

특집논문-04-09-3-02

경계 방향성을 고려한 스테레오 동영상의 움직임-변이 동시추정 기법

김용태^{a)*}, 서형갑^{a)}, 박창섭^{b)}, 이재호^{b)}, 손광훈^{a)}

Edge-Directional Joint Disparity-Motion Estimation of Stereoscopic Sequences

Yongtae Kim^{a)*}, Hyunggap Seo^{a)}, ChangSeob Park^{b)}, JaeHo Lee^{b)} and Kwanghoon Sohn^{a)}

요 약

본 논문에서는 스테레오 동영상 CODEC (Coder and decoder)을 위한 효율적인 변이와 움직임의 동시추정 기법을 제안한다. 동시추정 기법은 좌우 움직임 벡터와 이전 시점의 변이 벡터를 이용해서 현재 시점의 변이 벡터를 예측한다. 하지만 동시추정 기법은 추정 오류가 축적되고 가려진 영역으로 인해 벡터들이 잘못 추정될 경우 성능이 매우 떨어질 수가 있다. 이런 문제점을 해결하기 위해서 동시추정 기법과 별도로 공간적인 변이 벡터의 예측을 수행한다. 즉, 동시추정과 공간적인 변이 벡터의 예측을 통해서 정확한 변이 벡터들을 구해내어 전체 부호화 효율을 높일 수 있다. 마지막으로 본 논문에서는 역방향 사진트리 분할 기법을 제안한다. 역방향 사진트리 분할 기법은 사진트리 분할 정보를 보내지 않고도 상세한 변이맵을 얻어낼 수 있기 때문에 변이 보상의 성능을 높일 수 있었다. 실험 결과를 통해서 제안 알고리즘이 기존 알고리즘과 비교하여 수행시간이 9배가량 감소하고 주관적 화질에서 좋은 결과를 보임을 확인하였고 객관적 화질 평가에서 0.5~1.5dB가량 PSNR이 높아짐을 확인하였다.

Abstract

This paper presents an efficient joint disparity-motion estimation algorithm for stereo sequence CODEC. Disparity vectors are estimated by the left and right motion vectors and previous disparity vectors for every frame. In order to obtain more accurate disparity vectors, we include a spatial prediction process after the joint estimation. From joint estimation and spatial prediction, we can obtain accurate disparity vectors and then increase coding efficiency. Finally, we proposed the backward quadtree decomposition, which helps the encoder to have a more detailed disparity vector map without transmitting additional coding bits for quadtree information. We confirmed superior performance of the proposed method through computer simulation.

Keywords : Joint disparity-motion estimation, Stereo sequence, Edge-directional technique

I. 서 론

이론적으로 스테레오 영상의 전송을 위해서는 기존 영상에 비해 두 배의 대역폭이 필요하지만, 두 영상은 서로 수

평하고 인간의 눈 사이 거리만큼 떨어진 곳에서 획득된 영상이므로 상호 큰 유사성 (Correlation)을 가지고 있다. 그러므로 이 유사성을 이용해 데이터양을 크게 줄일 수 있는 장점이 있다^{[1][2][3]}. 스테레오 영상 부호화에서 가장 기본이 되는 것은 두 영상의 유사성을 찾아 그 변이를 벡터로 표현하는 변이 추정 (Disparity estimation)이다^{[4][5][6]}. 변이 추정의 방법은 시간 중복성을 없애기 위해 사용되는 움직임 추정 (Motion estimation)과 비슷하여 보통의 경우 움직

a) 연세대학교 전기전자공학과

Dept. of Electrical and Electronic Engineering, Yonsei University

b) 한국방송 방송기술연구소

KBS Broadcast Technical Research Institute

입 추정에서 활용되는 방법을 그대로 이용한다. 우리는 스테레오 동영상으로부터 움직임 추정과 변이 추정을 수행하고 변이 벡터, 움직임 벡터와 차영상 정보를 전송한다. 수신부에서는 이 정보들을 수신하여 스테레오 동영상을 재구성한 후 3차원 디스플레이를 통해 사용자들에게 입체감을 느끼게 해준다. 기존의 스테레오 동영상 CODEC은 많은 정보량과 연산 시간을 필요로 하였다. 대부분의 기존 방식들은 전 방향 탐색 알고리즘 (Full search algorithm)을 사용하는데 이는 가장 적은 오류 벡터를 추정할 수 있지만 많은 연산량을 필요로 하는 기법이다. 이 문제점을 해결하기 위해서 많은 논문에서 변이 및 움직임 벡터의 동시추정 기법이 연구되었다^{[7][8][9]}. 하지만 기존의 동시추정 기법은 이전의 잘못된 예측 과정으로 인해 파생된 변이 벡터들의 오류가 축적되어 이후 시점 프레임에 전달되는 문제점을 가지고 있다. 게다가 가려진 영역에서는 전혀 다른 변이 벡터를 예측하기 때문에 동시추정 기법은 많은 단점을 가지고 있다. 이러한 문제점들은 변이 보상기법에서 많은 오류를 발생시키기 때문에 스테레오 부호화기의 성능을 높이기 위해서는 이를 보완할 방법이 필요하다.

본 논문에서는 변이와 움직임 필드 상에서 상호 관계에 기초를 둔 효율적인 변이 및 움직임 동시추정 기법을 제안한다. 제안 방법은 경계 성분을 검출하여 공간적으로 변이 벡터를 예측해 내는데 이 공간적인 예측 기법은 동시추정 기법을 보완하여 보다 정확한 변이 벡터를 예측할 수 있게 해준다. 이는 공간적인 변이 벡터의 예측과 보정 과정을 통해서 동시추정 기법으로 구한 벡터들의 오류들이 상당히 줄어들기 때문이다.

본 논문의 전체적인 구성은 다음과 같다. 제 2 장에서는 스테레오 동영상 CODEC과 변이 벡터와 움직임 벡터의 상호 관계에 대하여 소개하고 제 3 장에서는 공간적인 변이 벡터의 예측 기법을 제안하고, 제 4 장에서는 모의실험을 통해서 제안 알고리즘의 우수성을 검증하였다. 마지막으로 제 5장에서는 결론을 맺는다.

II. 변이 및 움직임 동시추정 기법

1. 스테레오 동영상 CODEC의 구조

그림 1은 스테레오 동영상의 압축 순서도를 보여주고 있다. 스테레오 동영상 CODEC은 기본 레이어 (Base layer)에

왼쪽 시점의 동영상을 확장 레이어 (Enhancement layer)에 오른쪽 시점의 동영상을 입력으로 하여 부호화 과정을 수행한다^[10]. 그림 1에서 기본 레이어는 시간축의 중복성을 없애는 움직임 추정을 통해서만 현재 프레임을 보상한다. 하지만 확장 레이어의 경우 이전 시점의 프레임으로부터 움직임 추정과 같은 시점의 왼쪽 영상으로부터의 변이 추정을 통하여 보상 과정을 수행한다. 우리는 변이 및 움직임 벡터의 동시추정 기법의 성능을 높이기 위해서 그림 1과 같이 단순한 GOP (Group of pictures) 구조를 사용한다.

기본 레이어와 확장 레이어 사이에는 변이 보상 기법을 통해서 중복성을 제거하는데 변이 추정 오류가 크게 발생하여 변이 추정이 실패할지라도 확장 레이어는 움직임 추정을 통해서 적은 오류로 현재 프레임을 보상할 수 있다. 그러므로 가려진 영역과 같이 변이 추정이 불가능한 영역이 존재하더라도 스테레오 동영상 CODEC은 확장 레이어의 프레임을 완벽하게 보상할 수 있다. 기본 레이어는 독립적으로 부호화하게 되는데 이는 기존의 2차원 복호화기에서도 복원되게 하기 위해서다. 2차원 디스플레이 환경에서는 기본 레이어의 스트림을 전송받아 2차원 영상을 감상할 수 있고 3차원 디스플레이를 가지고 있을 경우에는 기본 레이어에 확장 레이어를 더해서 전송 받아 스테레오 동영상을 복원하여 입체 영상을 감상할 수 있게 해준다. 스테레오 동영상 CODEC에서 왼쪽 동영상은 움직임 추정 기법을 이용하여 기본 레이어로 압축되고 오른쪽 동영상은 확장 레이어로 압축되어 전송된다. 오른쪽 동영상에 대한 압축은 움직임 및 변이 보상과 DCT (Discrete cosine transform) 부호화기를 통해서 수행된다. 이런 과정을 통해 압축된 좌, 우 동영상은 시스템 MUX에 넣어 패킷 단위로 전송한다. 하지만 이런 개념의 스테레오 동영상 CODEC는 한 시점의 동영상을 압축할 때 필요한 코딩 비트량과 연산량을 두 배 가까이 필요로 한다^[11].

2. 변이 및 움직임 벡터의 상호 관계

$R_i^{(l)}$ 과 $R_i^{(r)}$ 은 좌, 우 영상 평면이라 하고 $z = (x, y)$ 는 $R_i^{(l)}$ 의 한 점이라고 가정한다. $R_i^{(l)}$ 과 $R_i^{(r)}$ 상의 두 점 $z = (x, y)$ 와 $\tilde{z} = (\tilde{x}, \tilde{y})$ 이 서로 매칭이 된다는 것은 z 와 \tilde{z} 가 실제 공간상의 한 점 $R_i^{(l)}$ 과 $R_i^{(r)}$ 으로 투영된 지점임을 의미한다. 이 경우 우리는 변이 벡터 d_i 를 $z = (x, y)$ 에서 $\tilde{z} = (\tilde{x}, \tilde{y})$ 로의 벡터로 정의한다. 이와

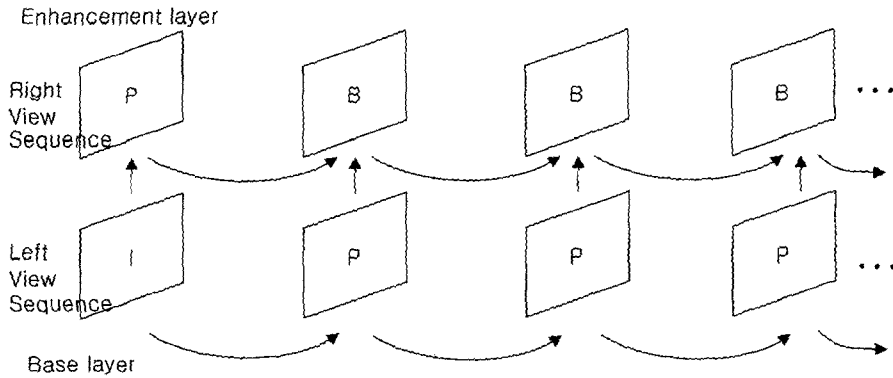


그림 1. 스테레오 동영상의 압축 순서도
Fig. 1. A stereoscopic video sequences

비슷하게 $R^{(t)}$ 에서 $R^{(t+1)}$ 로의 움직임 벡터를 왼쪽 동영상일 경우 $m_l(z, t)$, 오른쪽 동영상일 경우 $m_r(z, t)$ 로 정의한다^[9].

수식 (1)은 연속한 스테레오 프레임 쌍에서 두개의 움직임 벡터와 두개의 변이 벡터들을 통해서 움직임 및 변이 벡터의 일관성 있는 조건식을 표현하였다. 그림 2는 t 시점에서 각 프레임에서의 매칭되는 지점을 간략히 보여주고 있다. 그림 2를 수식으로 표현하면 수식 (1)과 같다.

$$\| d_l(z, t) + m_r[z + d_l(z, t), t] - d_l[z + m_l(z, t), t + 1] - m_l(z, t) \| = \delta \quad (1)$$

수식 (1)에서 $\delta = 0$ 으로 가정할 경우 t 시점의 왼쪽 움직임 벡터를 오른쪽 움직임 벡터, t 시점과 $t + 1$ 시점에서의 변이 벡터로부터 예측할 수 있다. 이는 스테레오

CODEC의 수행 시간을 감소시키고 부호화되는 비트량도 줄일 수 있다. 하지만 움직임 또는 변이 벡터들이 부정확할 경우 잘못된 동시추정을 수행하게 되고 이렇게 잘못 예측된 벡터들의 오류는 축적되어 다음 시점에서 동시 추정할 때 많은 오류를 발생 시키게 된다^[11]. 이러한 이유 때문에 동시추정 기법은 초기 예측 벡터로 사용되거나 추정된 벡터들의 신뢰도를 구하는 과정의 조건으로만 사용되었다.

본 논문에서는 블록 기반의 스테레오 CODEC 구조를 사용하기 때문에 수식 (1)에 필요한 변이 벡터와 움직임 벡터들을 직접 구해낼 수가 없다. 또한 좌영상에서 우영상으로의 변이 추정과정을 대신해서 동시추정 기법을 수행하고자 하기 때문에 논문의 목적에 맞게 수식 (1)을 수정해야만 한다. $t + 1$ 시점에서 이미 좌, 우 움직임 벡터와 이전 t 시점에서의 변이 벡터를 알고 있다. 위의 세 벡터들로부터 $d_r(z, t + 1)$ 을 구해내기 위해서 수식 (2)를 사용한다.

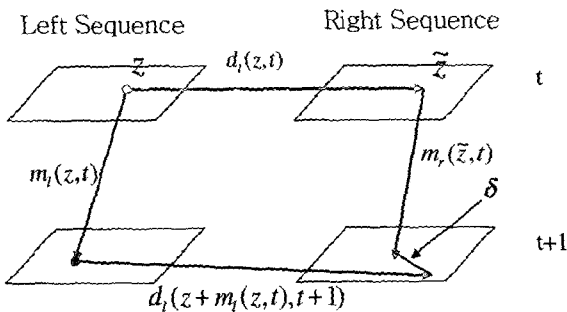


그림 2. 변이 및 움직임 벡터의 관계
Fig. 2. Relationship between disparity and motion vectors

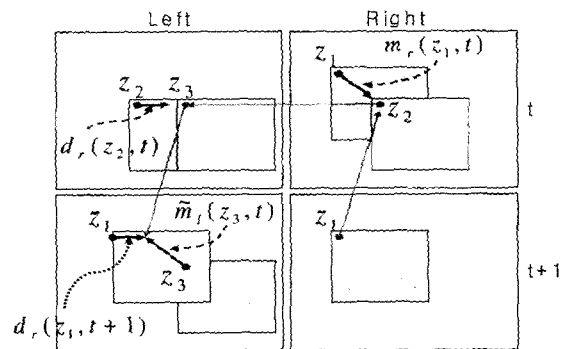


그림 3. 블록 단위 동시 추정 기법
Fig. 3. A flow of the block-based joint estimation

$$d_r(z_1, t+1) = m_r(z_1, t) + d_r(z_2, t) + \tilde{m}_r(z_3, t) \quad (2)$$

z_1 : 초기 위치, $z_2 : z_1 + m_r(z_1, t)$
 $z_3 : z_1 + m_r(z_1, t) + d_r(z_2, t)$

그림 3은 블록 기반의 동시 추정 기법의 흐름도를 상세히 보여주고 있다. 수식 (2)에서는 세 벡터를 통해서 $d_r(z, t+1)$ 을 쉽게 구해내지만 세 벡터는 모두 다른 시작점을 가지고 있다. 그러므로 블록 단위의 변이 및 움직임 벡터로부터 각 시작점에 대한 벡터의 값을 유추해야만 한다.

초기 위치 z_1 에서의 $m_r(z_1, t)$ 과 z_2 에서의 $d_r(z_2, t)$ 은 쉽게 구해낼 수 있지만 z_3 에서의 $\tilde{m}_r(z_3, t)$ 는 직접 구해낼 수 없다. 여기서, \tilde{m}_r 은 m_r 의 역방향 움직임 벡터를 의미한다. \tilde{m}_r 에 대한 정보가 없기 때문에 m_r 로부터 이를 구해내는 부가적인 연산이 필요하다. 역방향 움직임 벡터 영역은 3가지로 분류될 수 있다. $R^{(t)}$ 과 $R^{(t+1)}$ 사이의 움직임 벡터들이 일대일로 매칭 되는 일반 영역, 한 지점에 여러 벡터들이 매칭이 되는 중첩 영역, 어떤 벡터들도 매칭이 되지 않는 빈 영역으로 나누어진다. 일대일 매칭되는 일반 영역의 경우 \tilde{m}_r 은 m_r 의 반대 방향이므로 m_r 의 부호를 반대로 해줌으로써 \tilde{m}_r 을 구해낼 수 있다. 겹쳐진 영역의 경우 매칭되는 여러 벡터들 중에 가장 정확한 벡터를 찾아내는 과정을 수행하는데 수식 (3)을 이용하여 가장 적은 오류를 내는 벡터를 매칭 벡터로 선택하게 된다.

$$VEC_{error} = \sum_{i=1}^4 |m_i(z_i, t) - m_{candi}(z, t)| \quad (3)$$

여기서, $m_i(z_i, t)$ 는 주변의 인과적인 블록들의 미리 구한 움직임 벡터들을 가리키고 $m_{candi}(z, t)$ 는 현재 지점 z 에 매칭되는 여러 움직임 벡터들 중에 하나를 의미한다. 일반 영역과 마찬가지로 가장 작은 오류를 내는 움직임 벡터를 우선으로 만들면서 현재 지점의 \tilde{m}_r 을 구해낼 수 있다. 마지막으로 빈 영역의 경우 주변의 움직임 벡터를 이용해서 가장 가까이 있는 움직임 벡터를 해당 지점의 움직임 벡터로 대체한다.

왼쪽 영상의 역방향 움직임 벡터를 계산한 후, 수식 (2)를 이용해서 블록위의 변이 및 움직임 벡터로부터 $d_r(z, t+1)$

을 구해낼 수 있다. 하지만 하나의 매크로블록 내에 여러 변이 벡터의 값이 존재하기 때문에 블록단위의 변이 추정을 수행하기 위해서는 블록마다 대푯값을 하나씩 선택해야 한다. $m_r(z, t)$, $\tilde{m}_r(z, t)$ 과 $d_r(z, t)$ 을 구하는 과정에서 오류들이 발생하기 때문에 변이 벡터 필드 상에 적지 않은 오류들이 존재하게 된다. 그러므로 블록내의 변이 벡터 값들의 평균값을 대푯값으로 설정할 경우 크게 잘못된 일부 변이 벡터 값들로 인해 잘못된 대푯값을 선택할 수 있다. 이런 오류를 최소화시키기 위해서 본 논문에서는 평균값대신 미디언 값을 대표 벡터 값으로 사용한다. 즉, $t+1$ 시점에서 동시추정 기법을 이용해서 블록기반의 변이 벡터를 생성하기 위해서는 블록내의 변이 벡터 값들의 미디언 값을 구해내고 이 값을 블록의 대표 벡터로 설정한다.

III. 경계 방향성을 고려한 공간적인 변이 벡터의 예측

이 절에서는 경계 방향성을 고려한 공간적인 변이 벡터의 예측 기법에 대해서 제안한다.

변이 및 움직임 동시추정 기법을 사용하여 $t+1$ 시점에서 블록 단위의 변이 벡터를 생성하였다. 하지만 위에서도 언급했듯이 동시추정 기법은 변이 벡터의 오류가 전달되기 때문에 많은 문제점을 가지고 있다. 또한 가려진 영역에서 정확한 변이 추정과 움직임 추정이 어렵기 때문에 동시추정의 성능이 저하되기도 한다. 위의 문제점에 대한 해결 방안으로 본 논문에서는 동시추정 기법에 추가적인 과정을 첨가하였다. 그림 4는 제안 알고리즘의 흐름도를 보여주고

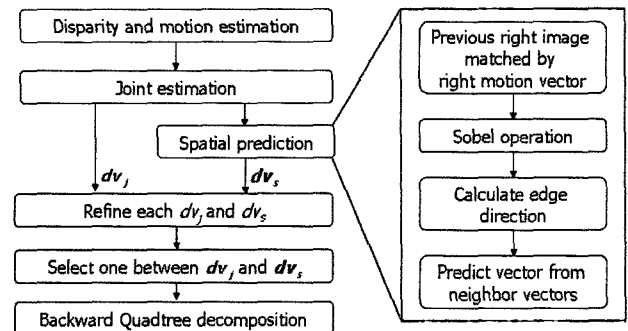


그림 4. 제안 알고리즘의 흐름도
 Fig. 4. A flow chart of the proposed method

있다. 공간적인 변이 벡터의 예측 기법은 $m_x(z, t)$ 을 이용하여 매칭 영역 계산, 소벨 연산, 경계의 방향성 계산과 변이 벡터의 예측으로 모두 4개의 과정으로 나누어져 있다. 동시추정 기법과 공간적인 변이 벡터의 과정을 통해서 우리는 dv_x 와 dv_y 의 두 개의 예측 변이 벡터들을 구해낸다. 이 벡터들에 대해서 보정 과정을 수행한 후 가장 작은 정합 오류를 내는 변이 벡터를 선택하게 된다. 마지막으로 선택된 벡터를 기준으로 매크로블록을 규칙적인 역방향 사진 트리를 이용하여 부호화한다.^[12]

1. 경계의 방향성 검출

경계의 방향성을 결정하기 위해서 먼저 현재 매크로블록의 질감 정보를 알아내야만 한다. 그렇지만 복호화기에서도 부호화기와 마찬가지로 경계의 방향성을 검출하기 위해서는 현재의 매크로블록의 질감 성분을 사용할 수는 없다. 그러므로 미리 알고 있는 오른쪽 움직임 벡터에 의해 t시점에 매칭되는 영역을 이용하여 질감 정보를 유추해내어 현재 매크로블록의 질감 정보를 대신한다.

경계의 방향은 그림 5.(a)와 같이 수평, 수직, 두 개의 대각성분으로 모두 네 종류가 있다.

현재 매크로블록이 이 네 경계의 방향 중 어느 쪽에 속하는지 판별하기 위해서 본 논문에서는 소벨 연산을 수행하였다. 소벨 연산에 사용된 소벨 마스크는 그림 5.(b, c, d 와 e)로 각 방향에 대한 마스크가 존재한다. 움직임 벡터로 보

상한 질감 성분에 대해서 각 마스크를 이용하여 소벨 값을 구해내고 값들의 절대값의 합을 구한다. 그 중 가장 큰 절대 값을 가지는 방향을 현재 매크로블록의 경계의 방향으로 결정한다.

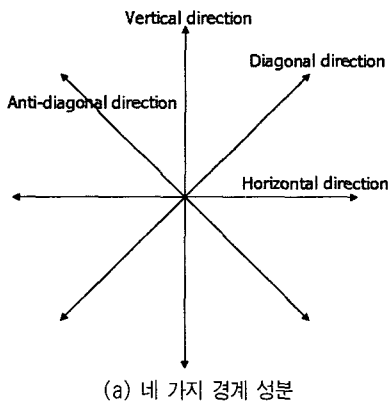
더 정확한 경계 방향의 판단을 위해서 평활한 영역에 대해서도 고려를 하였다. 소벨 연산을 수행한 후 우리는 각 방향에 대한 소벨 값의 절대 값의 합을 가지고 있다. 이 절대 값들을 이용하여 현재 매크로블록이 평활한 영역인지를 판별한다.

$$sum = sum_h + sum_v + sum_d + sum_{ad} \quad (4)$$

여기서 sum_h, sum_v, sum_d 과 sum_{ad} 는 각 방향에 대한 소벨 값들의 절대 값의 합을 의미한다. sum 이 어느 임계 값 (TH_h)보다 작을 경우 해당 매크로블록을 평활한 영역으로 분류한다. 영상의 특성을 고려해서 임계 값은 수식 (5)을 이용해서 결정한다.

$$TH_h = \frac{1}{N_x N_y} \sum_{(i,j) \in I^o} (I_h(i,j) + I_v(i,j) + I_d(i,j) + I_{ad}(i,j)) \times c, \quad c \in [0, 1] \quad (5)$$

$I_h(i, j), I_v(i, j), I_d(i, j)$ 와 $I_{ad}(i, j)$ 는 각각 수평, 수직, 두 개의 대각 성분의 소벨 값을 의미하고 N_x 와 N_y 는



-1	-2	-1
0	0	0
1	2	1

(b) 수평 방향

-1	0	1
-2	0	2
-1	0	1

(c) 수직 방향

-2	-1	0
-1	0	1
0	1	2

(d) 대각 방향(45°)

0	1	2
-1	0	1
-2	-1	0

(e) 대각 방향(-45°)

그림 5. 각 방향에 대한 소벨 마스크

Fig. 5. Sobel mask for each direction. (a) four edge directions (b) horizontal direction (c) vertical direction (d) diagonal direction (e) anti-diagonal direction

영상의 폭과 높이를 의미한다. 상수 c 을 조절하면서 평활 영역을 조절할 수 있다. 상수 c 가 높을 경우 임계 값 TH_h 는 높아지게 되고 많은 매크로블록을 평활한 영역으로 판단하게 된다. 반대로 상수 c 가 낮아지게 되면 임계 값 TH_h 는 작아지게 되고 평활한 영역으로 선택될 매크로블록들이 줄어들게 된다.

2. 공간적인 변이 벡터의 예측

기존의 공간적인 벡터의 예측은 인과적인 매크로블록만을 이용하여 현재 매크로블록을 예측하였다. 이는 이후의 매크로블록들이 아직 부호화되지 않아 그 정보가 복호화단에서 사용될 수 없기 때문이다. 즉 부호화단에서는 인과적인 매크로블록에 대한 정보만을 이용해서 현재 매크로블록의 벡터를 예측하는 방법밖에 없었다. 그러나 제안 알고리즘에서는 동시추정 기법을 이용해서 주위의 모든 블록의 벡터들을 미리 알고 있으므로 효율적인 변이 벡터의 예측을 수행할 수 있다.

그림 6에서 현재 매크로블록을 포함하여 주위에 $v_1 \sim v_9$ 의 9개의 벡터들이 존재한다. $v_1 \sim v_4$ 까지는 이전에 구한 변이 벡터들이고 $v_5 \sim v_9$ 는 동시추정 기법을 이용하여 구한 벡터들이다. 우리는 이 9개의 변이 벡터들을 이용해서 공간적인 변이 벡터의 예측을 수행한다. 이전 단계에서 경계 성분이 어떤 방향으로 현재 매크로블록을 지나가는지 또는 현재 매크로블록이 평활한 영역인지 아닌지를 조사하였다. 만약 매크로블록에 경계 성분이 존재할 경우 해당 블록이 경계의 어느 영역에 속하는지 모르기 때문에 두 개의 예측 벡터를 가지게 된다. 그림 6.(a)에서 경계는 블록을 수평 방향으로 지나가고 있다. 이러한 경우는 두개의 객체가 수평 경계를 기준으로 접하고 있는 상태이므로 현재 블록이 위의 객체에 속하거나 아니면 아래의 객체에 속하게 된다. 위의 객체의 경우 v_1, v_2 와 v_3 을 포함하고 있고, 아래의 객체의 경우 v_7, v_8 와 v_9 을 포함하고 있다. 각각 포함된 벡터들에 대해서 미디언 필터를 수행하고 나온 결과 벡터를 예측 벡터로 설정한다. 경계가 있는 블록의 경우 두 개의 예측 벡터가 나오게 되는데 각각 dv_{s1} 와 dv_{s2} 로 정의한다. 편의상 dv_{s1} 은 왼쪽 부분에 있거나 위쪽인 영역의 예측 벡터이고 dv_{s2} 는 오른쪽 부분이

거나 아래쪽에 있는 영역의 예측 벡터로 설정한다. 그림 6.(a)와 같은 방법으로 그림 6.(b, c와 d)에도 예측 벡터들을 구해내는데 차이점은 미디언 필터에 대한 입력 벡터들이 다르다. 평활한 영역의 경우 9개의 변이 벡터들에 대해서 미디언 필터를 수행하여 예측 벡터를 구해내게 된다. 이 경우 하나의 예측 벡터 dv_s 만을 구해낸다.

공간적인 변이 벡터의 예측을 수행한 후 우리는 동시추정 기법으로 구한 변이 벡터와 함께 두 개의 예측 변이 벡터를 가지게 된다. 이 두 예측 벡터에 대해서 보정 과정을 수행한다. 보정 과정은 예측 벡터를 주변으로 작은 검색 범위를 지정하여 그 범위 내에서 다시 변이 벡터를 검색하는 과정이다.

마지막으로 두 예측 벡터 (dv_s 와 dv_j)중에 보정 과정에서 가장 적은 오류를 내는 벡터를 선택한다. 이 선택 정보와 보정된 벡터는 복호화단으로 전송된다. 본 논문에서는 선택 정보를 그냥 전송하지 않고 RLC (Run length coding) 방법과 산술 부호화 부호화기를 사용하여 보다 효율적으로 압축하여 전송한다.

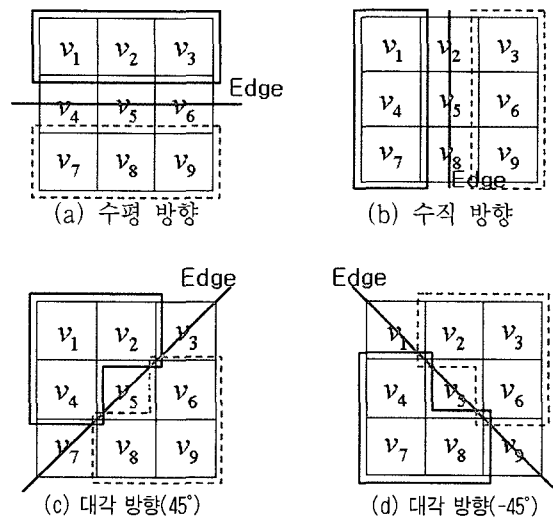


그림 6. 공간적인 변이 벡터의 예측

Fig. 6. Spatial disparity vector prediction (a) horizontal direction (b) vertical direction (c) diagonal direction (d) anti-diagonal direction

IV. 모의실험 결과 및 분석

본 논문에서는 동시추정 기법을 위한 경계 방향성을 고

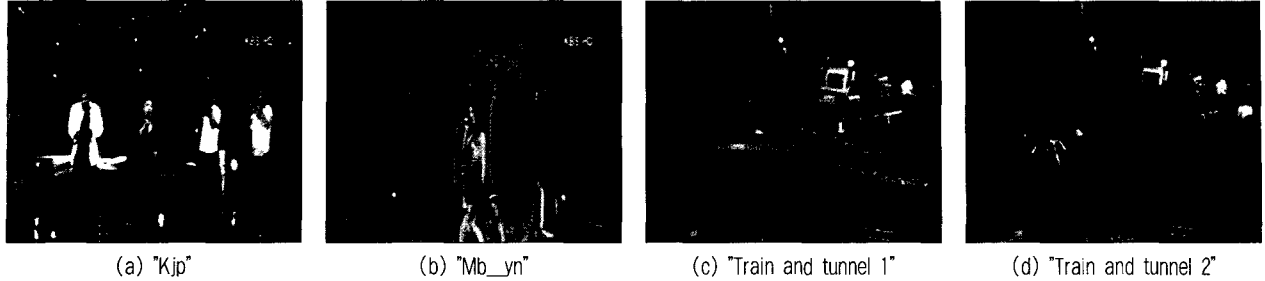


그림 7. 실험 영상 (좌영상의 첫 번째 프레임)
 Fig. 7. Test sequences (the first left frame for each sequence) (a) "Kjp" (b) "Mb_yn" (c) "Train and tunnel 1" (d) "Train and tunnel 2"

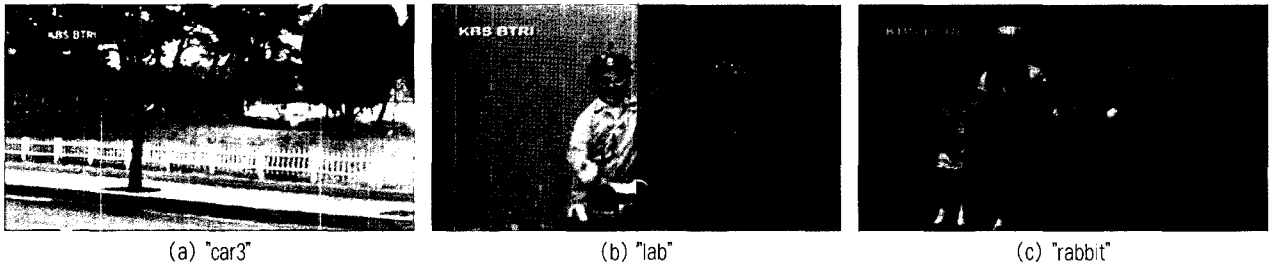


그림 8. HD급 스테레오 동영상 (좌영상의 첫 번째 프레임)
 Fig. 8. HD stereoscopic sequences (the first left frame for each sequence) (a) "car3" (b) "lab" (c) "rabbit"

려한 변이 벡터의 예측 기법에 대해서 제안하였다. 실험은 그림 7의 "Kjp", "Mb_yn"과 "Train and Tunnel" 120 프레임 스테레오 동영상과 그림 8의 "car3", "lab"와 "rabbit"의 240 프레임을 가지는 HD급 스테레오 동영상에 대해서 수행하였다. 각 동영상의 크기는 "Kjp"와 "Mb_yn"의 경우 960×1080이고 "Train and tunnel"은 352×288, "car3", "lab"와 "rabbit"은 HD급으로 1920×1080이다.

변이 추정에 사용된 각 파라미터는 영상의 특성을 고려하여 결정되었으며 아래의 표 1과 같다.

표 1. 실험에 사용된 파라미터
 Table 1. Experimental parameters

		"Kjp" 와 "Mb_yn"	Train and tunnel	"car3", "lab"과 "rabbit"
블록의 크기		16×16		
변이 검색 범위	수평	-64~+64	-32~+32	-64~+64
	수직	-16~+16	-4~+4	-8~+8
움직임 검색 범위		-32~+32	-16~+16	-32~+32
보정 과정 검색 범위	수평	-4~+4		
	수직	-2~+2		

기존 알고리즘으로는 주어진 검색 범위 내의 모든 검색 지점에서 정합 오류를 찾는 전 방향 탐색 알고리즘을 사용하였다. 표 2에서는 제안 알고리즘과 전 방향 탐색 알고리즘을 주어진 코딩 비트량에서 PSNR을 비교하였다. 코딩 비트량은 역방향 사진트리 분할 기법에서 분할 정도를 이용하여 조절하였다. 대부분의 실험 영상에 대해서 제안 알고리즘의 기존의 알고리즘과 비교하여 우수한 성능이 나옴을 확인할 수 있다. 구체적으로 살펴보면 "Kjp" 스테레오 동영상은 코딩 비트량은 일정하면서 PSNR은 0.8dB 정도 증가한다. 대부분의 스테레오 동영상에 대해 0.3~0.8dB 가량 PSNR이 증가하고 코딩 비트량은 약간 감소하는 것을

표 2. 비슷한 코딩 비트량에서 PSNR 비교
 Table 2. PSNR comparison in similar coding bits

	기존 방식		제안 방식	
	코딩 비트량	PSNR	코딩 비트량	PSNR
Kjp	52341	28.33	48321	29.12
Mb_yn	41270	28.43	37356	28.78
Train_1	7867	27.38	7505	27.65
Train_2	8859	26.68	8066	27.29

표 3. HD급 스테레오 동영상의 PSNR 비교 (dB)
Table 3. PSNR comparison of HD stereoscopic sequences

		19.38Mbps	38.76Mbps
car3	Side by Side	31.99	33.38
	전 방향 탐색 알고리즘	32.10	33.55
	제안 알고리즘	32.43	33.74
lab	Side by Side	35.49	36.23
	전 방향 탐색 알고리즘	35.17	38.25
	제안 알고리즘	36.48	38.31
rabbit	Side by Side	35.34	36.21
	전 방향 탐색 알고리즘	35.54	37.42
	제안 알고리즘	35.94	37.47

확인할 수 있다.

기존 알고리즘과 비교하여 제안 알고리즘은 연산량이 비약적으로 감소한다. "Kjp"와 "Mb_yn" 스테레오 동영상의 경우 기존 알고리즘에서 변이 추정을 수행하는데 55,000~56,000msec 정도의 연산 시간이 걸리는 반면 제안 알고리즘에서의 변이 추정은 동시추정 기법을 사용하기 때문에 평균 6,200~6,300msec 정도의 시간만을 필요로 한다. 즉 연산 시간이 약 9배 정도 감소하게 된다. 다른 스테레오 동영상도 마찬가지로 비슷한 실험 결과를 얻었다. 이 결과로부터 본 논문의 제안 알고리즘이 스테레오 동영상에서 변이 추정에 필요한 연산량을 크게 낮추어 주는 것을 확인하였다. 이를 통해 스테레오 동영상 CODEC의 구현 시 발생하는 변이 추정의 계산의 복잡도 문제를 해결하여 스테레오 입체 방송 시 지연을 줄일 수 있다.

전체 스테레오 동영상 CODEC 시스템에 대해서 기존 알

고리즘과 제안 알고리즘을 비교하였다. 기존 알고리즘으로는 위에서 소개한 전 방향 탐색 알고리즘과 그림 9의 "Side by Side" 방식을 사용하였다. "Side by Side" 방식의 경우 스테레오 동영상을 수평 성분을 반으로 줄여 하나의 좌, 우 영상을 하나의 영상으로 만들어 기존의 2차원 동영상 CODEC으로 압축하는 방식이다.

표 3은 240 프레임의 "car3", "lab"과 "rabbit" HD급 스테레오 동영상에 대해 "Side by Side" 방식, 전 방향 탐색 알고리즘과 제안 알고리즘의 평균 PSNR을 보여주고 있다. 목표 비트 레이트는 방송 대역폭을 고려하여 19.38Mbps와 38.76Mbps로 설정하였다. "car3" 스테레오 동영상의 경우 움직임 매우 크고 영상의 질감 정보가 매우 복잡하기 때문에 PSNR이 작게 나온다. 나머지 두 영상 "lab"과 "rabbit"은 배경이 간단하여 상대적으로 높은 PSNR을 가진다. 전체적으로 제안 알고리즘이 가장 성능이 좋게 나오고 그 다음으로 전 방향 탐색 알고리즘과 "Side by Side" 방식 순서로 나온다. "lab" 스테레오 동영상의 경우 19.38Mbps에서 "Side by Side" 방식이 전 방향 탐색 알고리즘보다 좋게 나오는데 이는 "lab" 영상이 다른 영상과 비교하여 좌, 우 영상의 불일치 문제가 심각하기 때문이다. 좌, 우 영상의 불일치가 심할 경우 변이 추정의 성능이 떨어지게 되어 전체 부호화 성능을 크게 떨어뜨릴 수가 있다. 이런 이유로 전 방향 탐색 알고리즘의 성능이 떨어지게 되어 상대적으로 "Side by Side" 방식의 PSNR이 올라가게 된 것이다.

그림 10은 19.38Mbps에서 "rabbit" HD급 스테레오 동영상의 240프레임에 대한 PSNR의 변화를 보여주고 있다. 몇 프레임에서 기존 방식의 PSNR이 높을 때도 있지만 대부분의 프레임에 대해서 제안 알고리즘이 더 높은 PSNR을 가지는 것을 알 수 있다. 제안 알고리즘과 전 방향 탐색 알고

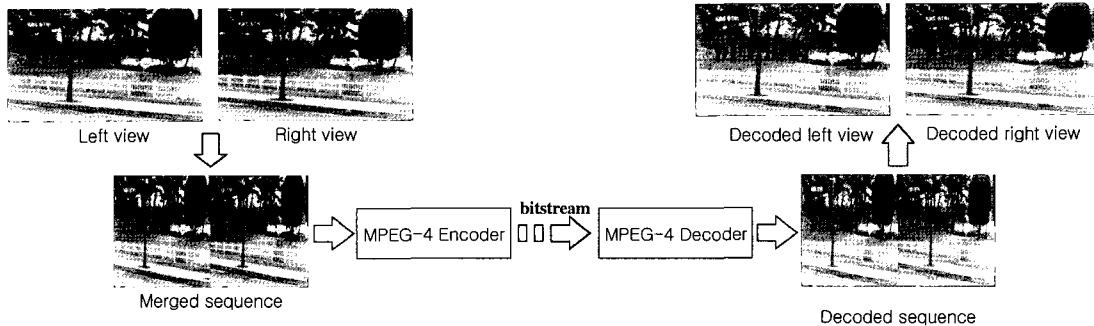


그림 9. "Side by Side" 기법
Fig. 9. "Side by Side" method

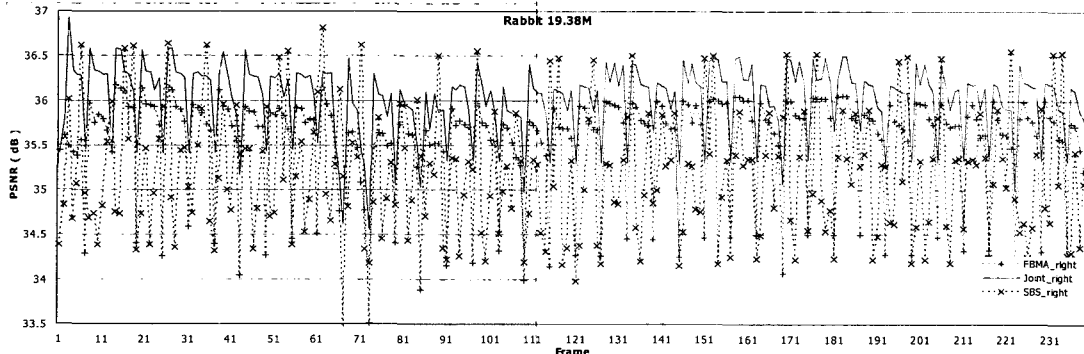


그림 10. 19.38Mbps에서 "lab" HD급 스테레오 동영상의 PSNR
 Fig. 10. PSNR of "lab" HD stereoscopic sequence (19.38Mbps)

리즘의 우영상의 경우 움직임 추정과 변이 추정을 동시에 수행하기 때문에 보다 PSNR이 높은 보상 영상을 만들어 낼 수 있다. 그렇지만 "Side by Side" 방식은 움직임 추정만을 이용하여 우 영상을 복원하기 때문에 효율적인 부호화 과정을 수행할 수가 없다. 또한 "Side by Side" 기법은 움직임에 민감하게 반응하여 PSNR의 변화폭이 크지만 변이 추정을 사용하는 다른 두 알고리즘은 상대적으로 일정한 PSNR을 가지게 된다.

주관적인 화질 평가에서 "Side by Side" 방식의 경우 원 영상을 반으로 축소하고 복호화단에서 다시 내삽법을 이용해서 원래의 크기로 복원하기 때문에 결과 영상이 전체적으로 흐릿해지게 된다. 전 방향 탐색 알고리즘은 이웃 블록간의 상관성을 고려하지 않고 비용 함수를 최소화시키는 벡터를 선택하기 때문에 블로킹 현상이 발생하게 된다. 하지만 제안 알고리즘의 경우 동시추정 기법과 공간적인 변이 벡터의 예측 기법을 이용하여 변이 벡터들을 정확히 예측하기 때문에 블로킹 현상이 줄어들게 된다.

V. 결 론

본 논문에서는 변이 및 움직임 벡터의 동시추정을 위한 경계의 방향성을 고려한 공간적인 변이 벡터 예측 기법을 제안한다. 또한 경계의 방향성을 고려한 벡터 예측 기법을 통해 변이 벡터의 신뢰도를 높일 수 있었고 역방향 사진트리 분할 기법을 이용해서 부가적으로 분할 정보를 보내지 않고도 매크로블록을 효율적으로 나눌 수가 있었다.

실험 결과를 통해서 동시 추정 기법으로 변이 벡터를 추정할 경우 많은 장점이 있음을 확인하였다. 동시추정 기법은 미리 구한 좌, 우 두 움직임 벡터와 이전 시점의 변이 벡터를 이용해서 현재 시점의 변이 벡터를 예측하기 때문에 직접 변이 벡터를 구하는 방식과 비교하여 많은 계산상의 이득을 얻는다. 실험 결과 제안 알고리즘이 대략 9배 정도 연산량의 이득이 있다. 본 논문에서는 다양한 실험 영상과 비트 레이트에서 실험을 수행하였는데 모든 환경에서 제안 알고리즘이 기존 알고리즘들과 비교하여 높은 PSNR을 얻었다. 또한 주관적인 화질 평가에서도 제안 알고리즘이 가장 좋은 결과 영상을 보여주고 있다.

본 논문에서 제안한 기법들을 향후 다시점 동영상으로 확장하여 적용하고자 한다. 다시점 동영상의 경우 기존의 2차원 영상이나 스테레오 영상과 비교하여 더 많은 연산량과 코딩 비트를 필요로 하기 때문에 제안 알고리즘을 적용할 경우 많은 이득이 있을 것으로 예상된다.

참 고 문 헌

- [1] M. G. Perkins, "Data compression of stereopairs," IEEE Trans. on Comm., vol. 40 no. 4, pp.684-696, April 1992.
- [2] H. S. Kim and K. Sohn, "Feature-based disparity estimation for intermediate view reconstruction of multiview images," Proc. CISST, vol. 2, pp. 1-8, June 2001.
- [3] J. E. Lim, K. H. Sohn, "MPEG-2 Compatible multiview sequence encoder," Proc. CISST, pp 379-385, July 2002.
- [4] C. W. Lin, E. Y. Fei and Y. C. Chen, "Hierarchical disparity estimation using spatial correlation," IEEE Trans. on Consumer Electronics, vol. 44 no. 3, pp. 630-637, Aug. 1998

- [5] D. Tzovaras, N. Grammalidis and M. G. Strintzis, "Disparity field and depth map coding for multiview 3D image generation," Signal Processing : Image Comm, 11, pp. 205-230, 1998.
- [6] M. Ebroul Izquierdo, "Stereo image analysis for multiview-viewpoint tele- presence applications," Signal Processing: Image Comm., vol. 11, no. 3, pp. 231-254, 1998.
- [7] S. Malassiotis and M.G. Strintzis, "Joint motion/disparity MAP estimation for stereo image sequences," IEE Proc.-Vision, Image, Signal Proc., vol. 143, no. 2, pp.101-108, 1996.
- [8] I. Patras, N. Alvertos and G. Tziritas, "Joint disparity and motion field estimation in stereoscopic image sequences," Proc. IEEE ICPR, pp.359-643, 1996.
- [9] M. Ebroul Izquierdo, "Stereo matching for enhanced telepresence in three dimensional videocommunications," IEEE Trans, On Circuit and Systems, vol. 7, no.4, pp.629-643, 1997.
- [10] A. Puri, R.V. Kollaritis and BG. Haskell, "Stereoscopic video compression using temporal scalability," Proc. SPIE VCIP, vol. 2501, pp.745-756, 1995.
- [11] K.H. Sohn, J.R. Ryou and J. Lim, "Efficient stereoscopic video coding using joint disparity-motion estimation," Circuits, Systems and Signal Processing, vol. 23, no.1, pp.57-76, 2003.
- [12] 최승철, 김용태, 손광훈, "효율적인 다시점 영상 부호화를 위한 역방향 사진트리 변이 추정," 대한 전자 공학회 하계종합학술대회, pp 1911-1914, 2003년 7월.

저 자 소 개



김 용 태

- 2000년 2월 : 연세대학교 기계전자공학부 학사
- 2002년 2월 : 연세대학교 전기전자공학과 석사
- 2002년 9월~현재 : 연세대학교 전기전자공학과 박사과정
- 주관심분야 : 3차원 동영상 부호화/복호화, 움직임 및 변이 추정, MPEG-2/4, H.26x



서 형 갭

- 2003년 2월 : 홍익대학교 전기전자컴퓨터공학부 학사
- 2003년 9월~현재 : 연세대학교 전기전자공학과 석사과정
- 주관심분야 : 3차원 동영상 부호화/복호화, 움직임 및 변이 추정, MPEG-2/4, H.26x



박 창 섭

- 1980년 : 한양대학교 전자공학과, 학사
- 1982년 : 한양대학교 산업대학원, 석사
- 2001년 : 한국과학기술원 정보 및 통신공학과, 박사
- 1981년~현재 : 한국방송 방송기술연구팀
- 주관심분야 : 영상신호처리, 컴퓨터비전, 입체 TV

저 자 소 개



이 재 호

- 1994년 : 경북대학교 전자공학과 학사
- 1996년 : 경북대학교 전자공학과 석사
- 1996년~현재 : 한국방송 방송기술연구팀
- 주관심분야 : 영상신호처리, 컴퓨터 비전, 입체TV



손 광 훈

- 1983년 : 연세대학교 전자공학과 공학사
- 1985년 : University of Minnesota, MSSE
- 1991년 : North Carolina State University, Ph.D
- 1993년 : 한국전자통신연구원 선임연구원
- 1994년 : Georgetown University, Post-doctor fellow
- 1995년~현재 : 연세대학교 전기전자공학과 부교수
- 주관심분야 : 3차원 영상처리, 영상 통신