2014년 1월 전자공학회 논문지 제 51권 제 1 호
Journal of The Institute of Electronics and Information Engineers Vol. 51, NO. 1, January 2014 http://dx.doi.org/10.5573/ieie.2014.51.1.107

논문 2014-51-1-14

영상의 저 비트 변환을 이용한 SAD 블록 정합 알고리즘

(Reduced-bit transform based block matching algorithm via SAD)

김 상 철*, 박 순 용**, 진 성 일***

(Sang-Chul Kim, Soon-Yong Park, and Sung-Il Chien©)

요 약

영상의 저 비트 변환 기반의 비트 플레인 정합방법(Bit-Plane Matching : BPM)은 기존의 블록 정합 방법들과 비교해 계산량을 줄이고 간단한 하드웨어 구조 설계를 통해 블록 정합 결과를 획득할 수 있지만, 블록 정합의 정확도가 비교적 낮은 문제점을 가지고 있다. 본 논문에서는 기존의 BPM방법들과 비교해 블록 정합의 정확도를 증가시키면서 동시에 논리 연산으로 정합 결과를 계산할 수 있는 저 비트 변환 기반의 절대 오차합(Reduced-bit transform based Sum of Absolute Difference : R-SAD)을 이용한 블록 정합 알고리즘을 제안한다. 이 방법은 현재 영상과 참조영상을 각각 2-bit의 영상으로 변환하고, 2-bit의 4레벨에 대한 입출력 관계를 이용하여 진리표를 획득한다. 진리표는 Karnaugh map을 통해 간소화 되어 논리 연산으로 절대 오차를 계산할 수 있다. 제안된 방법의 성능 평가를 위한 움직임 보상(Motion Compensation) 실험에서, R-SAD는 기존의 블록 정합 방법들과 비교해 높은 정확도의 정합결과를 획득할 수 있었다.

Abstract

The reduced-bit transform based bit-plane matching algorithm (BPM) can obtain the block matching result through its simple calculation and hardware design compared to the conventional block matching algorithms (BMAs), but the block matching accuracy of BPMs is somewhat low. In this paper, reduced-bit transform based sum of the absolute difference (R-SAD) is proposed to improve the block matching accuracy in comparison with the conventional BPMs and it is shown that the matching process can be obtained using the logical operations. Firstly, this method transforms the current and the reference images into their respective 2-bit images and then a truth table is obtained from the relation between input and output 2-bit images. Next, a truth table is simplified by Karnaugh map and the absolute difference is calculated by using simple logical operations. Finally, the simulation results show that the proposed R-SAD can obtain higher accuracy in block matching results compared to the conventional BPMs through the PSNR analysis in the motion compensation experiments.

Keywords : block matching algorithm, bit-plane matching, reduced-bit transform, SAD, combinational logic

Ⅰ. 서 론

블록 정합 알고리즘(Block Matching Algorithm: BMA)은 동영상의 움직임 추정(Motion estimation)이나 스테레오 비전의 깊이 맵(Depth map)획득을 위해 영상간 블록들의 유사도를 측정하는 방법으로 널리 사용되고 있다. BMA는 영상을 먼저 블록 단위로 분할하고, 한 영상의 현재 블록을 나머지 다른 영상의 미리 정의된 영역내 후보 블록들간 정합기준(Matching criterion)을 계산

* 정회원, ㈜ LG전자 TV연구소
(TV R&D Center, LG Electronics)
** 정회원, 경북대학교 IT대학 컴퓨터학부
(School of Computer Science and Engineering, Kyungpook National University)
*** 정회원, 경북대학교 IT대학 전자공학부
(School of Electronics Engineering, Kyungpook National University)
© Corresponding Author (E-mail: sichien@ee.knu.ac.kr)
※ 본 연구는 국방과학연구소의 민군기술협력진흥센터 및 한국원자력연구원의 지원으로 수행되었음.
접수일자: 2013년10월18일, 수정완료일: 2014년1월1일

하여 최소의 왜곡을 가진 하나의 후보 블록을 찾는 방법이다[1,2]. 다양한 BMA들 중 블록간의 유사도 판별을 위해 널리 사용되고 있는 절대 오차합(Sum of Absolute Differences: SAD)은 현 블록과 후보 블록 내 픽셀 간 오차들의 절대값을 누적하여 블록정합 결과를 계산한다. 이를 통해, SAD는 비교적 정확한 블록 정합결과를 획득할 수 있음에도 불구하고, 많은 계산을 요구하기 때문에 실시간 시스템 적용에는 제한적이다[3].

최근에는 블록 정합의 계산량을 저감하기 위해 입력 영상을 저 비트로 변환하여 블록의 유사도를 계산하는 비트 플레인 정합방법(Bit-Plane Matching: BPM)들이 제안되었다[2, 4~7]. BPM은 밴드 패스 필터(Band-pass filter: BPF)를 이용하여 원 영상을 1-bit 또는 2-bit의 표현레벨을 가지는 비트 플레인으로 변환하고, Boolean exclusive-OR(XOR)연산을 통해 블록 내 픽셀 간 서로 동일하지 않은 픽셀들의 수(Number of Non-Matching Points : NNMP)를 누적하여 블록 정합 결과를 계산한다. 따라서 한 번에 여러 개의 픽셀들을 병렬로 연산할 수 있는 조합 논리 회로로 블록 간의 정합결과를 계산할 수 있기 때문에 기존의 SAD에 비해 계산량과 하드웨어 복잡도를 현저히 저감할 수 있지만, 블록 정합의 정확도가 비교적 낮은 단점을 가지고 있다.

본 논문에서는 기존의 BPM에서 주로 사용하는 NNMP 블록 정합방법들 보다 정합 정확도가 높으면서 동시에 조합논리회로로 구현이 가능한 Reduced-bit SAD(R-SAD)블록 정합 방법을 제안한다. 이 방법은 입력 영상들을 저 비트의 비트 플레인들로 변환하고, 비트 플레인들간 절대오차 계산 결과에 대한 입출력 관계를 이용하여 진리표를 획득한다. 진리표는 Karnaugh map을 통해 간소화 되어 논리식으로 표현할 수 있으며, 이를 통해 절대오차를 계산한다. 따라서 기존 SAD의 8-bit 뺄셈기를 사용하는 방법보다 하드웨어 복잡도가 낮은 조합 논리 회로를 이용하여 절대 오차계산이 가능하다. 이와 함께, R-SAD는 기존 BPM방법들과 동일하게 비트 플레인을 사용하지만 블록 정합결과의 표현 레벨이 기존의 NNMP보다 많아 비교적 정확한 블록 정합결과를 획득할 수 있는 장점이 있다.

Ⅱ. 비트 플레인 기반의 블록정합방법

비트 플레인(bit-plane) 영상으로부터 생성된 이진 영상을 이용한 블록 정합 방법은 기존의 그레이레벨 영상에서 블록 정합에 필요한 산술 연산들을 간단한 이진 논리 연산으로 대처할 수 있는데 장점을 가진다[8]. 이를 위해 원 영상을 비교적 효과적으로 나타낼 수 있는 비트 플레인 생성 방법과 생성된 비트 플레인을 이용한 정합방법은 블록 정합결과의 정확도를 결정하는 중요한 척도가 된다. 대표적인 BPM방법으로는 1BT(one-bit transform)방법과 2BT(two-bit transform)방법이 있다[4~5].

1BT는 원 영상을 이진 영상으로 변환하여 얻어지는 에지 성분들이 정확한 블록정합을 계산하는데 중요한 역할을 한다[4]. 일반적으로 에지 성분을 추출하는 가장 간단한 방법으로 고역 통과 필터를 사용할 수 있지만, 이 경우 원 영상에서의 고주파 노이즈 성분들을 함께 추출하는 단점을 가진다. 이러한 문제점을 최소화 하기 위해 1BT에서는 대역통과 필터를 이용하여 원 영상의 중간 주파수 영역을 가지는 에지를 사용한다[4]. 이를 위해, 먼저 원 영상의 효과적인 한 개의 비트 플레인 생성을 위한 다중 밴드패스 필터링된 영상과 원 영상을 비교한다. 원 영상에 대한 하나의 비트 플레인 획득은 다음의 식에 의해 결정된다.

$$B(i,j)=\begin{cases}1, & \text{if } I(i,j)\geq I_F(i,j)\\ 0, & otherwise,\end{cases} \tag{1}$$

식 (1)에서 I_F는 원 영상 I를 밴드패스 필터링한 결과 영상으로 다음 식의 커널 행렬과 원영상간의 컨볼루션(convolution)을 통하여 획득한다.

$$K(i,j)=\begin{cases}1/25, & \text{if } i,j\in\{0,4,8,12,16\}\\ 0, & otherwise,\end{cases} \tag{2}$$

1BT는 현재 영상과 참조영상에 대해 식 (1)을 적용하여 비트 플레인으로 변환 후, 블록의 정합을 위해 다음의 정합 기준을 사용한다.

$$NNMP_{1BT}(m,n)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}B^t(i,j)\oplus B^{t-1}(i+m,j+n) \tag{3}$$

식 (3)에서 $B^t(i, j)$와 $B^{t-1}(i, j)$는 각각 현재 영상과 참조영상으로부터 얻어진 비트 플레인들이고, $\oplus$는 Boolean exclusive-OR(XOR)연산을 나타낸다. $NNMP_{1BT}(m, n)$은 B^t와 B^{t-1}에 해당하는 블록 내 픽셀 간 서로 동일하지 않은 픽셀들의 수를 누적하여 블록 정합 결과를 획득한

다. 이를 통해, 1BT는 한 번에 여러 개의 픽셀들을 병렬로 연산할 수 있는 조합 논리 회로로 하드웨어를 구성할 수 있어 기존의 SAD에 비해 블록 정합 간 계산량과 하드웨어 복잡도를 현저히 저감할 수 있지만, 사용할 수 있는 데이터가 한정적이므로 비교적 블록 정합의 정확도가 낮은 단점을 가지고 있다.

2BT는 1BT의 블록 정합 정확도를 향상하기 위해 원 영상을 2-bit의 4개 표현레벨을 가지는 영상으로 변환하고, 1-bit의 데이터를 하나의 비트 플레인으로 나타내어 한 영상에 대해 두 개의 비트 플레인을 획득한다[7]. 이를 위해, 픽셀들의 평균값과 표준편차를 구한 후, 다음의 식을 통해 원 영상을 두 개의 비트 플레인으로 변환한다.

$$B_1(i,j) = \begin{cases} 1, & I(i,j) \geq \mu \\ 0, & otherwise \end{cases}$$
$$B_2(i,j) = \begin{cases} 1, & I(i,j) \geq \mu+\sigma \text{ or } I(i,j) \geq \mu-\sigma \\ 0, & otherwise \end{cases} \quad (4)$$

식(4)의 B_1, B_2 는 원 영상의 첫 번째와 두 번째 비트 플레인을 나타내고, μ와 σ는 각각 픽셀들의 평균값과 표준편차를 나타낸다. 그리고 다음의 식을 통해 두 영상의 비트 플레인 간 동일하지 않는 픽셀들의 수를 누적하여 블록의 정합 결과를 계산한다.

$$NNMP_{2BT}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left\{B_1^t(i,j) \oplus B_1^{t-1}(i+m,j+n)\right\} \| \left\{B_2^t(i,j) \oplus B_2^{t-1}(i+m,j+n)\right\} \quad (5)$$

식 (5)의 B_1, B_2, B_3와 B_4는 각각 현재 영상과 참조영상에 대한 첫 번째 비트 플레인과 두 번째 비트 플레인을 차례로 나타낸다. $NNMP_{2BT}$는 블록 정합 결과의 정확도를 향상하기 위해 두 개의 비트 플레인을 사용함에도 불구하고, 식 (5)의 OR연산을 사용하기 때문에 블록 정합결과가 표현할 수 있는 레벨의 수는 $NNMP_{1BT}$와 동일하다.

이외에도, 최근에 연구된 C-1BT(Constrained one-bit transform)방법은 블록 내 처리 픽셀과 함께 인접한 픽셀들의 정합 정도를 같이 평가할 수 있는 제약점을 주어 기존의 1BT방법의 적은 데이터 사용으로 인한 정합오류 문제를 상당히 향상하였다[6]. 이 방법은 영상에 따라 종종 기존 2BT방법보다 정확한 블록 정합 결과를 획득할 수 있지만, 1-bit의 제한된 데이터를 사용하기 때문에 블록 정합 정확도를 효과적으로 개선하는데는 한계가 있다.

또한, 2BT의 블록 정합결과를 보완하기 위해 $NNMP_{2BT}$에서 2차로 파생된 확장 연산자를 사용한 $NNMP_{2BT,SD}$[7] 방법과 $NNMP_{2BT}$의 정합 결과에 대한 표현범위 확장을 위해 $NNMP_{2BT}$에 사용되는 OR 연산대신 산술연산을 사용한 ENNMP와 비트 플레인에 가중치를 곱해주어 표현범위를 확장한 WNNMP(M, L)도 제안되었다[3]. 하지만 $NNMP_{2BT,SD}$ 방법은 2차 연산을 수행하기 위한 또 다른 프로세스가 추가되며, ENNMP, WNNMP(M, L)는 블록 정합 결과의 표현 범위는 증가시켰지만 실제로 정합 결과의 표현 가능한 레벨의 수는 증가되지 않아 블록 정합의 정확도 향상에는 제한적이다.

Ⅲ. 저 비트 절대 오차를 이용한 블록 정합 방법

기존의 SAD는 8-bit 데이터의 원 영상을 사용하여 블록을 정합하기 때문에 PE(processing element)안에 비트 연산을 하는 간단한 논리 연산기들과 함께 절대오차를 계산할 수 있는 8-bit 절대값 뺄셈기가 필요하다. 일반적으로, SAD를 계산하기 위해 다음의 식을 사용한다.

$$SAD(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|I^t(i,j) - I^{t-1}(i+m,j+n)|$$
$$where, -s+1 \leq m,n \leq s \quad (6)$$

식 (6)의 $I^t(i, j)$와 $I^{t-1}(i, j)$는 각각 현재 영상과 참조 영상에서의 (i, j)위치에 대한 픽셀값을 나타내고, s와 N은 블록의 검색 영역과 분할된 블록의 크기를 나타낸다. 기존 SAD와 달리, R-SAD는 원 영상의 표현 레벨 수를 줄여 블록 정합결과를 계산하기 때문에 BPM을 이용한 기존 방법들과 유사하게 논리 연산을 통해 정합 결과를 계산할 수 있으며, 이로 인해 기존 SAD에 비해 계산량과 하드웨어 복잡도를 줄일 수 있다. 또한, R-SAD는 비트 플레인의 수를 증가 시킬수록 정합결과의 정확도를 향상시킬 수 있지만, 본 논문에서는 하드웨어의 복잡도와 블록 정합 결과의 정확도를 고려하여 $R\text{-}SAD_{2\text{-}bit}$와 $R\text{-}SAD_{3\text{-}bit}$에 대해서 다룬다.

먼저, $R\text{-}SAD_{2\text{-}bit}$는 8-bit의 원 영상을 한 픽셀당 4개의 화소값을 가지는 2-bit영상으로 변환하고, 이들로부터 획득되는 두 개의 비트 플레인들을 이용하여 2-bit의 절대오차를 계산한다. 이를 위해 기존의 1BT[4]에서 사용한 비트 플레인 변환 과정에서 0과 1인 부분을 각각 두 영역

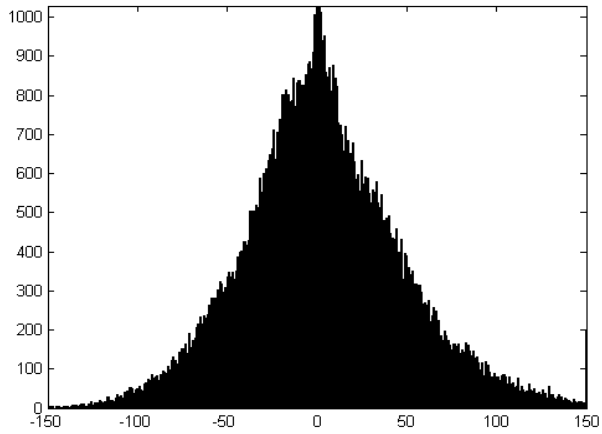


그림 1. Mobile 동영상 실험 셋의 한 프레임에 대한 E 히스토그램.

Fig. 1. E Histogram of an one frame for the Mobile test sequence.

으로 분할하기 위해 식 (1)의 $I_F(i, j)$를 좌항으로 이항하여 다음의 식과 같이 오차 영상을 정의한다.

$$E(i,j) = I(i,j) - I_F(i,j) \tag{7}$$

식 (7)의 E는 오차 영상을 나타낸다. 8-bit의 원 영상을 4개의 표현레벨을 가진 2-bit영상으로 변환하기 위해서는 3개의 임계값이 필요하다. 이를 위해, 우리는 다양한 테스트 영상들에 대한 오차영상의 히스토그램을 분석 하였다. 그 결과, 대부분의 실험 영상들이 0을 기준으로 좌우 대칭되는 오차 영상의 히스토그램 분포를 보였다. 그림 1은 테스트로 사용된 Mobile 동영상 셋의 한 프레임에 대한 오차 영상의 히스토그램을 나타낸 것이다. 그림 1과 같이 다양한 실험 영상들에 대한 오차 히스토그램들은 0을 기준으로 양과 음의 값을 가지는 픽셀들의 분포가 거의 동등함을 알 수 있었다. 따라서 중간 임계값은 오차 영상의 양과 음의 값을 구분하는 0으로 선택하였다. 또한, 상위 및 하위 임계값은 E 히스토그램의 양과 음의 값들의 평균값 또는 각 영역의 히스토그램 분포상에 median 값을 이용하여 상위와 하위 임계값을 결정할 수 있다. 본 논문에서는 적절한 임계값 설정을 위해 양과 음의 각 영역에 대한 평균값을 이용하여 상위와 하위 임계값을 결정하는 것을 추천하며, 다양한 영상들에 대한 E 히스토그램 분포에서 평균값을 이용한 방법을 통해 상위 및 하위 임계값을 각각 +30, -30으로 고정 할 수 있었다. 이를 통해, 원 영상을 비트 플레인으로 변환하는 계산량을 최소화할 수 있다. 이들 임계값들을 이용하여 8-bit입력 영상을 2-bit의 영상으로 변환하는 식은 다음과 같다.

$$T(i,j) = \begin{cases} 3, & \text{if } E(i,j) \geq M_+ \\ 2, & elseif\, 0 \leq E(i,j) \leq M_+ \\ 1, & elseif\, M_- \leq E(i,j) \leq 0 \\ 0, & otherwise \end{cases} \tag{8}$$

식 (8)에서 $T(i, j)$는 변환된 2-bit 영상을 나타내고, M_+와 M_-는 각각 상위 임계값과 하위 임계값을 나타낸다. 본 논문에서는 원 영상을 2-bit영상으로 변환 후 현재영상과 기준 영상간의 R-$SAD_{2\text{-bit}}$는 다음의 식에 의해 계산된다.

$$R\text{-}SAD_{2-bit}(m,n) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |T^t(i,j) - T^{t-1}(i+m, j+n)|, \quad -s \leq m,n \leq s \tag{9}$$

식 (9)에서 $T^t(i, j)$와 $T^{t-1}(i, j)$는 각각 현재 영상과 참조영상의 2-bit 변환 영상을 나타내고, $N\times N$, (m, n)과 s는 각각 블록의 크기, 참조 영상 내 블록의 변위, 그리고 블록의 검색 범위를 나타낸다. 식 (9)에서 입력영상으로 사용된 2-bit영상들은 식 (8)에 의해 각각 0~3까지의 값을 가지기 때문에 두 영상 T^t와 T^{t-1}사이에는 총 16가지의 입력 조합을 가진다. 이 경우 식 (9)의 누산을 제외한 절대오차에 의해 계산되는 최대값은 3이고 최소값은 0이므로 총 4가지의 출력 경우의 수를 가진다. 따라서 우리는 16가지의 입력 조합을 표현할 수 있는 4-bit의 입력(X_1, X_2, X_3, X_4)과 4가지의 출력값을 표현할 수 있는 2-bit의 출력(Y_1, Y_2)을 이용하여 조합 논리연산으로 식 (9)의 절대오차를 대신할 수 있다. 이 조합 논리 연산에 사용된 입력 값들은 0과 1만을 나타내는 1-bit값이므로, 식 (8)의 0~3의 값을 이진수로 표현할 경우 다음의 식과 같이 2개의 비트 플레인으로 나타낼 수 있다.

$$\begin{aligned} X_1(i,j) &= \begin{cases} 1, & \text{if } T^t(i,j) = 3 \text{ or } T^t(i,j) = 2 \\ 0, & otherwise, \end{cases} \\ X_2(i,j) &= \begin{cases} 1, & \text{if } T^t(i,j) = 3 \text{ or } T^t(i,j) = 1 \\ 0, & otherwise, \end{cases} \end{aligned} \tag{10}$$

식 (10)에서 X_1과 X_2는 각각 현재 영상의 2-bit영상에 대한 첫 번째와 두 번째 비트 플레인을 나타낸다. 참조영상에 대해서도 식 (10)을 이용하여 두 개의 비트 플레인 X_3와 X_4를 획득할 수 있다. 식 (9)의 절대오차 계산과 식 (10)의 비트 플레인 변환 식을 통해 우리는 4-bit의 입력과 2-bit의 출력에 대한 진리표를 표 1과 같이 획득할 수 있으며, 진리표의 입출력 관계를 Karnaugh map을 통해

표 1. 2-bit의 절대 오차를 계산하는 조합논리회로에 대한 진리표.
Table 1. Truth table for combinational logic circuit calculating 2-bit absolute difference.

X_1	X_2	X_3	X_4	Y_1	Y_2
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

다음의 식과 같이 간소화할 수 있다.

$$Y_1 = [(X_1 \cdot \overline{X_3}) \cdot (X_2 \| \overline{X_4})] \| [(\overline{X_1} \cdot X_3) \cdot (\overline{X_2} \| X_4)]$$
$$Y_2 = X_2 \oplus X_4 \tag{11}$$

식 (11)에서 $\|$, $\cdot$, $\overline{\ }$, $\oplus$ 는 각각 Bool 대수의 OR, AND, NOT 그리고 XOR 연산을 나타낸다. 식 (11)의 조합논리 연산을 통해 절대오차 계산 후, Y_1과 Y_2의 결과값이 누적되어 R-SAD$_{2\text{-bit}}$의 최종 블록정합결과가 된다. 식 (9)의 누산은 2-bit의 누산기를 통해 구현 가능하며, 이는 Half-adder의 조합이나 FPGA를 이용한 카운터 설계로 하드웨어 제작이 가능하다.

이와 함께, 우리는 원 영상을 3-bit의 영상으로도 변환이 가능하다. 이 경우, E 히스토그램에서 M_+와 M_-를 기준으로 양의 영역과 음의 영역을 균등하게 분할함으로써 2-bit에서 3-bit로 간단하게 확장이 가능하다. R-SAD$_{2\text{-bit}}$와 유사하게 R-SAD$_{3\text{-bit}}$는 64개의 서로 다른 입력 쌍과 8개의 출력 쌍으로 나타낼 수 있다. 따라서 R-SAD$_{2\text{-bit}}$와 동일한 방법으로 6-bit의 입력(X_1, X_2, ..., X_6)과 3-bit의 출력(Y_1, Y_2, Y_3)을 가지는 조합논리회로로 표현가능하다. R-SAD$_{3\text{-bit}}$는 R-SAD$_{2\text{-bit}}$에 비해 비교적 복잡한 논리 연산을 필요로 하기 때문에 하드웨어 복잡도가 다소 높지만 여전히 조합 논리 회로로 절대오차를 계산할 수 있기 때문에 빠르게 블록정합 결과를 계산할 수 있다.

이와 함께, 제안된 방법은 절대 오차합을 조합논리회로로 표현할 수 있을 뿐만 아니라 기존의 BPM보다 블록정합 결과를 표현 할 수 있는 레벨의 수가 많아 블록정합의 정확도가 비교적 높다. 왜냐하면 기존의 2BT는 입력으로 각 픽셀의 2개의 비트 플레인들을 통해 4개의 서로 다른 값을 사용할 수 있지만, NNMP$_{2BT}$의 계산 과정에서 두 비트플레인간 OR연산을 사용하기 때문에 블록 정합 결과의 표현 가능한 레벨 수는 N×N크기의 블록일 때 $0 \le NNMP_{2BT} \le N \times N$으로 1BT방법의 블록 정합 결과를 표현하는 레벨 수와 동일하다. 이와는 대조적으로 R-SAD$_{n\text{-bit}}$는 $N \times N$크기의 블록에 대해 $0 \le R\text{-}SAD_{n\text{-}bit} \le (2^n - 1) \times N \times N$ 의 표현레벨을 가지기 때문에 기존의 2BT방법뿐만 아니라 2BT방법을 확장한 비트 플레인 방법들보다 블록 정합결과를 표현할 수 있는 레벨의 수가 많다. 게다가 R-SAD는 기존의 SAD를 근사화한 방법으로 2BT방법의 비트 플레인 분할 방법과 달리 오차 히스토그램의 균등한 분할로 임계값 설정이 가능하기 때문에 원 영상에 대한 저 비트 변환 영상을 쉽게 획득할 수 있다. 따라서 제안된 방법은 2-bit와 3-bit의 변환보다 높은 n-bit의 변환도 쉽게 구현이 가능하다.

Ⅳ. 실험 결과 및 분석

R-SAD의 성능 평가를 위해 우리는 기존의 블록 정합 방법들과 제안된 블록 정합방법을 이용하여 움직임 보상(motion compensation)결과를 획득하고, 원 영상과 재생성된 영상간의 PSNR(peak signal to noise ratio)을 계산하여 성능을 비교 분석한다. PSNR은 다음의 식에 의해서 계산된다.

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} I(i,j) - I_r(i,j)^2} \tag{12}$$

식 (12)에서 H와 W는 각각 입력 영상의 세로와 가로의 픽셀수를 나타낸다. 이 실험을 위해 우리는 한 개의 CIF(352×288)포맷의 “Forman”과 네 개의 SIF(352×240)포맷의 ”Football”, ″Tennis”, ″Mobile”, ″Garden”의 영상 시퀀스들을 이용하여 실험결과를 획득하였다.

그림 2는 Tennis 영상 시퀀스들 중 한 프레임에 대한 움직임 보상 결과를 나타내고 있다. 그림 2에 표시된 상

표 2. 블록 크기가 16×16과 4×4이고, 검색 영역이 각각 16 픽셀과 4 픽셀로 설정할 때, 다양한 실험 영상에 대한 기존 방법들과 제안된 방법의 평균 PSNR(dB)결과.

Table 2. Average PSNR(dB) of several test image sequences of existing methods and proposed method, with block size of 16 X 16 and 4 X 4 pixels and search range of 16 and 4 pixels, respectively.

	Methods	Video Sequence(Frame size, Sequence length)					
		Forman (352 × 288) (300 frames)	Football (352 × 240) (125 frames)	Tennis (352 × 240) (150 frames)	Mobile (352 × 240) (300 frames)	Garden (352 × 240) (115 frames)	Avg
Block size (16×16) Search range (16 pixel)	SAD	32.11	22.88	29.45	23.94	23.79	**26.43**
	1BT	30.37	21.83	28.14	23.61	23.32	**25.45**
	2BT	30.71	22.11	28.47	23.65	23.42	**25.67**
	C-1BT	30.91	22.12	28.69	23.68	23.41	**25.76**
	$ENNMP_{2BT}$	30.96	22.35	28.76	23.78	23.54	**25.89**
	$WNNMP(M)_{2BT}$	30.81	22.21	28.77	23.70	23.46	**25.79**
	$WNNMP(L)_{2BT}$	30.88	22.36	28.69	23.77	23.46	**25.85**
	$R\text{-}SAD_{2\text{-}bit}$	31.22	22.44	28.98	23.83	23.59	**26.01**
	$R\text{-}SAD_{3\text{-}bit}$	31.63	22.82	29.22	23.90	23.76	**26.27**
Block size (4×4) Search range (4 pixel)	SAD	32.84	25.17	31.24	25.82	26.00	**28.21**
	1BT	28.80	22.26	27.75	22.90	22.19	**24.78**
	2BT	29.85	23.10	29.03	23.70	23.92	**25.92**
	C-1BT	29.28	22.67	28.65	23.40	22.86	**25.37**
	$ENNMP_{2BT}$	30.07	23.48	29.40	24.25	24.47	**26.33**
	$WNNMP(M)_{2BT}$	29.79	22.83	28.80	23.48	23.86	**25.75**
	$WNNMP(L)_{2BT}$	30.02	23.61	29.43	24.29	24.42	**26.35**
	$R\text{-}SAD_{2\text{-}bit}$	30.12	23.44	29.51	24.47	24.54	**26.42**
	$R\text{-}SAD_{3\text{-}bit}$	30.96	24.49	30.12	25.28	25.51	**27.27**

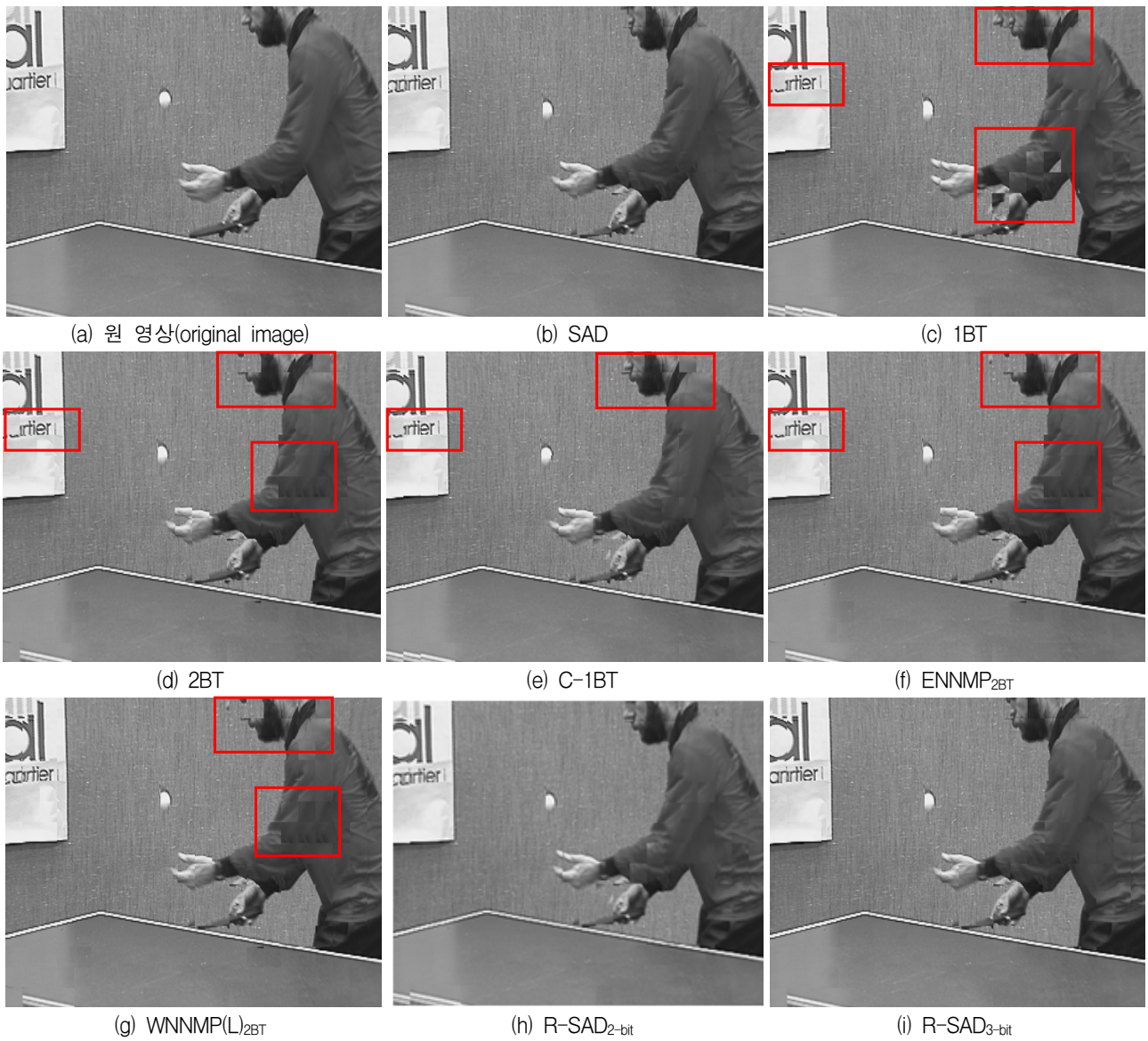
자는 움직임 추정이 정확하지 않아 움직임 보상 간 오차가 발생한 부분들을 표시하고 있다. 그림 2(b)는 8-bit영상에 대한 SAD의 결과로서 그림 2(a)의 원 영상과 유사하게 움직임이 보상된 결과를 획득할 수 있었다. 하지만, 그림2(c)의 1BT는 사람의 얼굴 부분과 팔 부분에서 움직임양을 정확하게 추정하지 못함으로써 다수의 오차들을 생성하였다. 그림 2(d)의 2BT는 1BT방법과 비교해 비교적 움직임 보상이 잘된 결과영상을 획득하였지만, 여전히 사람의 몸과 머리 부분에서 오차가 발생한 영역들을 관찰할 수 있었다. 그림 2(e)의 C-1BT는 한 개의 비트 플레인만을 사용했음에도 불구하고 2BT보다 움직임 보상 간 오차가 적게 발생했음을 보여준다. 하지만, 달력의 일부 글자가 보상 되지 못하였고, 몸과 머리 부분의 오차는 여전히 발생하였다. 그림 2(f)의 ENNMP는 2BT와 유사한 결과를 획득하였지만, 그림 2(g)의 WNNMP(L)은 달력의 글자 부분을 적절히 표현하였고, 기존 방법들 중 가장 좋은 움직임 보상결과를 획득할 수 있었다. 이들과는 대조적으로, 제안된 $R\text{-}SAD_{2\text{-}bit}$는 기존의 비트 플레인 방법들과 동일한 개수의 비트 플레인을 사용함에도 불구하고 오차가 많이 제거된 움직임 보상 결과를 획득할 수 있었다. 특히, $R\text{-}SAD_{3\text{-}bit}$는 영상의 표현 레벨이 현저히 감소되었음에도 불구하고, 기존의 8-bit 영상을 사용한 SAD와 유사한 결과를 획득할 수 있었다.

다양한 실험 영상에서의 움직임 보상에 대한 제안된 방법과 기존 방법의 성능 평가 결과를 표 2에 정리하였다. 표 2의 상위 부분은 블록의 크기가 16×16픽셀이고 검색 범위가 16픽셀일 때 움직임 보상을 통해 획득한 재생성된 영상 시퀀스들의 평균 PSNR을 나타내고 있다. C-1BT는그림 2(d)와 (e)의 실험결과와 유사하게 하나의 비트 플레인을 사용함에도 불구하고 16×16픽셀의 블록을 사용한 대부분의 실험결과에서 평균 PSNR이 2BT방법보다 좋은 결과를 획득하였다. 또한, 최근에 제안된 ENNMP와 WNNMP도 블록 정합 결과의 표현 범위를 확장하여 기존의 2BT방법보다 좋은 결과를 획득할 수 있었다. 이와 함께, 제안된 $R\text{-}SAD_{2\text{-}bit}$는 기존의 2BT기반의 블록 정합기준들 중 가장 좋은 성능을 보여

주었으며, 특히, R-$SAD_{3\text{-}bit}$의 PSNR값은 단지 3개의 비트 플레인 만을 사용했음에도 불구하고 모든 실험결과에서 8-bit 데이터를 사용한 기존 SAD와 유사한 PSNR을 가진다.

최근에는 움직임 보상 시 가변블록을 사용하여 움직임을 추정하거나 작은 블록의 조합으로 큰 블록의 움직임을 추정하기 때문에 작은 블록들에 대한 블록 정합 정확도 평가도 필요하다[9]. 표 1의 하위 부분은 4×4픽셀 크기의 블록과 4픽셀의 검색 영역에 대한 평균 PSNR결과를 나타내고 있다. C-1BT는 블록의 크기가 작아지면서 영상을 표현할 수 있는 레벨의 한계로 2BT보다 PSNR이 낮은 결과를 획득하였다. 그 외의 실험 결과에서는 전반적으로 16×16픽셀의 블록을 사용한 실험결과와 유사한 경향을 보였으며, 제안된 R-$SAD_{2\text{-}bit}$와 R-$SAD_{3\text{-}bit}$는 대체적으로 기존의 블록 정합방법들과 비교해 높은 평균 PSNR을 보이고 있어 비교적 적은 오차로 움직임 보상을 수행한 것을 알 수 있다.

이와 함께, 하드웨어 복잡도 측면을 평가하기 위해 2-bit의 값을 사용하는 BPM블록 정합 방법들 중 가장 간단한 하드웨어 구조를 가지는 2BT방법과 제안된 방법의



(a) 원 영상(original image) (b) SAD (c) 1BT

(d) 2BT (e) C-1BT (f) $ENNMP_{2BT}$

(g) $WNNMP(L)_{2BT}$ (h) R-$SAD_{2\text{-}bit}$ (i) R-$SAD_{3\text{-}bit}$

그림 2. 블록 크기가 16×16픽셀이고 탐색 영역이 16 픽셀일 경우, Tennis영상에 대한 움직임 보상결과.
Fig. 2. Result of motion compensation for the one of the frame for Tennis sequence set with block size of 16×16 pixels and search range of 16 pixels.

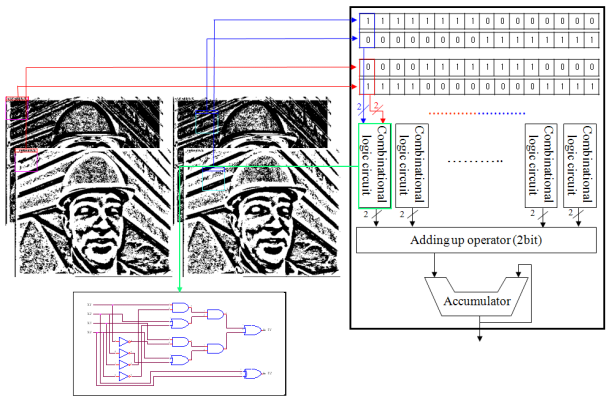


그림 3. 2BT의 NNMP를 계산하는 하드웨어 구조.
Fig. 3. Hardware structure for calculating NNMP in 2BT.

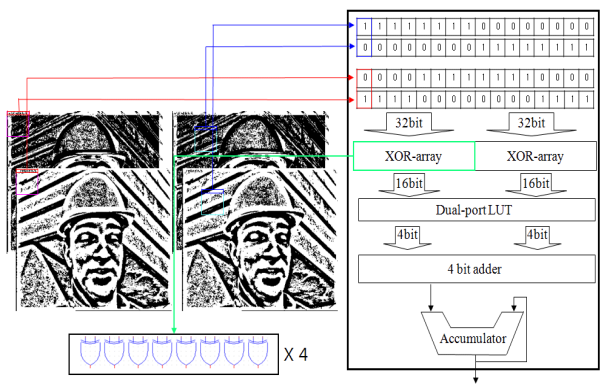


그림 4. 제안한 방법의 R-SAD$_{2\text{-bit}}$의 하드웨어 구조
Fig. 4. Hardware structure for calculating R-SAD$_{2\text{-bit}}$ in proposed method.

하드웨어 구조를 그림 3과 4에 각각 나타내어 비교하였다. 그림 3의 왼쪽 부분은 입력된 서로 다른 두 프레임을 2개의 비트 플레인으로 각각 변환한 결과 영상이며, 16×16블록의 정합 과정을 계산하기 위해 16×1의 블록정합 결과값을 16번 누적하여 최종 블록의 정합결과를 획득한다. 이를 위해 16-bit 두개의 입력을 받는 32개의 XOR gate 두 세트를 사용하며, 연산을 통해 계산된 총 16개의 픽셀 쌍들 중 서로 같지 않은 것들의 결과값들은 모두 1이 된다. 이후 0~16까지 4bit 데이터로 바꾸어주는 이중 룩업 테이블(Dual look-up table)의 입력으로 사용된다. 그리고 2개의 4-bit 덧셈기를 통하여 16×1의 블록정합 결과를 계산한 후 다음 처리기인 누산기(Accumulator)에 누적한다. 이 과정을 16번 반복하게되면 식 (5)의 두 블록 간의 NNMP를 계산할 수 있다.

그림 4는 그림 3과 비교해 2-bit의 값을 두 개 입력받을 수 있는 16개의 조합논리회로(combinational logic circuit)들이 존재하며, 조합논리회로를 거쳐 출력되는 값은 2-bit 의 값이며 16개의 2-bit값을 합하여 블록의 16×1에 해당하는 값을 계산할 수 있다. 이처럼 제안된 방법은 기존 2BT방법과 비교해 조합논리회로가 비교적 복잡한 단점이 있지만, 하드웨어 복잡도에 비해 상대적으로 높은 블록정합 정확도를 가진다. 최근에는 다양한 응용분야에서 움직임 추정 방법을 사용하는데 하드웨어의 복잡도 보다는 정확한 움직임 정보를 찾는 것을 보다 중요시하는 경향이 있다. 이러한 측면에서 제안된 방법은 기존의 다양한 2BT 방법들과 비교해 하드웨어 복잡도 대비 블록 정합 정합도가 높은 알고리즘으로 평가된다.

V. 결 론

제안된 블록 정합 방법은 2-bit또는 3-bit영상의 절대오차 합을 이용하는 방법으로 기존의 비트 플레인을 사용하는 블록 정합 방법과 같이 조합논리 회로로 구현이 가능하다. 이 방법은 기존의 비트 플레인 기반의 블록 정합방법들과 비교해 정합결과의 표현범위가 넓어 비교적 정확한 블록 정합결과를 획득할 수 있었으며, 특히, R-SAD$_{3\text{-bit}}$의 경우 8-bit의 원 영상을 사용하는 전형적인 SAD방법과 비교해 블록의 정합 정확도가 유사한 실험결과들을 획득할 수 있었다. 이처럼 제안된 블록 정합방법은 비교적 높은 정확도의 블록 정합결과를 획득할 수 있음에도 불구하고, 조합 논리 회로로 구현이 가능하기 때문에 하드웨어 복잡도가 낮은 장점을 가지고 있다.

REFERENCES

[1] B. J. Tippetts, D. J. Lee, J, K. Archibald and K. D. Lillywhite, “Dense disparity real-time stereo vision algorithm for resource-limited systems,” *IEEE trans. Circuits and Syst. Video Technol*, vol. 21, no. 10, pp. 1547-1555, 2011

[2] C. Y. Choi, and J. C. Jeong, “Enhanced two-bit transform based motion estimation via extension of matching criterion,” *IEEE trans. Consumer Electron*, vol. 56, no. 3, pp. 1883-1889, 2010..

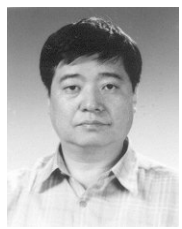
[3] Z. L. He, C. Y. Tsui, K. K. Chan, and M. L. Liou, “Low-power VLSI design for motion estimation using adaptive pixel truncation,” *IEEE trans. Circuits and Syst. Video Technol*, vol. 10, no. 8, pp. 669-678, 2000.

[4] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE trans. Circuits and Syst. Video Technol*, vol. 7, no. 4, pp. 702-706, 1997.
[5] A. Erturk and S. Erturk, "Two-bit transform for binary block motion estimation," *IEEE trans. Circuits and Syst. Video Technol*, vol. 15, no. 7, pp. 938-946, 2005.
[6] O. Urhan and S. Erturk, "Constranied one-bit transform for low complexity block motion estimation," *IEEE trans. Circuits and Syst. Video Technol*, vol. 17, no. 4, pp. 478-482, 2007.
[7] N. J. Kim, S. Erturk and H. J. Lee, "Two-bit transform based block motion estimation using second derivatives," *IEEE trans. Consumer Electron*, vol. 55, no. 2, pp. 902-910, 2009.
[8] Y. K. Ko, H. C. Oh, and S. J. Ko, "VLSI design for motion estimation based on bit-plane matching", *Journal of the Institute of Electronics and Information Engineers of Korea*, vol. 38, no. 5, pp. 57-65, 2001.
[9] Y. B. Jang, S. M. Oh, B. C. Kim, and H. J. Yoo, "Efficient SAD processor for motion estiamtion of H.264", *Journal of the Institute of Electronics and Information Engineers of Korea*, vol. 44, no. 2, pp. 74-81, 2007.

저 자 소 개

김 상 철(정회원)
2006년 경일대학교 전자공학과 학사 졸업
2008년 경북대학교 전자공학과 석사 졸업
2013년 경북대학교 전자전기 컴퓨터학부 박사 졸업
2013년 1월～현재 ㈜ LG전자 TV 연구소 선임연구원
<주관심분야 : 디스플레이 화질개선, 영상처리, 컴퓨터 비전>

박 순 용(정회원)
1991년 경북대학교 전자공학과 학사 졸업
1993년 경북대학교 전자공학과 석사 졸업
2003년 미국 뉴욕주립대 스토니 브룩 박사 졸업
2005년 2월～현재 경북대학교 IT대학 컴퓨터학부 부교수
<주관심분야 : 3차원 컴퓨터비전, 로봇비전>



진 성 일(정회원)-교신저자
1977년 서울대학교 전자공학과 학사 졸업
1981년 한국과학기술원 전자공학과 석사 졸업
1988년 Carnegie Mellon 대학교 박사 졸업
1977년～1978년 (주)대영전자공업 연구원
1982년～현재 경북대학교 IT대학 전자공학부 교수
<주관심분야 : 영상처리, 컴퓨터비전, 패턴인식>