메모리 스트림 할당 기법을 이용한 영상처리용 엔트로피 프로세서 설계

이선근* · 정우열**

Design of the Entropy Processor using the Memory Stream Allocation for the Image Processing

Seon-Keun Lee* · Woo-Yeol Jeong**

요 약

IT산업의 가속화로 인하여 다양한 미디어 환경이 조성되고 있는 현대사회에 3D-TV 등의 실시간 영상화면은 매우 중요한 이슈이다. 이러한 고화질의 실시간 영상은 매우 다양한 분야에 적용되고 있으며 CCTV 등의 영상 성능에 중요한 파라미터가 되고 있다. 그러나 이러한 고화질의 영상이라도 보안에 취약한 단점이 있기 때문에 보안채널 또는 각종 보안 알고리즘을 이용하여 이러한 단점을 없애고자 하는 시도가 매우 활발히 진행 중에 있다. 본 연구에서는 이러한 단점을 별도의 보안기술을 부가하여 처리속도를 감소시키는 것이 아니라 영상처리 자체에 보안기능을 부가함으로서 실시간 처리 및 보안성을 증대시키기 위한 방안을 제시한다.

ABSTRACT

Due to acceleration of the IT industry and the environment for a variety of media in modern society, such as real-time video images 3D-TV is a very important issue. These high-quality live video is being applied to various fields such as CCTV footage has become an important performance parameters. However, these high quality images, even vulnerable because of shortcomings secure channel or by using various security algorithms attempt to get rid of these disadvantages are underway very active. These shortcomings, this study added extra security technologies to reduce the processing speed image processing itself, but by adding security features to transmit real-time processing and security measures for improving the present.

키워드

image processing, memory management, stream cryptographic, huffman code, compression
영상처리, 메모리 관리, 스트림 암호, 허프만 코드, 압축

Ⅰ. 서 론

IT 환경의 발달로 3D-TV, VOD 등과 같은 다양한 영상정보매체 기술은 매우 다양한 분야로 발전을 거듭하고 있다. 특히, 초고속 정보 통신망을 비롯한 여러 종류의 채널과 매체를 통한 다양한 형태의 영상정보 서비스의 발달은 매우 비약적이다.

이러한 멀티미디어 환경이 구축되고 수요가 창출됨에 따라 멀티미디어를 구성하는 음성정보와 영상정보를 동시에 처리해야 하는 문제가 발생하게 된다. 이때

* 전 원광대학교 전기전자 및 정보공학부(caiserisk@googlemail.com)
** 한려대학교 멀티미디어정보통신공학과(jeongyeol@hanmail.net)
접수일자 : 2012. 07. 23 심사(수정)일자 : 2012. 09. 05 게재확정일자 : 2012. 10. 05

영상정보의 처리는 음성정보에 비하여 많은 양의 신호처리와 대용량의 저장매체가 필수 불가결하게 된다[1][9]. 이러한 이유로 영상압축에 대한 관심이 고조되고 있는 것이 현실정이다. 또한 다양한 멀티미디어 발전과 더불어 개인정보도 다양하게 노출되고 있는 것이 현실이다. 압축과 개인정보 누출에 대한 정보는 많은 자원을 필요로 하는 분야이기 때문에 이러한 제약조건 등으로 인한 미디어 환경 발전이 다소 늦추어지는 경향이 발생한다[12][13].

일반적으로 영상미디어에서 영상압축[2]의 일반적인 구조는 크게 세 단계로 나누어진다. 첫째는 영상화소 간 상관관계를 제거하는 예측 및 변환 단계, 둘째는 상관관계를 제거한 영상 화소를 한정된 몇 개의 코드 또는 심볼로 바꾸는 양자화 단계, 마지막으로 양자화된 영상화소를 최종적인 비트 형태로 표현하는 엔트로피 부호화 단계로 나누어진다. 이때 엔트로피 부호화기는 전체 시스템의 병목현상이 발생되는 부분으로서 시스템 부하가 가장 심각하기 때문에 엔트로피 부호화기의 성능은 전체 시스템의 성능과 직결된다[3].

그러므로 본 논문은 하드웨어 효용성과 압축효율의 극대화를 실현하고 이와 동시에 개인정보 누출을 미연에 막을 수 있도록 허프만 부호화의 순간적 디코딩과 렘펠-지브 부호화의 데이터열 처리방식, 그리고 스트림 암호 방식 등을 적용한 메모리 스트림 할당기법(MSA : Memory Stream Allocated Method) 을 사용하여 영상처리 장치에 적용할 수 있는 엔트로피 프로세서를 설계하였다.

설계된 엔트로피 프로세서는 MSA를 적용함과 동시에 영상 보상과정을 축소함으로써 전체 엔트로피 부호화기의 처리효율이 증가했으며 부호화를 수행하면서 압축과정 및 보안기능이 동시에 수행되기 때문에 엔트로피 프로세서 구현시의 비용절감이 크게 증대됨을 확인하였다.

II. 영상 부호화 기법

영상정보 압축은 IT 기기의 저장 공간을 절약할 수 있으며, 일정한 정해진 대역폭을 통하여 데이터 전송시간을 단축하여 전송효율을 높일 수 있다. 데이터 저장 공간 감소와 전송 효율 향상 및 정보유출은 시스템의 경제적 비용을 절감할 수 있을 뿐 아니라 시스템의 성능 향상에도 기여한다. 카메라로 찍은 동영상 정보는 그림 1과 같이 세 가지 기능을 이용하여 영상 정보를 압축하게 된다. 공간적 압축은 DCT(Discrete Cosine Transform)를 이용하여 화소값이 저주파 성분으로 집중하도록 변환하고 고주파 성분을 제거함으로써 수행된다[4][5]. 화면 내 압축을 위해서는 블록단위로 DCT를 수행하여 블록의 에너지를 저주파 성분에 집중시킨 후 양자화 과정을 수행한다[6].

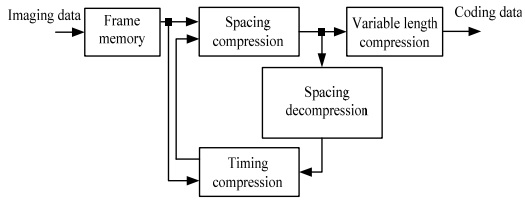

그림 1. 영상정보 압축
Fig. 1 Image information compression

timing compression은 인접 픽쳐 간의 유사성을 이용한 방법으로 과거 또는 미래의 영상으로부터 현재 블록과 가장 유사한 블록을 추출하고 상대적인 차이값을 부호화하여 전송함으로써 시간적 중복성을 제거하는 압축 방법이다. 가변 길이 압축은 공간적, 시간적으로 압축된 데이터의 발생 확률에 따라 서로 다른 길이의 부호를 부여함으로써 발생 부호의 평균 길이를 줄이는 방법이다[2][7].

멀티미디어와 인터넷 콘텐츠의 발달은 실시간 영상처리라는 문제점을 발생시킨다. 이러한 실시간 영상정보의 구현은 고효율 압축률을 가지며 동영상에 대한 코덱 기술이 존재해야 가능한 분야이다. 현재 동영상 표준인 MPEG-x에 대한 압축과 부호화에 대하여 일반적으로 사용되는 알고리즘은 발생확률에 따라 부호화를 수행하는 허프만 부호화 기법이다. 허프만 부호화의 특징은 여타의 부호화 기법들과 비교하여 알고리즘 구현에서 우수하다는 장점이 있으나 손실부호화 기법이라는 것이 단점이다. 그러므로 영상 및 음성정보가 아닌 데이터의 실시간 정보 전송은 커다란 오류를 발생하게 되며 이때 발생하는 오류에 대하여 정정능력이 작다는 것이 단점이다[3].

이러한 손실 부호화 기법과는 다르게 비손실 부호화 기법은 code book을 이용하여 부호화를 수행하는 기법으로써 부호화 과정 중에 부호 북에 저장된 내용과 일치하는 정보가 유입될 경우 별도의 부호화를 수행하지 않고 저장된 부호를 이용하여 부호화를 수행하는 기법이다. 이러한 비손실 부호화 기법은 오류 정정에 매우 효과적이기 때문에 정보에 대한 신뢰성이 높게 요구되는 분야에 주로 사용되어진다[2].

MPEG-2 비디오 인코더의 블록도는 그림 2와 같다. 프레임 메모리에서는 영상 데이터를 프레임별로 저장하고 변환부에서 데이터를 움직임 보상부에서 제공되는 기준 영상 데이터와 비교하여 차이 값을 계산한 후 DCT를 수행하게 되며 그 결과는 양자화부에서 양자화 된다. 양자화 된 데이터는 역양자화부와 역변환부를 거쳐 움직임 보상부로 전달된다. 움직임 추정부는 매크로 블록단위로 프레임간 움직임 벡터를 추정하여 움직임 보상부로 전달하고, 움직임 보상부는 프레임간 차분 부호화를 위해 필요한 기준 프레임 데이터를 제공한다. 엔트로피 부호화기는 가변길이 부호화를 수행하고 이 결과를 버퍼에 저장하여 프로그램 다중화부로 출력한다. 비율 제어부는 출력버퍼의 넘침이나 모자람이 발생하지 않고 영상 부호화 출력 비트율을 일정하게 유지할 수 있도록 양자화 파라미터를 제어한다.

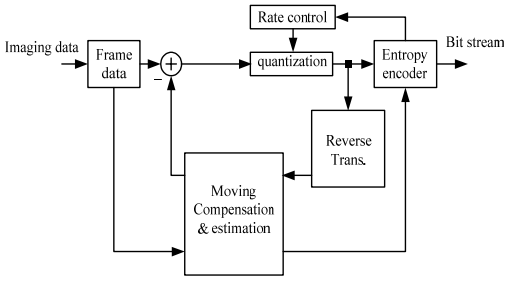

그림 2. 비디오 인코더의 블록도
Fig. 2 Block diagram of video encoder

III. 메모리 스트림 할당(MSA)을 이용한 엔트로피 프로세서

대용량의 영상 데이터 처리 및 전송효율을 하드웨어적인 측면에서 극대화하기 위하여 MSA를 이용한 엔트로피 프로세서를 제안한다. 부호화를 수행하기 위해 본 논문에서 제안한 MSA는 기존 허프만 부호화 방식에 대하여 부호 테이블 및 사전(dictionary)을 이용하여 영상 데이터를 처리하도록 하였다. 따라서 부호화와 압축과정에 소요되는 처리 시간 및 영상 데이터를 처리하는 과정 중에 발생되는 병목현상에 의한 처리시간에 대하여 처리효율이 향상되도록 하였다. 또한 처리속도 향상 및 메모리 포화, 정보누출 방지[8]를 위하여 데이터에 대한 생성과 재생을 반복 사용함으로써 압축을 수행하고 복원하는데 걸리는 시간을 줄일 수 있는 방법에 대해 고찰한다.

이진 분류기(binary classification)는 부가정보의 선택여부에 따라 이진 순방향 분류기와 이진 역방향 분류기의 두 종류가 있다. 이러한 이진 분류기의 특징은 부호화와 복호화의 복잡도가 낮으며 효율적인 엔트로피 부호화를 수행할 수 있다. 그러나 이러한 이진 분류기는 정보의 분류를 두 가지 종류로써 구별하여 부호화를 수행하기 때문에 정보의 상관성이 크고 부호북을 생성하는 정보의 양이 방대해질 수 있다. 이러한 특징은 정보의 원활한 흐름을 방해하는 병목현상을 유발하게 되며 영상정보와 같은 데이터량이 많은 정보를 처리할 경우 더욱 두드러진다. 이런 단점을 제거하기 위하여 제안된 MSA 기법은 단순 이진 분류가 아닌 다치 분류를 선택하였다. 이러한 다치 분류는 메모리 사용을 극대화하게 되며 데이터의 지연을 제거할 수 있기 때문에 병목현상을 없앨 수 있다.

다치 단방향 분류 부호기(multi-value one-way classification coder)는 현재의 심볼을 대상으로 적절하게 설정된 범위(searching window)로 분류하고 분류된 정보를 부가 정보로 사용하는 방법이다.

즉, 입력 정보원을 적당한 기준값과 비교해서 입력 심볼들을 여러가지 범위로 분류하고 매 심볼마다 1비트씩의 분류 부가 정보를 갖게 된다. 이때 기준값과 비교하여 일정범위 보다 큰 값들인 *Sig*(significant symbol)와 기준값보다 작은 값들인 *Nonsig*(non-significant symbol), 그리고 분류 부가 정보인 MAP(binary classification map)으로 분류되는 정보종류와 각각에 대하여 기준값이 달라질 경우에 해당하는 정보들로 나눌 수 있다.

입력 정보원 X 집합에 대한 정보량을 $I(X)$로 정의하고, 큰 값으로 분류된 정보원 Sig 집합의 정보량을

$I(Sig)$로, 작은 값으로 분류된 정보원 $Nonsig$ 집합의 정보량을 $I(Nonsig)$로, 그리고 이진 분류 부가 정보원 Map 집합의 정보량을 $I(Map)$으로 정의하고, 심볼 수를 M, 입력 시퀀스의 수를 N, 문턱값을 K, 심볼의 확률을 p_i라 할 때 매 입력 심볼마다 분류 정보를 부가하므로 Map 집합의 크기는 N이 되고 Map 집합의 심볼인 '0'과 '1'의 각각의 확률 값은 식 (1)과 식 (2)로 구해진다. 또한 전체크기에 대하여 Sig 집합의 크기는 $N\times PM_1$이 되고 $Nonsig$ 집합의 크기는 $N\times PM_0$가 된다.

$$P_{map}\{0\}=\sum_{i=0}^{k-1}p_i=PM_0 \tag{1}$$

$$P_{map}\{1\}=\sum_{i=0}^{k-1}p_i=PM_1 \tag{2}$$

Map 집합의 정보량은 식 (3)과 같이 Map 집합 크기와 Map 집합 엔트로피와의 곱으로 표현 할 수 있다.

$$\begin{aligned} I(Map) &= Size(Map)\times H(Map) \\ &= N\times(-PM_0\log_2 PM_0 - PM_1\log_2 PM_1) \end{aligned} \tag{3}$$

유사한 방법으로 Sig 집합의 정보량과 Nonsig 집합의 정보량을 계산하면 각각 식 (4)와 식 (5)와 같다. 여기에서 H(Map)은 MAP에 대한 엔트로피이다.

$$\begin{aligned} I(Sig) &= \mathrm{Size}(Sig)\times H(Sig) \\ &= N\times PM_1\times\left(-\sum_{i=K}^{M}\frac{p_i}{PM_1}log_2\frac{p_i}{PM_1}\right) \end{aligned} \tag{4}$$

$$\begin{aligned} I(Nonsig) &= \mathrm{Size}(Nonsig)\times H(Nonsig) \\ &= N\times PM_0\times\left(-\sum_{i=0}^{K-1}\frac{p_i}{PM_0}log_2\frac{p_i}{PM_0}\right) \end{aligned} \tag{5}$$

입력 정보원은 세 가지 종류의 정보원으로 분류되고 각각의 정보원들에 대한 정보량은 식 (3)과 식 (4), 식 (5)와 같이 된다. 그러므로 전체 입력정보원은 각각의 정보원들에 대한 대수적인 합으로 표현이 가능하다. 즉, 식 (6)과 같이 표현할 수 있다.

$$\begin{aligned} &N\times(-PM_0\log_2 PM_0 - PM_1\log_2 PM_1)+N\times PM_1 \\ &\quad\times\left(-\sum_{i=K}^{M}\frac{p_i}{PM_1}log_2\frac{p_i}{PM_1}\right) \\ &\quad+N\times PM_0\times\left(-\sum_{i=0}^{K-1}\frac{p_i}{PM_0}log_2\frac{p_i}{PM_0}\right) \\ &=-\left(N\times\sum_{i=K}^{M}p_i\log_2 p_i+N\times\sum_{i=0}^{K-1}p_i\log_2 p_i\right) \\ &=-N\times\sum_{i=0}^{M}p_i\log_2 p_i \end{aligned} \tag{6}$$

식 (6)은 입력 신호원에 대한 전체 정보량이 되므로 분류된 세 종류의 정보원에 대한 전체 정보량은 식 (7)과 같다.

$$I(X)=I(Sig)+I(Nonsig)+I(Map) \tag{7}$$

식 (7)과 같이 다치 단방향 분류는 입력 신호원의 정보량에 대한 변화없이 신호원을 단순히 분류하고 매 심볼마다 분류를 위한 부가 정보를 추가하게 되지만 분류 전 정보량과 분류 후의 전체 정보량에는 변화가 없다. 즉, 정보량의 변화가 없기 때문에 기준값을 임의의 갯수로써 다치 분류를 설정하여 정보량을 분산하여도 전체에 대한 정보량 손실은 없다는 것을 의미한다.

허프만 부호기법은 가변길이 부호어를 저장하고 있는 허프만 테이블을 사용하여 입력 심볼을 부호화한다. 신호원의 크기가 작은 경우에는 발생 가능한 모든 심볼들에 대해 서로 다른 가변길이 부호어를 사용한다. 그러나 신호원의 크기가 매우 큰 경우에는 많은 양의 메모리를 요구하므로 발생빈도가 높은 일부 심볼에 대해서만 서로 다른 가변길이 부호어를 사용한다. 일반적으로 허프만 테이블의 크기가 커짐에 따라 평균 비트율은 감소하지만 요구되는 메모리의 양은 증가하게 된다. 제한된 메모리 사용으로 인해 사용할 수 있는 허프만 테이블의 개수가 신호원의 개수보다 적은 경우에는 몇몇 신호원들에 대해서는 허프만 테이블의 공유가 필요하다. 이때 허프만 공유 테이블을 사용하는 과정 중에서 지시함수의 오판으로 인한 메모리의 미사용 영역이 존재하게 되어 한정된 메모리에 대한 비효율적인 현상이 더욱 심화될 수 있다.

이상과 같은 허프만 알고리즘의 단점을 보완하기 위하여 본 논문에서 발생확률에 따른 MSA 기법을 이용한 엔트로피 부호화방법을 제안한다.

허프만 알고리즘에서 모든 $m=1,2,\cdots,M$에 대하여 $\mu(m)\neq n$를 만족하는 정수 $n\in\{1,\cdots,N\}$를 발생시킬 수 있다. 이는 허프만 테이블 H_n가 어떤 신호원에 대해서도 적절하게 설계되지 못했음을 의미한다. 이러한 경우에 인덱스 데이터가 영(=0)을 의미하므로 확률 분포 $F_n'=\{P_n'(k)|k=1,...,K\}$ 가 적절히 정의되지 않는다. 따라서 허프만 테이블 H_n내의 부호어들의 길이는 계산될 수 없게 되고 테이블 H_n는 다음 단계의 과정을 수행할 수 없게 된다. 이와 같이 기존의 제한된 반복강하 알고리즘에서는 미사용 허프만 테이블이 일단 발생하면 이 테이블은 계속해서 사용될 수 없게 되어 결과적으로 주어진 메모리를 충분히 활용하지 못하는 결과를 낳게 된다. 이러한 미사용 허프만 테이블을 재사용하기 위한 방법으로 M개의 신호원들 중 하나의 신호원 S_r를 적절히 선택하여 $\mu(r)=u$로 놓는다. 이와 같은 처리는 반복계산 단계에서 H_u가 S_r에 최적화되도록 하여 다음 단계에서 H_u가 최소한 S_r의 부호화에는 사용될 수 있도록 한다. 신호원 S_r의 선택에 있어서 평균 비트율의 감소량을 미사용 테이블 처리 이득 $g(m)$이라 하면 식 (8)과 같이 나타낼 수 있다.

$$g(m)=0, \qquad I_{\mu(m)}=\{m\}\text{일때}$$

$$g(m)=P'_{\mu(m)}\tilde{b}_1(F'_{\mu(m)})-(P'_{\mu(m)}-P'_m)\times\tilde{b}_1(G'_{\mu(m),m})-P_m\tilde{b}_1(F_m), \quad I_{\mu(m)}\neq\{m\}\text{일때} \tag{8}$$

여기에서 $P'_n=\sum_{r\in 0}P_i$ 이고, $G_{\mu(m),m}$는 $S'_{\mu(m)}$에서 S_m을 제외함으로써 얻어지는 합성 신호원의 확률 분포이다.

만일 $G_{\mu(m),m}=\{q(1),\cdots,q(K)\}$로 표시한다면 식 (9)와 같이 주어진다.

$$q(k)=\frac{P'_{\mu(m)}\cdot p'_{\mu(m)}(k)-P_m\cdot p_m(k)}{P'_{\mu(m)}-P_m}, \quad k=1,2,\cdots,K \tag{9}$$

식 (8)에서 $\tilde{b}_l(F)$는 확률분포가 F인 신호원을 그 확률 분포에 적합하도록 설계된 크기 l인 허프만 테이블을 사용하여 부호화한 경우의 평균 비트율을 의미한다. 식 (9)에서 확률분포가 어떠한 신호원에 대해서도 일정한 값을 가지도록 할 경우 허프만 테이블의 미사용 부분은 사라지게 될 것이다. 또한 정보원에 대한 부호화를 수행하기 위하여 사용되어지는 허프만 테이블의 확률분포가 일정하게 되면 압축을 위해 사용되는 사전 또한 공유하여 사용할 수 있게 되며, 이는 허프만 테이블과 사전 사이에서 상호간에 미사용 테이블이 존재하는지, 아니면 사전가 포화되는지를 자동적으로 확인하고 감시할 수 있게 된다. 이는 가변길이 부호화 방식을 사용하는 부호기의 메모리에 대한 효율을 극대화할 수 있다는 것을 의미하게 되며 이러한 알고리즘을 구현할 경우 소비되는 비용 역시 크게 감소하게 된다.

MSA 기법은 이와 같은 사전 사용시 searching window의 stack point 결정을 스트림 암호알고리즘을 사용하여 수행한다는 것이다.

일반적으로 스트림 암호알고리즘은 채널오류가 없는 이상적인 채널에서는 문제가 없지만 그렇지 않은 채널에서는 블록 대체로 인한 비트오류확산이 발생된다. 스트림 암호알고리즘 적용시 송신단에서 균일분포를 갖는 암호문 입력으로부터 송신블록대체가 일어날 확률은 2^{-n}이며 그 반대확률은 $1-2^{-n}$이다. 마찬가지로 수신단에서 균일분포를 갖는 복호문 입력으로부터 수신블록대체가 일어날 확률은 2^{-n}이며 그 반대확률은 $1-2^{-n}$이다[8][10]. 이때 임의의 n에 대하여 P_M은 송신대체 시 수신단에서 미검출되는 미검출 확률, P_F는 송신 비대체시에도 채널오류로 인하여 수신단에서 대체블록으로 검출되는 오검출 확률, 채널의 비트 오류율(BER)을 B라 둘 때 스트림 암호 적용시 전체 비트오류율 P_E는 다음과 같이 계산된다.

P_M=(송신단에서 블록 대체될 확률)×(수신단에서

대체블록을 미 검출할 확률)×(미 검출에 따른 블록 내 평균 오류확산비트 수)

$$=(2^{-n})[1-(1-B)^n](n/2)=(n)2^{-(n+1)}[1-(1-B)^n] \quad (10)$$

P_F=(비 대체블록에 채널오류가 발생될 확률)×(수신단에서 대체블록으로 판단할 오검출 확률)×(오검출로 인한 블록 내 평균 오류확산비트 수)

$$=[1-(1-B)^n](2^{-n})(n/2)=(n)2^{-(n+1)}[1-(1-B)^n] \quad (11)$$

$$P_E=P_M+P_F+B=2P_M+B \quad (12)$$
$$=(n)2^{-n}[1-(1-B)^n]+B$$

표 1. 스트림 암호알고리듬(sc)의 전체 비트 오류율(BER 또는 n에 따른)

Table 1. Total bit error rate of stream cryptographic algorithm

BER	total error rate of sc ($k=16,\ n=8$)	n	total error rate of sc ($BER=10^{-5}$)
10^{-1}	1.1779790×10^{-1}	6	1.5632498×10^{-5}
10^{-2}	1.2414228×10^{-2}	7	1.3833208×10^{-5}
10^{-3}	1.2491268×10^{-3}	8	1.2499912×10^{-5}
10^{-4}	1.2499125×10^{-4}	9	1.1584116×10^{-5}
10^{-5}	1.2499912×10^{-5}	10	1.0977844×10^{-5}
10^{-6}	1.2499991×10^{-6}	11	1.0591593×10^{-5}
10^{-7}	1.2499999×10^{-7}	15	1.0068753×10^{-5}
10^{-8}	1.2500000×10^{-8}	20	1.0003819×10^{-5}
10^{-9}	1.2500000×10^{-9}	25	1.0000186×10^{-5}
10^{-10}	1.2500000×10^{-10}	31	1.0000004×10^{-5}

이때 암호시스템을 설계할 경우 전체 비트 오류율은 표 1의 왼쪽 결과와 같아지며 $n=8$일 때 전체 비트오류율은 BER에 비하여 평균 1.25배정도 증가하였음을 알 수 있다. 그러나 이러한 증가는 표 1의 오른쪽 결과에서 알 수 있는 바와 같이 n이 커질 경우 전체 비트오류율은 1에 근사하게 되므로 스트림 암호 적용에 따른 오류확산은 무시될 수 있다.

그림 3은 허프만 테이블과 사전, 스트림 암호 코드와의 메모리 할당 맵을 보여주고 있다. 외부로부터의 정보원이 인가되면 사전 영역의 정보원과 정합여부를 판별한 후 허프만 테이블 영역, 사전 영역, 업데이트 영역 중의 어느 한 곳으로 진로가 결정되

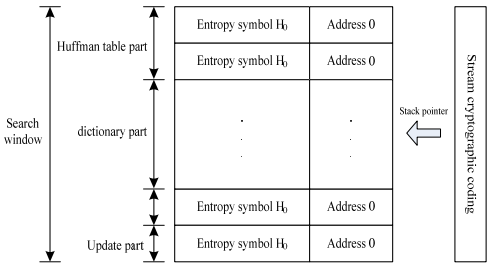

그림 3. 허프만 테이블과 사전 공유 메모리

Fig. 3 Huffman table and dictionary shared memory

며 정보원이 인가될 때마다 스택 포인터는 메모리의 정보원이 인가될 때마다 스택 포인터는 메모리의 전체 영역을 전역 탐색하면서 메모리에 대한 포화 및 미사용 영역을 제어하게 된다. 또한 스택 포인터는 식 (8)과 같이 메모리에 대한 효율을 증가시키기 위하여 항상 $g(m)$이 '0'이 되도록 address를 할당하게 된다. 사용할 수 있는 허프만 테이블의 개수가 신호원의 개수보다 매우 작은 경우($N\ll M$) 즉, 영상 데이터와 같이 데이터량이 한정된 메모리에 비하여 매우 큰 경우의 신호들에 대해서는 허프만 테이블의 수를 증가시킨다 하더라도 비효율적이며 적절하지 않을 수 있다. 그래서 가변길이보다는 고정길이 부호화되는 것이 유리하다. 성능향상 정도는 N값이 작을수록 커진다. 그러므로 고정길이 부호화를 위한 MSA 기법의 기능은 다음과 같이 표현이 가능하다.

$$\mu(m)=\ \underset{1\le n\le N}{\arg\min}\ b_m^n\cdot I_{IC},\quad \text{여기서 } m=1,\ 2,\ \cdots,\ M \quad (13)$$

여기에서 I_{IC}는 MSA 기법에서 인덱스 집합을 의미한다. 즉, 인덱스의 선택에 따라서 지시함수가 탐색해야할 메모리 공간의 영역이 변화하도록 한다. 그림 3에서 스택 포인터는 지시함수 $\mu(m)$을 식 (13)과 같이 변형하였을 경우 허프만 영역, 사전 영역, 업데이트 영역의 진행 영역을 결정짓는 지시함수로 변한다. 즉, MSA 기법에서 스택 포인터는 지시함수의 기능을 포함하게 된다. 만일 $b_m^{\mu(m)}\ge\log_2 K$이면 $\mu(m)=0$으로 놓는다.

그림 4는 인덱스 집합으로 인한 지시함수의 변화를 도시하고 있다. 인덱스 판별부가 첨가되어 엔트로피

프로세서는 더욱 많은 정보에 대하여 선별할 수 있는 기능을 가지게 되며 이러한 지시함수의 집합인 스택 포인터 & sc는 매우 효율적인 메모리 관리를 수행할 수 있게 된다.

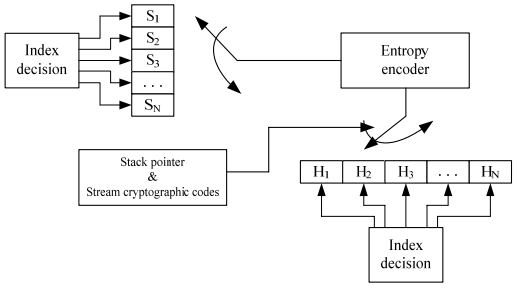

그림 4. 다차원 정보원에 대한 엔트로피 부호기
Fig. 4 Entropy coder for multidimentional sources

본 논문에서는 영문과 국문 텍스트에 대해서 발생빈도 및 압축률을 조사하고, MSA 기법을 적용하였을 때 압축률을 조사하였다. 바바라, 보트 영상과 텍스트에 대한 허프만, LZW, 그리고 MSA 기법의 상호 성능평가를 위하여 Matlab을 이용한 모델링을 수행하였으며 이에 대한 결과는 그림 5로 표현하였다.

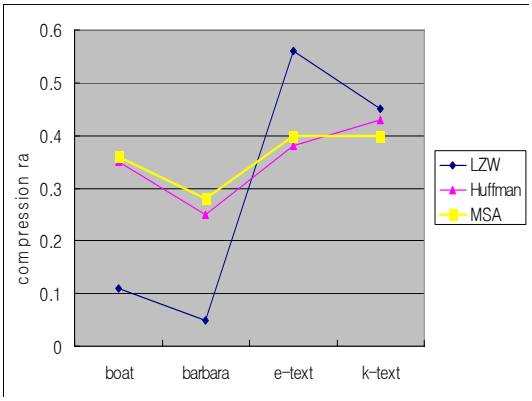

그림 5. MSA기법을 사용했을 때 압축 성능 비교
Fig. 5 Comparison of compression performance when using MSA method

표 2는 MSA 기법과 기존 압축 부호화에 대한 압축 성능을 보여주고 있다.

표 2. MSA 기법의 압축 부호화률[Kbytes]
Table 2. Compression encoding rate of MSA method

구분	원문	LZW	Huffman	MSA기법
부두	409	266	368	260
바바라	409	311	392	291
영문텍스트	665	290	411	400
국문텍스트	728	406	410	375

영상인 경우, 손실 압축방식인 허프만 부호화 방식이 비손실 압축방식인 LZW에 비하여 압축률이 높으며 MSA 기법 부호화 방식이 허프만 부호화 방식에 비하여 더욱 높은 압축률을 보임을 확인하였다[2][11].

이상과 같이 MSA기법의 부호화 및 압축 부분은 다음과 같이 구성된다. 첫째로 다치 단방향 분류 기능이다. 기준값을 여러 개 설정하여 인덱스 집합을 크게 함으로써 정보량은 변하지 않으면서 정보원을 분류한다. 둘째로는 정보원의 발생확률에 따른 엔트로피 부호화를 수행한다. 즉, 허프만 테이블과 사전을 동시에 사용함으로써 메모리 내부에서 사용되지 않는 메모리를 없애고 가변길이 부호화를 수행하는 과정에서 발생하게 되는 메모리의 포화상태를 방지하도록 하였다. 허프만 테이블과 사전의 제어는 지시함수를 인덱스 집합을 이용하여 스택 포인트로 변환하여 사용한다.

Ⅳ. MSA기법 엔트로피 프로세서 설계

본 논문에서는 영상 데이터와 같이 용량이 매우 큰 정보에 대한 처리효율 및 정보누출 방지 기능을 증대시키고자 MSA기법을 이용하여 영상처리용 엔트로피 프로세서를 설계하였다.

MSA기법의 주요기능인 가변길이 부호화 기능과 압축 기능을 하나의 모듈로써 설계하였으며 허프만 테이블과 사전 공유 메모리를 첨가하여 구현하였다. 또한 설계된 MSA기법에 의한 엔트로피 프로세서는 허프만 테이블과 사전의 영역을 구별하여 사용하도록 하였으며 영역의 구별 및 제어는 스트림 암호 기능을

포함하는 스택 포인터를 사용하였다. 또한 입력 정보원과 내부에 존재하게 되는 기준값들에 대하여 정합 기능을 주어 스택 포인터로써 부호화와 압축 기능을 동시에 수행하면서 메모리의 포화 및 미사용에 대한 비효율적인 요소를 배제하였다.

MSA기법에 의한 엔트로피 프로세서는 입력정보 지연부와 헤더정보 부호화부, 부호화부로 구성되어 있으며 부호화부는 인덱스 제어부와 하부기능 부호화부, 버퍼제어부로 구성되어 있다.

4.1. 인덱스 제어부

그림 6은 인덱스 제어부로써 헤더정보의 부호화 과정 중에 발생되어지는 메모리의 비효율적인 사용으로 인한 메모리부의 메모리 효율 증가 및 병목현상을 제거하는 기능을 수행하는 부분이다.

8비트 데이터의 입력을 받고 24비트의 어드레스를 갖는 RAM을 이용하여 입력정보원이 사전에 포함된 내용인지 아닌지의 여부에 따라서 압축과 부호화를 동시에 수행할 것인지, 부호화만 수행할지를 결정하게 된다. 2^{24}개의 어드레스는 캡쳐된 데이터에 대한 전역탐색을 수행하게 되는 스택 포인터로써 기능을 수행하게 된다. 입력값이 저장된 값이 아닐 경우에는 메모리에 새롭게 업데이트 되는 동시에 저장된 값과의 차이를 메모리에 별도로 저장하여 다음에 유입되어지는 데이터와의 정합여부를 판별하기 위하여 다음 데이터를 기다리게 된다.

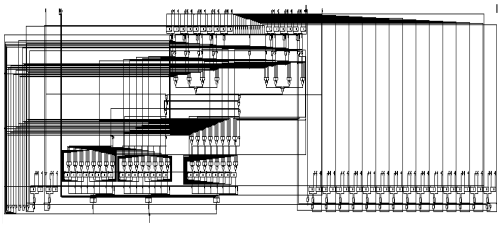

그림 6. 인덱스 제어부의 회로도
Fig. 6 Circuit of index control unit

만약 저장된 데이터와 입력 데이터가 일치할 경우 저장되어진 데이터에 해당하는 정보원을 부호화하기 위하여 메모리로부터 출력된다. 이러한 기능을 수행하게 되면 메모리의 전체 어드레스를 전역 탐색하게 되고 이것은 대용량의 정보값에 대하여 효율적인 메모리 관리가 가능해진다. 즉, 미사용 메모리 및 메모리 포화 상태를 억제시킴으로써 버퍼제어부에서의 병목현상을 제거할 수 있기 때문에 처리속도가 향상된다.

4.2. MSA기법에 의한 엔트로피 부호화부

MSA기법에 의한 엔트로피 부호화부의 전체 회로는 그림 7과 같다. VLC로 입력되는 외부 입력 신호들 중 pesh_val과 pesh는 pesh_clk의 상승 에지 동기를 맞추어 입력되며 caption과 caption_vld는 caption_clk의 상승 에지에 맞추어 입력된다. 나머지 입력들은 시스템 클럭인 clk의 하강 에지에 동기를 맞추게 설계하였다. 내부 신호들은 head_encoder와 indelay에서 출력되는 신호들은 상승 에지일 때 출력되어 이를 입력으로 받은 부호화부(submerge) 역시 packer와 pac_buf 만을 제외하고 모든 입출력이 상승 에지에서 동작하게 된다.

회로합성에는 SYNOPSYS Ver. 1999.10 Design Analyzer를 이용하였으며 모의실험은 SYNOPSYS Ver. 1999.10 vhdldbx를 이용하여 수행하였다. 모의실험 결과 기존의 엔트로피 프로세서의 전체 게이트 수는 91,768개이며 MSA기법을 이용한 엔트로피 프로세서의 게이트 수는 117,463개로써 기존 시스템에 비하여 크기면에서 28% 증가함을 확인하였다.

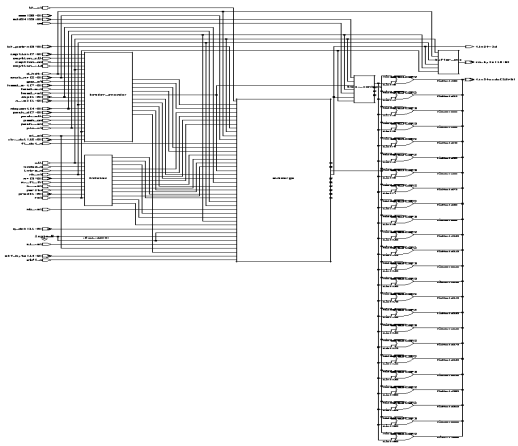

그림 7. MSA기법 엔트로피 프로세서 회로도
Fig. 7 Entropy processor circuit using MSA method

Ⅴ. 결 론

IT 환경에서 영상, 음성 그리고 데이터 통신에 대한 디지털 전송기술을 위해서는 정보의 부호화, 압축과정 그리고 정보누출 방지가 매우 효율적이어야 한다.

따라서 본 논문은 영상 시스템에서 소프트웨어로 구현했을 경우보다 하드웨어로 구현했을 경우 더욱 높은 데이터 처리율과 압축률 및 정보누출 방지 기능을 가질 수 있도록 MSA 기법을 제안하였고, 이를 이용하여 영상처리용 엔트로피 프로세서를 설계하였다. 이것은 가변 길이부호에 적용된 허프만 부호기법에 스트링 비손실 압축 알고리즘을 적용하여 부호화와 압축을 병행하여 수행하도록 하는 것으로써 부호화된 데이터는 중간 인터페이스 과정을 통하여 부호화를 최적화시킬 수 있다. 이렇게 구현된 MSA 기법 엔트로피 프로세서의 게이트 수는 기존의 엔트로피 부호화기보다 크기면에서 28% 증가하는 단점이 있으나, 허프만 부호나 LZW 압축방식에 비하여 데이터량이 방대한 영상정보의 부호화 및 압축에 더욱 효율적이라는 것을 확인하였다.

압축률 및 부호화 효율 비교에 있어서 텍스트인 경우, MSA기법은 허프만 부호화에 비하여 높고 LZW에 비하여 다소 낮음을 확인하였으며 영상인 경우에는 허프만 부호화와 LZW 부호화에 비하여 전체적으로 높은 효율을 나타냄을 확인할 수 있었다.

참고 문헌

[1] A. F. Inglis, “Video Engineering”, McGraw-Hill, New York, 1993.

[2] D. J. Legall, “MPEG : A Video Compression Standard for Multimedia Applications”, Commun. of the ACM“, Vol. 34, No. 4, pp. 47-58, April 1991.

[3] A. Puri, R. Aravind, and B. G. Haskell, “Adaptive Frame/Field Motion Compensated Video Coding”, Signal Processing : Image commun., Vol. 5, pp. 39-58, February 1993.

[4] D. Anastassiou, “Scalability for HDTV”, International Workshop on HDTV‘92, Signal Processing of HDTV, IV, pp. 9-15, 1993.

[5] E. Petajan, “The HDTV Grand Alliance System”, Proceedings of the IEEE, Vol. 83, No. 7, pp. 1094-1105, July 1995.

[6] K. Herrmann, “Architecture and VLSI Implementation of a RISC Core for a Monolithic Video Signal Processor”, in VLSI Signal Processing, VII, pp. 368-377, IEEE, New York, 1994.

[7] Chin-Chen Chang, Min-Shian Hwang, Tung-Shou Chen, “A new encryption algorithm for image cryptosystems”, The Journal of Systems and Software, Vol. 58, pp. 83-91, 2001.

[8] S. Li, X. Zheng, X. Mou, Y. Cai, "Chaotic encryption scheme for real time digital video", Proceedings of the SPIE on electronic imaging, San Jose, CA, USA, 2002.

[9] G. Chen, Y. Mao and C.K. Chui, "A symmetric image encryption based on 3D chaotic maps", Chaos Solitons Fractals, Vol. 21, pp. 749‐761, 2004.

[10] Shujun Li, Guanrong Chen and Xuan Zheng, "Chaos-based encryption for digital images and videos", chapter 4 in Multimedia Security Handbook, February 2004.

[11] Yaobin Mao, Guanrong Chen, and Shiguo Lian, "A symmetric image encryption scheme based on 3D chaotic Cat maps", Chaos, Solitons and Fractals, Vol. 21, pp. 749-761, 2004.

[12] 나성훈, 신현식, “VoIP 보안 관련 주요기술에 대한 분석”, 한국전자통신학회논문지, 5권, 4호, pp. 385-390, 2010.

[13] 김용연, “프랙털 영상 부호화에 관한 연구”, 한국전자통신학회논문지, 7권, 3호, pp. 559-566, 2012.

저자 소개

이선근(Seon-Keun Lee)

1995년 원광대학교 전자공학과(공학사)

1997년 원광대학교 대학원 전자공학과 (공학석사)

2003년 원광대학교 대학원 전자공학과 (공학박사)

2006년~2008년 원광대학교 전기전자 및 정보공학부 전임강사

※ 관심분야 : 이동통신시스템, 암호시스템, VLSI 설계

정우열(Woo-Yeol Jeong)

1982년 원광대학교 전자공학과(공학사)

1984년 경희대학교 대학원 전자공학과(공학석사)

1999년 원광대학교 대학원 전자공학과(공학박사)

1995년~현재 한려대학교 멀티미디어정보통신공학과 교수

※ 관심분야 : 이동통신시스템, 암호시스템, VLSI 설계