

논문 2007-44SD-5-9

# H.264 동영상 압축에서의 가변 블록과 다중 프레임을 지원하는 효율적인 움직임 추정 방법

( An Efficient Motion Estimation Method which Supports  
Variable Block Sizes and Multi-frames for  
H.264 Video Compression )

윤 미 선\* , 장 승 호\* , 문 동 선\* , 신 현 철\*\*

( Misun Yoon, Seungho Chang, Dongsun Moon, and Hyunchul Shin )

## 요 약

휴대용 단말기에서의 멀티미디어 정보의 이용이 일반화 되면서 동영상 압축 등의 데이터 처리를 위한 연산량이 크게 증가하고 있다. 특히, 휴대용 단말기에서의 소모 전력 감소와 실시간 처리를 위한 다양한 연구가 진행되고 있다. 본 연구에서는 H.264 인코더에서 67% 이상의 많은 연산량을 차지하는 움직임 추정기를 새로운 구조로 설계하였다. 근사적인 SAD를 사용하여 움직임을 예측하고 Alternative Row Scan (ARS) 방법을 개발하였으며, DAU, FDVS 알고리즘을 사용하여, JM8.2에 제시된 다중 프레임 움직임 추정보다 연산량을 평균적으로 75% 이상 감소시켰다. 또한, 다중 프레임 움직임 추정에서의 적절한 참조 프레임 수와 참조 프레임 블록의 크기를 결정하여 연산량을 감소시키면서도 화질 열화가 거의 없도록 하였다. iPROVE 검증틀을 사용하여 하드웨어/소프트웨어 Co-Simulation을 수행하여 제안한 움직임 추정기가 정상 동작함을 확인하였으며, 50MHz에서 30 CIF frames/sec가 처리됨을 확인하였다.

## Abstract

As multimedia portable devices become popular, the amount of computation for processing data including video compression has significantly increased. Various researches for low power consumption of the mobile devices and real time processing have been reported. Motion Estimation is responsible for 67% of H.264 encoder complexity. In this research, a new circuit is designed for motion estimation. The new circuit uses motion prediction based on approximate SAD, Alternative Row Scan (ARS), DAU, and FDVS algorithms. Our new method can reduce the amount of computation by 75% when compared to multi-frame motion estimation suggested in JM8.2. Furthermore, optimal number and size of reference frame blocks are determined to reduce computation without affecting the PSNR. The proposed Motion Estimation method has been verified by using the hardware and software Co-Simulation with iPROVE. It can process 30 CIF frames/sec at 50MHz.

**Keywords :** H.264 encoder, Video compression, Motion estimation, Absolute Difference, FPGA.

## I. 서 론

\* 학생회원, \*\* 평생회원, 한양대학교 전자전기 제어  
계측학과

(Electronic Engineering and Computer Science,  
Hanyang University)

※ 본 연구는 한국과학재단 특정기초연구  
(R012004000102680) 지원으로 수행되었음.

접수일자: 2007년3월8일, 수정완료일: 2007년4월24일

최근 디지털 컨버전스 (Digital Convergence)가 이루어지면서 디지털 멀티미디어에 다양한 기능이 통합되고 있다. 특히, 전력과 하드웨어 면적의 제약이 존재하는 휴대용 단말기에서는 멀티미디어의 다양한 기능을 실시

간 처리해야 하므로 데이터를 압축시키고, 하드웨어 복잡도를 최적화 할 수 있는 기술 개발이 필수적이다. 이에 따라 ITU-T에서는 H.264 등의 국제 규격들을 제정하였다<sup>[1]</sup>. 그중 H.264는 효율성과 신뢰성을 강조하고 있으며, MPEG-2, MPEG-4 (Part 2) 등 기존 동영상 압축 표준에 비해 1.5배 이상의 압축 성능을 갖는다<sup>[2]</sup>. 일반적으로 동영상 압축 방법에는 시간적 중복성과 공간적 중복성을 제거하는 방식이 있으며, H.264에서는 프레임의 상관관계를 이용한 움직임 예측을 통해 시간적 중복성을 제거하는 움직임 추정 (Motion Estimation)을 사용한다. 움직임 추정은 H.264 인코더의 67%이상의 연산량을 차지하는 부분으로 연산량을 감소시키고 고속 처리하기 위한 연구가 필요하다<sup>[5]</sup>.

움직임 추정 방법은 추정의 기본 단위에 따라 크게 화소 순환 알고리즘 (Pel Recursive Algorithm)과 블록 정합 알고리즘 (Block Matching Algorithm)으로 나뉘어진다. 현재 많은 비디오 부호화에서 사용되는 블록 정합 알고리즘은 화소 순환 알고리즘에 비해 데이터 흐름의 규칙성, 계산의 복잡도, 하드웨어의 구현 등에서 용이하기 때문에 움직임 벡터 (Motion Vector) 추정에 많이 사용되고 있다<sup>[6]</sup>.

본 연구에서는 움직임 추정기의 효과적인 스캔방법인 Alternative Row Scan (ARS) 방법을 개발하고, 근사 SAD를 이용한 움직임 예측을 사용하였으며, 기존의 Disable Approximation Unit (DAU)<sup>[3]</sup>와 Forward Dominant Vector Selection Method (FDVS)<sup>[4]</sup>를 통합 이용하여, JM8.2에 제시된 다중 프레임 움직임 추정 알고리즘 보다 연산량을 75% 이상 감소시켰다.

II절에서는 움직임 추정기, 가변블록 움직임 추정, 다중 프레임 움직임 추정과 DAU<sup>[3]</sup>에 대한 기존 연구에 대해 기술한다. III절에서는 제안한 움직임 추정기 구조, Alternative Row Scan (ARS) 방법과 FDVS를 이용한 다중 프레임 처리 방법을 기술하고, IV절에서는 제안한 움직임 추정기의 검증 및 평가에 대해 설명한다. 마지막으로 V절에서 결론을 내린다.

## II. 기존 연구

### 1. 움직임 추정기

움직임 추정은 현재 프레임 블록과 참조 프레임의 블록에서 Sum of Absolute Difference (SAD) 연산을 통해 정합 오차가 가장 작은 블록을 찾는 방법을 움직임 추정이라고 하며, 움직임 변위를 움직임 벡터라고 한다.

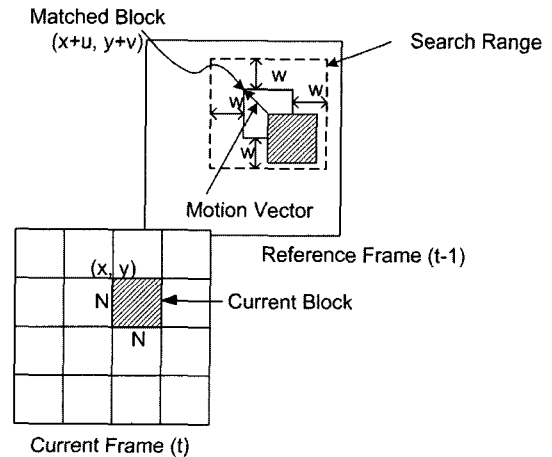


그림 1. 블록 정합 움직임 추정 방법  
Fig. 1. Block Matching Motion Estimation Method.

그림 1에서  $N \times N$ 은 블록의 가로와 세로 크기이며,  $(x, y)$ 는 현재 프레임에서의 기준 블록의 위치이다. 또한,  $(x+u, y+v)$ 는 참조 프레임의 탐색 영역에서 현재 프레임의 기준 블록과 유사도가 가장 높은 블록의 위치이며,  $(u, v)$ 는 움직임 벡터가 된다.

식 (1)에서 움직임 추정과 보상의 블록 사이즈는  $N \times N$  샘플이고,  $c_{ij}$ 와  $r_{ij}$ 는 현재 프레임과 참조 프레임의  $(i, j)$  위치의 휘도 값을 의미한다.

$$SAD = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |c_{ij} - r_{ij}| \quad (1)$$

### 2. 가변블록 움직임 추정

H.264는 그림 2와 같이 움직임 벡터를 찾기 위해 다양한 크기의 블록을 허용한다. 이는 이동뿐만 아니라 회전 및 다양한 국부적 움직임 추정에 적합하며, 압축 효율을 높일 수 있다는 장점이 있다. 표준안에 정의된 가변블록 크기는  $4 \times 4$ ,  $4 \times 8$ ,  $8 \times 4$ ,  $8 \times 8$ ,  $8 \times 16$ ,  $16 \times 8$ ,  $16 \times 16$  일곱 가지로, 움직임 벡터를 41개의 경우로 나

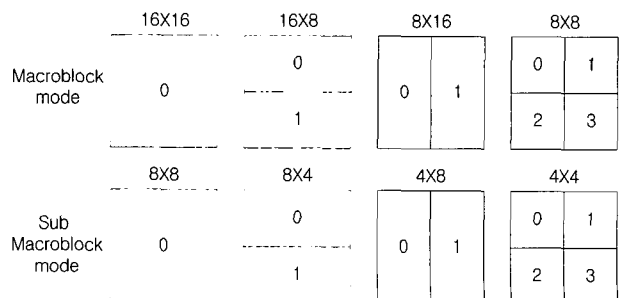


그림 2. H.264의 가변 블록 모드  
Fig. 2. H.264 Variable Block Mode.

타낼 수 있으며, Macroblock mode인 16×16, 16×8, 8×16, 8×8과 Sub-Macroblock mode인 8×8, 8×4, 4×8, 4×4블록 등으로 구성된다. 일반적으로 움직임이 단조로운 화면에서는 16×16 모드와 같이 큰 블록 단위의 움직임 보상이 이루어질 확률이 높고, 움직임이 세밀한 부분은 4×4와 같은 작은 블록 단위로 움직임 보상이 이루어질 확률이 높다.

### 3. 다중 프레임 움직임 추정

다중 프레임 움직임 추정은 이전의 참조 프레임 중 현재 블록과 유사도가 가장 높은 블록을 찾아 이를 움직임 보상에 사용하는 것이다. 그림 3은 다중 프레임 움직임 추정을 나타낸 것이다. 여러 장을 참조프레임 중 현재 블록과 가장 일치도가 높은 블록으로 움직임 보상하는 것이다. 따라서 반복되는 영상의 압축 효율을 높이기 위하여 다중 프레임을 사용하면 비트율을 낮출 수 있다. H.264에서는 기존 코덱보다 영상 압축을 강화하였으며, 5 프레임까지의 참조 프레임을 지원하여 보다 효율적인 움직임 추정을 할 수 있도록 하였다. 특히, 연속되는 움직임이 많은 영상, 반복적, 규칙적인 움직임을 갖는 영상의 경우, PSNR (Peak Signal-to-Noise Ratio)가 최대 1.0 dB까지 향상된다[JVT02]. 그러나 참조 프레임을 하나만 사용하는 기존의 영상 압축 코덱보다 많은 연산량을 필요로 하며, H.264 코덱의 전체 복잡도에서 움직임 추정이 대부분을 차지하는 주된 요인이 된다<sup>[8]</sup>.

따라서 이러한 많은 연산량을 줄이기 위한 알고리즘이 개발되었다. 그 중 본 연구에서는 Forward Dominant Vector Selection Method (FDVS)<sup>[4]</sup> 응용하여 사용하였다. FDVS는 움직임 벡터를 구할 때 여러 개의 참조 프레임 중 현재 매크로 블록과 가장 유사한 매크로 블록의 움직임 벡터를 선택하는 알고리즘이다.

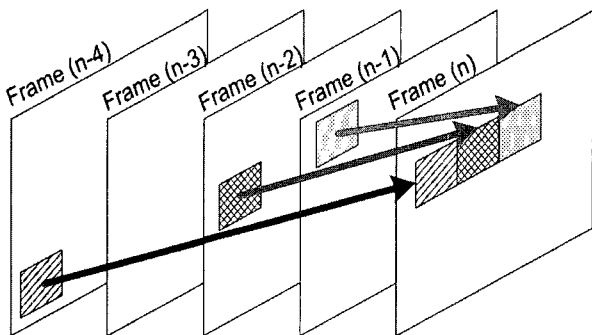


그림 3. 다중 프레임 움직임 추정  
Fig. 3. Multi-frame Motion Estimation.

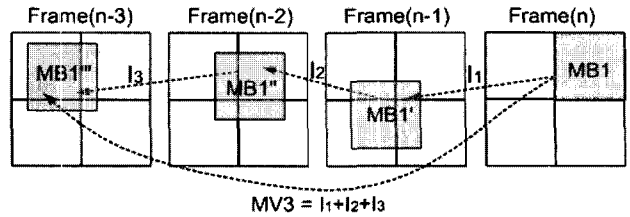


그림 4. FDVS 알고리즘  
Fig. 4. FDVS Algorithm.

그림 4에서 보이는 것과 같이 (n-1)번째 프레임, (n-2)번째 프레임과 (n-3)번째 프레임의 움직임 벡터를 각각 I1, I2, I3이라 하면, (n)번째 프레임에서 (n-3)번째 프레임으로의 움직임 벡터 MV3은 I1 + I2 + I3로 구해질 수 있다. 이 방법을 사용하여 (n-3)번째 프레임의 움직임 벡터는 다시 추정하는 과정을 거치지 않고 움직임 벡터를 효율적으로 구할 수 있다.

### 4. Disable Approximation Unit (DAU)

[3]에서 DAU 블록은 연산량을 줄여 전력 소모 감소를 목적으로 설계하였다<sup>[3]</sup>. 대략적인 SAD인  $\hat{D}(u,v)$ 를 계산하여 현재까지 탐색한 최소 SAD인  $D_{\min}(V_{ssf})$ 와 비교한 후 현재 탐색점의 정합오차 계산 필요 여부를 결정함으로써 전력 소모를 줄일 수 있다.

식 (2)는 일반적인 SAD 연산식을 나타내며, 식 (3)은 DAU에서 연산되는 대략적인 SAD 연산식을 나타낸다.  $(u,v)$ 는 현재 프레임과 참조 프레임 사이의 상대적인 좌표로, 움직임 벡터를 나타낸다.  $r(i,j)$ 는 현재 프레임에서 좌표  $(i,j)$ 의 휘도성분을 나타내고,  $s(i+u,j+v)$ 는 참조 프레임의 탐색창의 좌표  $(i+u,j+v)$ 의 휘도 성분을 나타낸다.

$$D(u,v) = \sum_{j=1}^N \sum_{i=1}^N |s(i+u,j+v) - r(i,j)| \quad (2)$$

$$\hat{D}(u,v) = \sum_{j=1}^{N-1} \left| \sum_{i=1}^N s(i+u,j+v) - \sum_{i=1}^N r(i,j) \right| \quad (3)$$

식 (2)와 식 (3)의 관계는 식 (4)로 표현할 수 있다.

$$\hat{D}(u,v) \leq D(u,v) \quad (4)$$

탐색 영역 전체  $V$ 에 대한 최소정합오차를  $D_{\min}(V)$ 라 하고 현재까지 탐색한 부분을  $V_{ssf}$ 라 하면 식 (5)가 성립됨을 알 수 있다.

$$D_{\min}(V) \leq D_{\min}(V_{ssf}) \leq \hat{D}(u,v) \leq D(u,v) \quad (5)$$

따라서  $D_{\min}(V_{ssf}) < \hat{D}(u,v)$ 인 경우에는 더 이상의 SAD 연산은 불필요하므로 PE에 Disable 신호를 보내고, 그 이후의 계산은 하지 않게 된다. 이를 이용하여 기존의 Full Search 방법 보다 57%이상의 연산량 감소 효과를 보였으며, 화질열화는 발생하지 않았다(표 1, 표 2 참조).

### III. 제안한 움직임 추정 기법

#### 1. 움직임 추정기

제안한 H.264 인코더의 움직임 추정기 구조는 가변 블록을 지원하는 1D Systolic Array 구조를 사용하였다<sup>[9]</sup>. 1D Systolic Array 구조는 2D Systolic Array 구조에 비하여 유연한 구조를 가지고 있으며, 작은 면적으로 구현이 가능하다. 또한, 가변블록을 지원하기 위해 움직임 벡터를 찾기 위한 SAD를 구하는 계산 과정은 최소 단위인 4×4 크기의 SAD를 누적시킴으로써 각각의 크기에 맞는 SAD 값을 구하게 된다. 제안한 전체적인 구조는 그림 5에서 보이는 것과 같다. 제안된 구조는 크게 제어 블록과 16개 PE 블록과 DAU로 나누어진다.

그림 5에서 현재 블록에 대한 입력은 8 bits의 1개의 입력을 받고 쉬프트 레지스터를 통하여 각각의 PE에 입력된다. 탐색창에서의 참조 블록에 대한 데이터는 8 bits의 2개의 버스를 통하여 입력된다. 각각의 신호는 동시에 전체의 PE에 공급되게 되며, 각각의 PE는 2개의 참조 블록 데이터 중 MUX를 통하여 1개의 데이터를 선택하여 PE의 입력으로 사용하게 된다. PE의 출력

은 SAD BUS0~SAD BUS12까지 13개의 버스에 나누어 출력하게 된다. 최종적으로 각각의 SAD BUS는 최소의 SAD 값을 찾아 움직임 벡터를 출력한다<sup>[9]</sup>.

본 연구에서는 [3]을 응용하되 DAU를 연산량 감소를 위해 사용할 뿐만 아니라, 탐색영역의 움직임 벡터를 예측하여 탐색창에서 SAD 연산의 시작 위치를 결정하기 위해 사용한다. 탐색영역의 움직임 벡터를 예측하기 위해  $\hat{D}(u,v)$ 를 계산하여, 최소의  $\hat{D}(u,v)$ 가 위치하는 Row부터 Processing Element (PE)에서 SAD연산을 시작한다. 또한, PE에서의 연산량 감소를 위해 식 (4)가 사용되는데 이를 위해 연산된  $\hat{D}(u,v)$ 의 저장이 필요하다.  $\hat{D}(u,v)$ 는 다중 프레임 움직임 추정을 지원하기 위해  $31 \times 31 \times 8 = 7.6\text{Kb}$ 와  $21 \times 21 \times 8 = 3.5\text{Kb}$ 를 사용하고 병렬처리하기 위해 Double buffering하여 총 22.2Kb를 사용하였다. 제안된 움직임 추정기의 처리속도는  $352 \times 288$ 의 CIF 사이즈 이미지를 30 Frames/sec로 처리 가능하고 이중에서 DAU 블록은 10.4%를 차지한다. 동작 주파수는 70MHz이며 JM8.2에서 제시된 다중 프레임 움직임 추정기에 비해 화질열화 없이 75%이상의 연산량을 감소시켰다.

#### 2. Alternative Row Scan (ARS)

동영상에서 최소의 SAD값은 대부분 탐색 영역 원점 가까이에 분포하며, 최소 SAD를 빠른 시간 내에 찾게 되면 많은 SAD 연산량을 감소시키고, 고속처리가 가능하다. 이를 이용하여 움직임 추정에서 연산량을 감소시키기 위한 다양한 방법들이 개발되었다<sup>[1]</sup>. [11]에서는

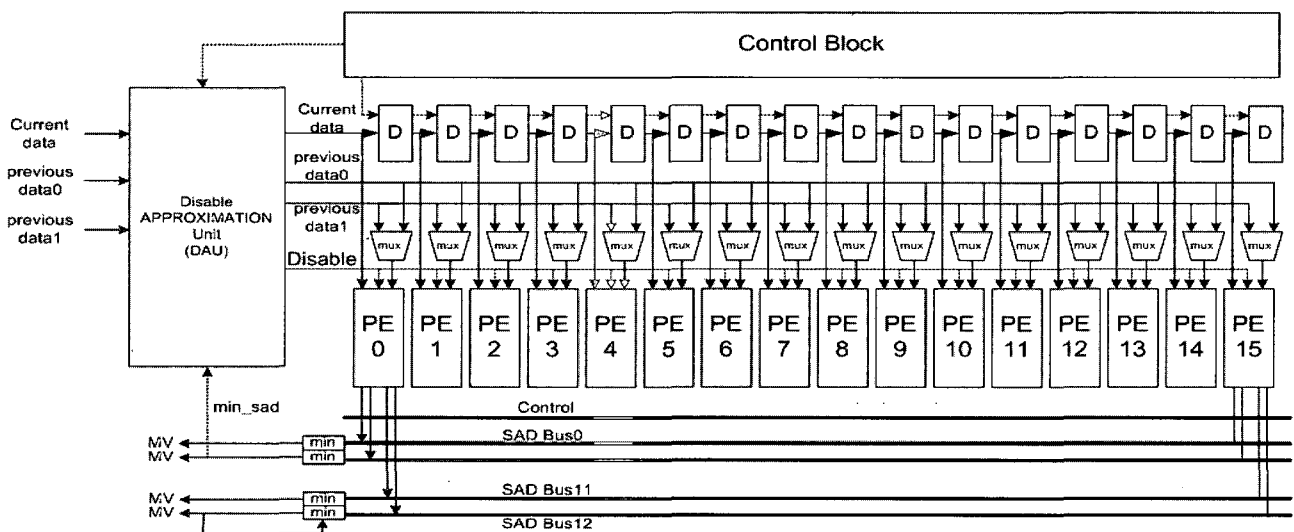


그림 5. 제안한 움직임 추정기 구조  
Fig. 5. Proposed Motion Estimation Architecture.

표 1. 제안한 Alternative Row Scan (ARS) 방법과 기존 스캔 방법의 연산량 비교

Table 1. Comparisons of the amount of computation for several typical full search methods.

	Raster scan (JM8.2)	DAU <sup>[3]</sup>	Spiral with SSDA <sup>[11]</sup>	ARS with all features	ARS without		
					Alternating	Prediction	SSDA
akiyo	103,809,024	41,418,812	32,634,688	25,423,588	26,297,664	25,742,237	28,493,824
news	103,809,024	44,161,744	32,120,896	26,674,720	27,645,200	26,961,807	29,890,560
forman	103,809,024	44,183,604	33,737,932	24,742,812	26,152,052	29,345,132	27,377,664
container	103,809,024	49,752,408	33,667,584	24,661,100	24,925,284	24,905,031	27,540,480
mother-daughter	103,809,024	46,302,816	37,567,168	27,282,280	30,609,740	30,926,884	28,790,784
average (%)	100 % (403.03 %)	43.49 % (175.35 %)	32.70 % (131.79 %)	24.81 % (100 %)	26.13 % (105.32 %)	26.56 % (107.06 %)	27.38 % (110.34 %)

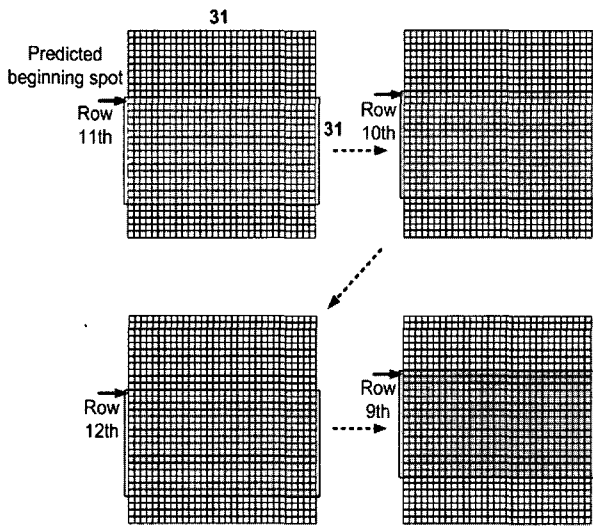


그림 6. 제안한 Alternative Row Scan  
Fig. 6. Proposed Alternative Row Scan.

움직임 예측을 통해 움직임 예측 위치부터 Spiral Scan Full Search하고 있다. 그러나 Spiral Scan 방법은 탐색 방향을 결정하기 위한 복잡도가 높은 단점이 있다. 따라서 본 연구에서는 하드웨어 복잡도를 낮추고 연산량을 최소화하기 위해 Raster Scan Full Search를 사용하였다. Raster Scan 방법은 Spiral Scan 방법에 비해 데이터의 흐름이 규칙적이므로 하드웨어 구현이 간단한 장점이 있다. 또한, 본 논문에서는 DAU를 이용하여  $\hat{D}(u,v)$ 가 최소가 되는 위치를 움직임 벡터로 예측하고, 최소의  $\hat{D}(u,v)$ 가 위치하는 Row부터 Alternative 순서로 SAD를 연산하는 Alternative Row Scan (ARS) 방법을 제안한다. 그림 6은 예측된 움직임 벡터가 11번째 Row인 경우의 예이다. 즉, 예측된 움직임 벡터가 11번째 Row부터이므로 SAD 연산을 시작하여 Rows 11 to 26, Rows 10 to 25, Rows 12 to 27, Rows 9 to 24

등의 순서로 탐색해간다.

표 1은 기존의 다양한 데이터 스캔 방법과 본 연구에서 새롭게 제안한 ARS의 연산량을 비교한 것이다. 연산량은 알고리즘을 C언어로 구현하였으며, 덧셈과 뺄셈을 1, 절대값은 연산과정에 if문이 포함되어 있기 때문에 2의 가중치를 주어 연산량을 계산하여 비교하였다. Raster Scan은 JM8.2의 Full Search 방법을 나타내고, Spiral with SSDA 방법은 [11]에서 제안한 스캔 방법이다. [11]에서 제안한 스캔 방법은 Spiral Full Search를 하되 Sequential Similarity Detection Algorithm (SSDA)을 적용한 알고리즘이다. DAU, Spiral with SSDA, 제안한 ARS 스캔 방법은 기존의 Full Search 방법 중 하나인 Raster Scan 방법에 비해 각각 약 56%, 약 67%, 약 75%의 연산량을 감소시킬 수 있으며, 본 논문에서 제안한 ARS가 가장 많은 연산량 감소효과가 있음을 알 수 있었다. 또한, ARS에 적용된 알고리즘의 효과를 알아보기 위한 연산량 비교를 표 1에 나타내었다. 그 결과 Disabling SSDA, prediction과 alternating scan 방법은 ARS에 비해 각각 10%, 7%, 5%의 연산량 증가가 나타났다. 표 2는 Full Search (Raster scan, Spiral scan), DAU<sup>[3]</sup>, Global Elimination Algorithm (GEA)<sup>[10]</sup>과 제안한 ARS 방법의 PSNR을 비교한 것이다. 그 결과 DAU와 제안한 ARS는 화질 열화 없이 연산량을 감소시킬 수 있었으며, GEA는 약 0.15 dB의 화질 열화가 발생하였다. GEA는 [10]에서 제안한 Fast block matching 알고리즘이다.

### 3. FDVS를 이용한 다중 프레임 처리

다중 프레임을 사용하여 움직임을 예측하는 것은 압축 효율을 높이는 장점이 있다. 하지만 이전프레임인  $F_{n-1}$ 에서 움직임 벡터를 찾는 것과 같은 방법으로  $F_{n-2}$

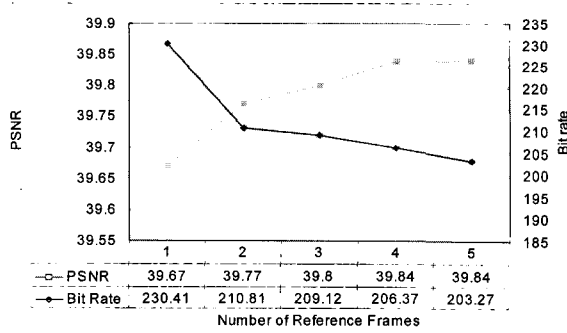


그림 7. 참조 프레임 수에 따른 PSNR과 Bit Rate  
Fig. 7. PSNR and Bit Rate of Multi-frame.

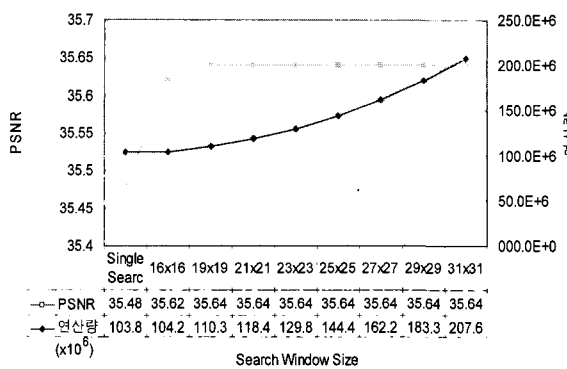


그림 8. 탐색 영역 크기에 따른 PSNR과 Bit Rate  
Fig. 8. PSNR and Bit Rate of Search Windows.

에서도 움직임 벡터를 찾으면 연산량이 참조 프레임 수에 비례해서 늘어나는 단점이 있다. 또한 필요한 메모리의 용량이 참조 프레임의 수와 비례해서 증가하게 된다. 따라서 본 연구에서는 H.264에서 제공되는 5장의 참조 프레임에서 연산량과 PSNR 및 Bit rate의 tradeoff를 고려한 참조 프레임 수와 각 참조 프레임의 참조 블록의 크기를 결정하였다.

그림 7은 참조 프레임 수에 따른 PSNR과 Bit Rate를 나타낸 것이다. 그림 7에서도 볼 수 있듯이, 연산량, 메모리의 용량, PSNR, Bit rate의 tradeoff를 고려하였을 때, 2장의 참조 프레임을 사용하는 것이 합리적임을 알 수 있다.

그림 8은 FDVS 알고리즘을 C언어로 구현하여 탐색 영역 크기에 따른 PSNR과 연산량을 나타낸 것이다. 그림 8의 Single Search는 다중 프레임 처리를 하지 않은 경우를 의미하며, 31x31은 H.264에서 제공하는 기존 다중 프레임 처리를 의미한다. 기존 연구 [4]에서는 탐색 영역을 16x16으로 사용하였으나, 본 연구에서는 그림 8에 보이는 바와 탐색 영역 크기에 따른 PSNR과 연산량을 도식화 하여, 화질 저하가 없는 21x21의 탐색

표 2. 제안한 Alternative Row Scan 방법과 기존 스캔 방법의 PSNR

Table 2. Comparisons of the PSNR for several typical search methods.

	Full Search	DAU <sup>[3]</sup>	GEA <sup>[10]</sup>	ARS with all features
akiyo	38.08	38.08	37.95	38.08
news	35.24	35.24	35.19	35.24
forman	35.23	35.23	34.81	35.23
container	35.34	35.34	35.24	35.34
mother-daughter	40.46	40.46	40.42	40.46
average (%)	36.87	36.87	36.72	36.87

영역을 사용하였다. 따라서 제안한 구조에서는 2개의 참조 프레임을 사용하며, 첫 번째 참조 프레임에서는 31x31의 탐색영역을 사용하고, 두 번째 참조 프레임에서는 21x21 탐색영역을 이용하여, 여러 참조 프레임을 사용하지 않고도 일반적인 다중 프레임을 사용한 효과를 얻을 수 있도록 하였다. 또한, FDVS 알고리즘을 사용하면 연산량을 감소시킬 수 있다. FDVS를 이용하여 움직임 추정기에서 이미 구한 움직임 벡터를 저장하여 재사용한다. 그림 8의 Single Search의 연산량은 103,809,024이고, 21x21 탐색영역의 연산량은 118,407,168이므로 FDVS 알고리즘을 이용한 다중 프레임 처리는 다중 프레임을 처리하지 않은 경우보다 약 14%의 연산량 증가만으로 다중 프레임 처리 지원이 가능하다.

#### IV. 검증 평가

본 연구에서 제안한 가변 블록과 다중 프레임을 지원하는 움직임 추정기는 JM8.2를 사용하여 성능을 검증·평가 하였으며, 소프트웨어와 하드웨어 통합 검증 툴인 iPROVE를 이용하여 하드웨어의 정상 동작함을 검증하였다. 본 연구에서 제안한 움직임 추정기는 VerilogHDL 설계하여 iPROVE의 Xilinx Virtex2 FPGA에서 맵핑시키고, 움직임 추정을 제외한 블록은 PC를 통해 C Level에서 처리하였다. iPROVE를 이용한 하드웨어/소프트웨어 Co-Simulation하여 제안한 움직임 추정기가 정상 동작함을 확인하였다. 움직임 추정기는 게이트 카운트한 결과 90,589 gates였으며, 그 중

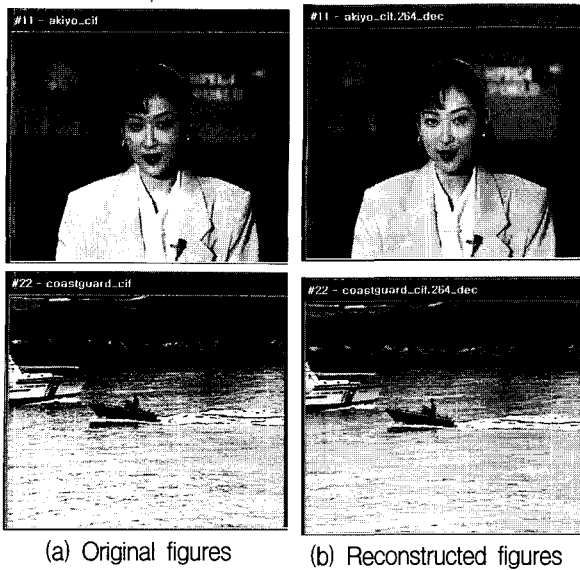


그림 9. Co-Simulation을 통한 검증 화면

Fig. 9. Reconstructed figures by using Co-Simulation.

DAU는 10.4% 차지하였다. 움직임 추정기는 동작 주파수 50 MHz에서 30 CIF Frames/sec 처리가능하다. 그림 9는 원본 이미지와 Co-Simulation 한 결과의 이미지를 보여준다.

## V. 결 론

본 연구에서는 H.264 인코더의 움직임 추정기를 하드웨어로 설계하였다. H.264에서는 기존 동영상 압축 표준에 비해 압축 효율과 영상 화질을 증가시키기 위해 가변블록과 다중 프레임 움직임 추정 알고리즘을 지원하고 있다. 움직임 추정기가 H.264 인코더의 대부분의 연산량을 차지한다. 따라서 본 연구에서 움직임 추정기의 연산량 감소를 위한 움직임 예측 및 Alternative Row Scan (ARS) 방법을 제안하였다. ARS는 DAU를 이용하여 탐색 영역의 연산 시작 위치를 결정하고, 연산 시작 위치의 Row부터 Alternating한 방식으로 데이터를 스캔 하는 방법을 말한다. 또한, 다중 프레임 움직임 추정을 위해 기존의 FDVS 알고리즘을 응용하여 다중 프레임의 복잡도를 낮추었다. 본 연구에서 제안한 Alternative Row Scan (ARS) 방법은 JM8.2에 제시된 가변블록을 지원하는 다중 프레임 움직임 추정에 비해 75% 이상의 연산량을 감소시키고 화질 열화가 없도록 하였다. 제안한 움직임 추정기는 iPROVE Xilinx Virtex2 FPGA Board를 이용하여 하드웨어의 동작을 검증하여 정상 동작함을 확인하였으며, 50MHz에서 30 CIF Frames/sec 처리가 가능하였다.

## 참 고 문 헌

- [1] I. Richardson, "H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia," John Wiley & Sons, 2003.
- [2] <http://www.kumnong.co.kr/>
- [3] Viet L. Do, Kenneth Y. Yun, "A Low-Power VLSI Architecture for Full-Search Block-Matching Motion Estimation," IEEE Trans on Circuits and Systems for Video Technology, Vol.8, No.4, pp393-398, Aug 1998.
- [4] M. Chen, Y. Chiang, H. Li, M. Chi, "Efficient multi-frame motion estimation algorithms for MPEG-4 AVC/JVT/H.264," Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on Vol 3, pp737-740, May 2004.
- [5] J. Zhang, Y. He, S. Yang and Y. Zhong, "Performance and complexity Joint Optimization For H.264 Video Coding," Circuits and Systems, ISCAS '03. Proceedings of the 2003 International Symposium on Vol. 2, pp. 25-28, May 2003.
- [6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in Proc. NTC81, New Orleans, LA, pp. C9.6.1-9.6.5, Nov. 1981.
- [7] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Study of Final Committee Draft of Joint Video Specification (ITU-T REC. H.264 | EC 14496-10 AVC)", Doc, JVT-G050, May. 2003.
- [8] C. Blanch and k. Denolf, "Memory Complexity Analysis of the AVC Codec JM1.7", ISO/IEC JTC1/SC29/WG11 MPEG2002/M8378, Fairfax, May. 2002.
- [9] S. Yeow Yap, J. V. McCanny, "A VLSI Architecture for Variable Block Size Video Motion Estimation," IEEE Trans on Circuits and Systems, Vol.51, No.7, pp 384-389, Jul 2004.
- [10] Y. Huang, C. Chen, C. Tsai, C. Shen, L. Chen, "Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results", Journal of VLSI Signal Processing on Volume 42, pp297-320, March 2006.
- [11] T. Moriyoshi, H. Shinohara, T. Miyazaki, I. Kuroda, "Real-Time Software Video Codec with a Fast Adaptive Motion Vector Search", Journal of VLSI signal Processing on Volume 29, pp239-245, November 2001.

저 자 소 개



윤 미 선(학생회원)  
 2005년 대전대학교 전자공학과  
 학사.  
 2007년 한양대학교 전자전기 제어  
 계측공학과 석사.  
 <주관심분야: 저전력 설계, H.264,  
 3D Graphics, 반도체 설계>



문 동 선(학생회원)  
 2004년 한양대학교 컴퓨터공학부  
 학사.  
 2006년 한양대학교 전자전기 제어  
 계측공학과 석사.  
 2007년 한양대학교 전자전기 제어  
 계측공학과 박사과정

<주관심분야: 저전력 설계, CAD&VLSI>



장 승 호(학생회원)  
 2006년 한양대학교 컴퓨터공학부  
 학사.  
 2007년 한양대학교 전자전기 제어  
 계측공학과 석사과정.  
 <주관심분야: 저전력 설계, H.264,  
 3D Graphics>



신 현 철(평생회원)  
 1978년 서울대학교 전자공학과  
 학사.  
 1980년 한국과학기술원 전기 및  
 전자공학과 석사.  
 1983년~1987년 U.C. Berkeley  
 Ph.D

1983년~1987년 Fulbright scholarship  
 1987년~1989년 MTS, AT&T Bell Lab's,  
 Murray Hill N.J., USA  
 1989년~현재 한양대학교 전자컴퓨터공학부 교수  
 1997년~2007년 현재 IDEC 한양대학교 지역센터  
 센터장  
 <주관심분야: CAD&VLSI, 통신용 반도체 설계,  
 저전력 설계>