어파인 변환을 통해 결정된 초기 변위 움직임 벡터 탐색 지점 $P^{'}_{init,k}$ 는 실제 참조 블록의 위치 $P_{R,k}$ 와 d_k 만큼의 오차가 있으므로, Fast DE 픽쳐에서 결정되는 탐색 영역의 크기 SR_{Fast} 는 아래와 같이 결정 될 수 있다.

$$SR_{Fast} = \frac{1}{L}\sum_{k=0}^{L-1} d_k \quad (7)$$

SR_{Fast} 는 Key picture에서 계산 후, 그림 9과 같이 Fast DE 픽쳐에서 변위 움직임 벡터 탐색을 위한 탐색 영역의 크기로 사용된다.

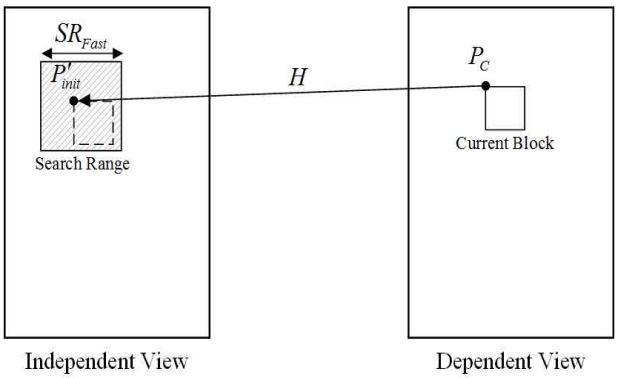


그림 9. 제안하는 변위 움직임 벡터 탐색 영역 크기
Fig. 9. Proposed search range for disparity motion vector estimation

Ⅳ. 실험 결과

본 장에서는 제안하는 방법의 성능을 확인하고자 한다. 표 2는 실험 조건을 나타내며, 성능 확인을 위해 사용된 참조 소프트웨어는 JCT-3V에서 배포하는 참조 소프트웨어 HTM16.2[16]를 사용하였다. 실험에 사용된 시점의 수는 2개이며, 탐색 방법은 Full search 방법을 적용하여 실험하였다. 나머지 실험 조건은 모두 JCT-3V에서 권장하는 Common Test Condition(CTC)를[17] 사용하였다.

표 2. 실험 조건
Table 2. Simulation condition

Coding structure	Random Access
QP setting	Independent view : defualt QP Dependent view : defualt QP+3
Intra period	24
Number of frames	100
Search method	Full search
Search range	64
Sequence	Balloons(1024x768) Kendo(1024x768) GTFly(1920x1088) UndoDancer(1920x1088) PoznanHall2(1920x1088) PoznanStreet(1920x1088) Shark(1920x1088)

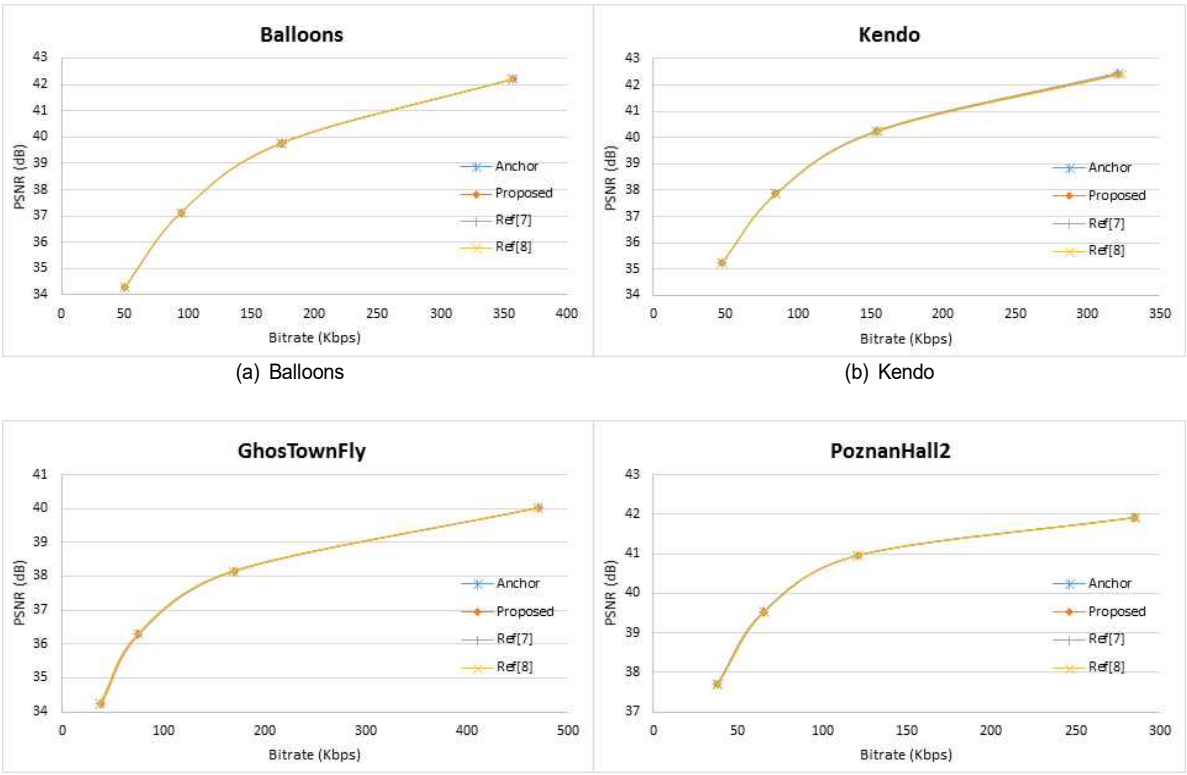
그림 10은 제안하는 기술과 비교 기술들 간의 RD- curve를 보여주고 있다. 그림 10의 (a), (b), (c), (d)는 각각 'Balloons', 'Kendo', 'GhosTownFly', 'PoznanHall2' 영상에서 종속 시점 영상에 대한 RD-curve를 보여주고 있으며, 제안하는 기술이 기존 MV-HEVC와 거의 동일한 성능을 가지고 있음을 보여주고 있다.

그림 11에서는 MV-HEVC의 변위 움직임 벡터 탐색 복잡도 대비 각 기술의 변위 움직임 벡터 탐색 복잡도를 나타내고 있다. 그림 11의 'Anchor'는 MV-HEVC의 변위 움직임 벡터 탐색 복잡도를 의미하며, 'Ref[7]' 및 'Ref[8]'은 비교 기술의 변위 움직임 벡터 탐색 복잡도를 나타낸다. 그리고 'Proposed'는 본 논문에서 제안하는 기술의 변위 움직임 벡터 탐색 복잡도를 의미한다.

그림 11의 MV-HEVC 변위 움직임 벡터 탐색 복잡도 대비 각 기술의 변위 움직임 벡터 탐색 복잡도 $Computational\ Time(\%)$은 아래와 같이 구할 수 있다.

$$Computational\ Time(\%) = \frac{DE\ Time_{Fast}}{DE\ Time_{Anchor}} \times 100 \quad (8)$$

식(8)에서 $DE\ Time_{Fast}$ 는 MV-HEVC에서 비교하는 기술을 적용하였을 때, 변위 움직임 벡터 탐색 복잡도를 나타내며, $DE\ Time_{Anchor}$ 는 MV-HEVC에서 Full Search



(a) Balloons (b) Kendo

(c) GhostTownFly (d) PoznanHall2

그림 10 . R-D 성능 비교

Fig. 10. Rate-Distortion performance comparison

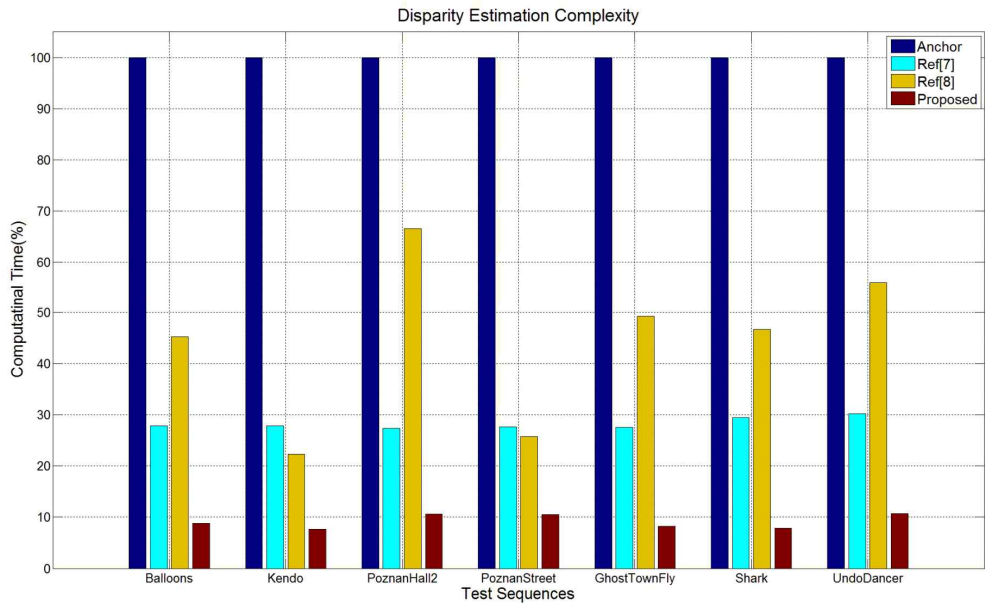


그림 11. 변위 움직임 벡터 탐색 복잡도

Fig. 11. Disparity motion vector Estimation Complexity

방법을 사용 할 때, 변위 움직임 벡터 탐색 복잡도를 의미한다. 따라서, 그림 11의 결과는 본 논문에서 제안하는 기술이 다른 기술들보다 변위 움직임 벡터 탐색 복잡도를 효과적으로 줄이는 것을 알려주고 있다.

표 3은 제안하는 방법의 성능을 나타내고 있다. 표 3에서 BD-rate(Y)[18]은 Y-component의 BD-rate를 나타내고 있으며, 변위 움직임 벡터 탐색을 사용하는 종속 시점 영상에 대한 부호화 효율을 나타낸다. 그리고 DE time[%]은 식(8)을 이용한 변위 움직임 벡터 탐색의 상대적인 복잡도를 의미한다.

표 3에서 보여주는 것과 같이 제안하는 기술은 MV-HEVC 변위 움직임 탐색 복잡도 대비 90.78%의 복잡도 감소와 0.10%의 부호화 효율 감소를 보여주고 있다. 또한, 제안하는 기술의 성능은 다른 비교기술 대비 거의 부호화 효율의 성능 저하 없이 변위 움직임 벡터 탐색 복잡도는 19.10%~31.37%를 더 감소시킨다.

이 표 3의 데이터를 살펴보면, 본 논문에서 제안하는 방법에 대해 'GTFly' 시퀀스는 급격한 부호화 효율 하락을 보인다. 'GTFly' 시퀀스는 그래픽 영상으로 SKIP 또는 머지 모드의 선택 비율이 60%가 넘는다. 따라서, 부호화하려는 주변의 움직임 벡터 또는 변위 움직임 벡터의 정확성에 따라 'GTFly' 시퀀스의 부호화 효율이 영향을 받는다. 그러나 제안하는 방법은 변위 움직임 벡터의 고속 탐색 알고리즘으로 변위 움직임 벡터의 정확성을 감소시키는 대신 부호화 복잡도를 줄이는 기술이므로 'GTFly'처럼 SKIP 또는 머지 모드의 선택 비율이 높은 시퀀스에 대해서 부호화 효율의 저하를 보일 수 있다.

표 4는 본 논문에서 제안하는 알고리즘에서 사용된 어파인 모델의 성능과 어파인 모델과 유사한 방법인 리지드 모델의 성능을 비교하는 데이터를 제시하고 있다. 표 4에서 사용한 앵커는 JCT-3V의 CTC조건을 사용하였으며, 움직임 탐색 방법으로 "TZ-Search" 방법을 사용하였다. 또한 어파인 움직임 모델과 리지드 움직임 모델은 초기 탐색 지점 결정에 사용하였으며, 움직임 벡터 탐색 영역의 크기는 앵커와 동일하게 '64'를 사용하였다. 단, 표 4에서는 본 논문의 제안하는 방법 중 하나인 움직임 벡터 탐색 영역 크기 결정 방법을 적용하지 않았다. 왜냐하면, JCT-3V의 "TZ-Search"은 탐색 영영 내 패턴 탐색을 통한 고속 탐색 방법이기 때문에 본 논문의 움직임 벡터 탐색 영역 크기 결정 방법과 비교하기 어렵기 때문이다.

표 4의 $KeyPicEncTime(\%)$ 는 앵커대비 Key 픽쳐들의 부호화 복잡도를 의미하며 아래와 같이 구할 수 있다.

$$KeyPicEncTime(\%) = \frac{KeyEncTime_{MotionModel}}{KeyEncTime_{Anchor}} \times 100 \quad (9)$$

표 3. 제안하는 방법의 성능
Table 3. Performance evaluation of proposed method

Sequence	[7]		[8]		Proposed method	
	BD-rate(Y)	DE time[%]	BD-rate(Y)	DE time[%]	BD-rate(Y)	DE time[%]
Balloons (1024x768)	0.04%	27.87%	-0.01%	45.26%	-0.04%	8.85%
Kendo (1024x768)	0.01%	27.83%	0.81%	22.33%	-0.13%	7.67%
PoznanHall2 (1920x1088)	-0.16%	27.61%	0.14%	38.81%	-0.37%	10.50%
PoznanStreet (1920x1088)	0.13%	27.67%	0.06%	25.74%	-0.02%	10.68%
GTFly (1920x1088)	0.02%	27.56%	0.06%	49.23%	1.26%	8.23%
Shark (1920x1088)	0.00%	29.50%	-0.10%	46.71%	0.04%	7.90%
UndoDancer (1920x1088)	-0.01%	30.23%	-0.09%	56.05%	-0.02%	10.72%
Average	0.00%	28.32%	0.12%	40.59%	0.10%	9.22%

식 (9)에서 $KeyEncTime_{MotionModel}$은 어파인 또는 리지드 모션 모델을 사용하는 경우, Key 픽쳐의 부호화 복잡도를 의미하며, $KeyEncTime_{Anchor}$는 앵커에서 Key 픽쳐의 부호화 복잡도를 의미한다. 또한 $Enc.Time(\%)$는 앵커의 전체 부호화 복잡도 대비 어파인 모션 모델 또는 리지드 모션 모델로 초기 탐색 지점을 결정하였을 때, 전체 부호화 복잡도를 비교한 것이다. 따라서, $Enc.Time(\%)$는 아래와 같이 구할 수 있다.

$$Enc.Time(\%) = \frac{EncTime_{MotionModel}}{EncTime_{Anchor}} \times 100 \qquad (10)$$

식 (10)에서 $EncTime_{MotionModel}$은 어파인 또는 리지드 모션 모델을 사용하는 경우, 전체 부호화 복잡도를 의미하며 $EncTime_{Anchor}$ 는 앵커의 전체 부호화 복잡도를 의미한다.

표 4에서 보여주는 것과 같이 초기 움직임 벡터 탐색 지점을 결정할 때, 어파인 변환을 사용하는 어파인 모션 모델이 리지드 변환을 사용하는 리지드 모션 모델보다 평균 -0.24%의 부호화 효율 향상을 보여주고 있으며, 전체 부호화 복잡도는 거의 동일하다는 것을 알수 있다. 또한, Key 픽쳐의 부호화 복잡도 비교를 통해 어파인 변환 행렬과 리지드 변환 행렬의 파라미터의 계산 복잡도의 차이는 평균 0.36%차이를 보이고 있으며, 이는 두 방법의 계산 복잡도는 거의 차이가 없다는 것을 보여주고 있다. 또한 표 3과 결과 표 4의 결과는 초기 탐색 지점 결정 방법으로 본 논문에서 제안하는 방법인 어파인 모션 모델이 효율적이지만, 부호화 복잡도를 줄이기 위해 변위 움직임 벡터 탐색 영역을 줄이는 경우, 종래의 기술과 거의 동일한 부호화 효율 저하부호화를 가지고 복잡도는 현저히 줄이는 것을 보여주고 있다.

V. 결 론

본 논문에서는 빠른 변위 움직임 벡터 탐색을 위해 어파인 변환을 이용한 초기 탐색 지점 결정 방법과 적응적인 탐색 영역 크기 결정 방법을 제안하였다. 제안하는 방법은 기존 MV-HEVC 대비 90.78%의 변위 움직임 벡터 탐색 복잡도를 줄였으며, 부호화 효율 성능 하락은 0.10%이었다. 따라서, 본 논문에서 제안하는 빠른 변위 움직임 벡터 탐색 방법이 깊이 정보 및 Camera calibration 정보를 사용하지 못하는 경우, 효율적으로 변위 움직임 벡터 탐색 복잡도를 줄이는 것을 보여주고 있다.

표 4. Affine 모션 모델과 Rigid 모션 모델의 성능 비교
Table 4. Performance comparison between affine motion model and rigid motion model

Sequence	Affine Motion Model			Rigid Motion Model		
	BD-rate(Y)	KeyPic Enc Time (%)	Enc.Time (%)	BD-rate(Y)	KeyPic Enc Time (%)	Enc.Time (%)
Balloons (1024x768)	-0.01%	101.45%	101.15%	0.17%	101.72%	100.88%
Kendo (1024x768)	-0.25%	100.66%	100.50%	0.22%	102.97%	100.45%
PoznanHall2 (1920x1088)	-0.44%	100.99%	99.38%	-0.40%	100.76%	99.38%
PoznanStreet (1920x1088)	0.01%	101.55%	100.24%	0.33%	101.00%	99.85%
GTFly (1920x1088)	0.06%	101.21%	100.83%	0.56%	101.87%	100.93%
Shark (1920x1088)	-0.12%	99.59%	98.98%	-0.01%	99.82%	98.49%
UndoDancer (1920x1088)	0.00%	99.87%	98.97%	0.08%	99.72%	98.68%
Average	-0.11%	100.76%	100.01%	0.13%	101.12%	99.81%

본 논문의 제안하는 방법은 Key 픽쳐의 부호화 이후 어파인 변환 행렬의 파라미터를 계산하는 과정으로 인해 다음 픽쳐간의 딜레이가 발생할 수 있다. 또한, Key 픽쳐들 중간에서 장면 변화가 발생하는 경우, 잘못된 모션 모델을 적용함으로써 부호화 효율을 저하시킬 우려가 있다. 그리고 더욱 효율적인 움직임 벡터 탐색 영역 크기 결정 방법에 대한 추가의 연구가 필요한 것으로 판단된다. 따라서, 이러한 문제들을 해결하기 위한 후속 연구를 계획하고 있다.

참 고 문 헌 (References)

[1] High Efficiency Video Coding, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.

[2] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649 – 1668, Dec. 2012.

[3] G. Sullivan, J. Boyce, Y. Chen, J.-R. Ohm, C. Segall, and A. Vetro, “Standardized extensions of high efficiency video coding (hevc),” IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 6, pp. 1001 – 1016, Dec. 2013.

[4] G. Tech, K. Wegner, Y. Chen, M. M. Hannuksela, and J. Boyce, “MV-HEVC draft text 5,” in Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1004, 5th Meeting: Vienna, Austria, Jul. 27 – Aug. 2 2013.

[5] G. Tech, K. Wegner, Y. Chen, and S. Yea, “3D-HEVC draft text 1,”in Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1001, 5th Meeting: Vienna, Austria, July 27 – Aug. 2 2013.

[6] G. Tech, Y. Chen, K. Muller, J.-R Ohm, A, Vetro, and Y. Wang "Overview of the Multiview and 3D Extensions of High Efficiency Video Coding", IEEE Trans. on CSVT, Vol. 26, no. 1, pp. 35-49, Jan. 2016.

[7] Y.-T. Chang and W.-H. Chung, “An adaptive search range algorithm for multiview motion and disparity estimation”, Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on, pp.550-554, 2-5 July 2012.

[8] S. Khattak, R. Hamzaoui and S. Ahmad, “Low-complexity multiview video coding”, Picture Coding Symposium (PCS), pp. 97-100, Krakow, Poland, 7-9 May 2012.

[9] Y. Chen, X. Zhao, L. Zhang and J. -W. Kang, “Multiview and 3D Video Compression Using Neighboring Block based Disparity Vectors”, IEEE Transaction on Multimedia, vol.18, no. 4, pp.576-589, Apr. 2016.

[10] Dong-Hoon Han, Suk-Hee Cho, Nam-Ho Hur,Yung-Lyul Lee, “Fast Mode Decision using Global Disparity Vector for Multi-view Video Coding”, JBE, Vol. 13, No. 3 , pp. 328-338, May, 2008.

[11] Yoon Jin Lee, Dong In Bae, Gwang Hoon Park,“HEVC Encoder Optimization using Depth Information”, JBE, 19, No. 5, pp. 640-655, September, 2014.

[12] Daemin Park, Haechul Choi, “Performance Analysis of 3D-HEVC Video Coding”, JBE, Vol. 19, No. 5, pp. 713-725, September, 2014.

[13] N. Boonthep, W. Chiracharit and K. Chamnongthai, “Adaptive search range determination for geometry based disparity and motion estimation of MVC”, Asia-Pacific Signal and Information Processing Association (APSIPA), 2014 Annual Summit and Conference on, pp.1-5, 9-12 Dec. 2014.

[14] B. W. Micallef, C. J. Debono and R. A. Farrugia, “Low complexity disparity estimation for immersive 3D video transmission”, Communications Workshops (ICC), 2013 IEEE International Conference on, pp.612-616, 9-13 June 2013.

[15] B. W. Micallef, C. J. Debono, and R. A. Farrugia, “Fast disparity estimation for Multi-view plus depth video coding”, 2011 Visual Communications and Image Processing (VCIP), pp. 1-4, 6-9 Nov. 2011.

[16] https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/tags/HTM-16.2/

[17] K. Müller and A. Vetro, “Common Test Conditions of 3DV Core Experiments”in Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-G1100, 7th Meeting, San José, US, 11 – 17 Jan. 2014.

[18] G. Bjontegaard, Calculation of Average PSNR Differences Between RDCurves, document VCEG-M33, ITU-T Q.6/SG16 VCEG, Apr. 2001.

저 자 소 개

이 재 영

- 2011년 : 세종대학교 정보통신공학과 학사
- 2013년 : 세종대학교 정보통신공학과 공학석사
- 2013년 3월 ~ 현재 : 세종대학교 정보통신공학과 박사과정
- 2014년 12월 ~ 2016년 1월 : UC San Diego Visiting Scholar
- ORCID : http://orcid.org/0000-0001-5049-2574
- 주관심분야 : 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템

한 종 기

- 1992년 : KAIST 전기및전자공학과 공학사
- 1994년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 3월 ~ 2001년 8월 : 삼성전자 DM 연구소 책임연구원
- 2001년 9월 : 세종대학교 정보통신공학과 교수
- 2008년 9월 ~ 2009년 8월 : UC San Diego Visiting Scholar
- ORCID : http://orcid.org/0000-0002-5036-7199
- 주관심분야 : 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템