

논문 2010-47SP-5-23

Polyphase 인코더의 병렬 처리에 대한 연구

(A Study for The Parallel Processing in The Polyphase Encoder)

조 동 식*, 나 성 응**

(Dong Sik Cho and Sung Woong Ra)

요 약

본 연구에서는 대용량의 영상 데이터를 압축할 수 있는 polyphase 인코더를 제안한다. polyphase 인코더는 병렬로 작동하는 다수의 내부 인코더로 구성되어 있으며 입력되는 영상 데이터를 분배기에 의해 내부 인코더로 분배한 후 각각의 내부 인코더가 분배된 비디오 신호를 병렬로 압축하는 구조로 되어있다. 이 구조는 내부 인코더의 샘플링 속도를 감소시켜서 대용량의 영상 데이터를 처리할 수 있지만 PSNR이 감소하고 참조 프레임 전송해야하는 단점을 갖는다. 본 연구에서는 polyphase 인코더의 PSNR이 감소하는 원인을 분석하였고 그 해결 방법과 참조 프레임 전송을 감소시키는 방법을 제시하였으며 시뮬레이션을 통해 polyphase 인코더가 기존의 단일 인코더와 동일한 PSNR을 발생시킴을 보였다.

Abstract

In this paper, we proposed a polyphase encoder that consists of multiple internal encoders. The multiple internal encoders were configured in parallel. Successive frames of image were distributed to separate encoders by way of a image divider and processed in parallel. In this way, the sampling rate of the encoder was reduced by the factor of number of encoders in parallel. In our design, however, the PSNR is exactly the same as that to be achieved with the conventional single-phase encoder, which should require a much higher sampling rate.

Keywords : Polyphase, Video, Parallel Processing, PSNR

I. 서 론

영상 압축 방식은 국제 표준화 기구(ISO)에서 개발된 MPEG-1, 2, 4 표준과 국제 전기 통신 연합(ITU)에서 개발된 H.26x 표준으로 발전되어 왔다.^[1~2] 이들 영상 압축 방식은 움직임 추정 (Motion Estimation), 예측 (Prediction), 이산 코사인 변환 (Discrete Cosine Transform), 양자화 (Quantization)등을 공통적으로 사용하며 이에 대한 여러 가지 고속 연산 알고리즘이 연구되어 왔다. 움직임 추정의 경우 그 연산에 가장 많은 시간이 소요되므로 다양한 고속 처리 알고리즘이 제안되어 왔다.^[3~4] 또한 압축하고자 하는 원본 영상 데이터의 크기가 증가하고 하드웨어 기술이 발전하면서 이에

대한 병렬 처리 기법이 연구되어 왔다. 병렬 처리 기법으로는 단일 인코더 내에서 1개 프레임을 여러 조각으로 나누어 병렬 처리하는 기법이 있다.^[5~6] 이 기법은 단일 인코더를 사용하기 때문에 영상 데이터의 용량이 증가하면 이를 처리하는데 제한 사항을 갖는다. 이러한 제한 사항을 극복하기 위하여 1개 프레임을 여러 개의 인코더가 분할하여 압축하는 기법이 있다.^[7~11] 이 기법은 여러 개의 인코더가 출력하는 비트 스트림을 동기화하여 저장 또는 전송하여야 한다.^[12] 그리고 이 기법에서 움직임 추정을 사용하는 경우 현재 슬라이스 내 매크로 블록의 움직임 벡터 연산을 위하여 참조로 사용되는 슬라이스가 상하 방향으로 1개 씩 더 필요하므로 각 인코더로 전송해야 할 데이터 량이 증가하게 된다.^[13] 또한 이 기법은 분할된 경계 부근에서 양자화 스케일 값의 불연속이 발생하는 경우 디코더로 출력되는 화면에 경계면이 보일 가능성이 있다.^[14] 그리고 이전 프레임의 매크로 블록에 대한 압축이 종료되면 현재 프레임

* 정회원, 국방과학연구소

(Agency for Defense Development)

** 정회원, 충남대학교 전기정보통신공학부

(Chungnam National University)

접수일자: 2010년1월20일, 수정완료일: 2010년5월25일

을 매크로 블록 단위로 압축을 시작하는 기법이 있다.^[15] 이 기법은 프레임의 종류가 I 또는 P 프레임인 경우에 가능하고 압축률을 높이기 위하여 B 프레임을 사용하기 곤란하다. 그 밖의 기법으로는 GOP (Group of Pictures) 단위로 여러 개의 인코더가 압축하는 기법이 있으나 지연 문제가 발생한다.^[16] 본 연구에서는 N 개의 내부 인코더가 순차적으로 영상 압축을 수행하는 Polyphase 인코더 방식을 제안한다.^[17] 이 방식은 1개 프레임을 여러 조각으로 분할하여 처리하는 방식이 아니라 입력되는 원본 영상 프레임을 Polyphase 인코더의 내부 인코더로 분배하여 압축한다. 이와 같이 프레임을 순차적으로 분배하면 비트 스트림의 동기화가 필요 없으며 슬라이스 단위로 압축하는 경우보다 인코더로 전송되는 데이터량을 감소시킬 수 있으며 디코더로 출력되는 화면에 경계면이 보이지 않게 된다. 또한 압축률을 높이기 위하여 B 프레임 사용이 가능하다. 그리고 입력되는 프레임의 샘플링 속도(sampling rate)가 감소되어 내부 인코더가 압축하기 위하여 필요한 시간이 증가되는 효과를 얻을 수 있다. 이 방식은 단일 인코더를 사용할 경우 압축하기 곤란한 고속 카메라에서 출력되는 영상 데이터나 대용량의 영상 데이터를 압축하는데 사용될 수 있다. 또한 이 방식은 변환 부호화(Transcoding)의 처리 속도를 향상시키기 위해서 사용될 수 있다.

그러나 Polyphase 인코더 방식은 단일 인코더에 비하여 프레임 사이의 유사성이 감소하기 때문에 PSNR (Peak Signal-to-Noise Ratio)이 감소되는 문제점과 내부 인코더로 현재 프레임과 참조 프레임을 전송해야 하는 문제점을 가지고 있다. 본 연구에서는 Polyphase 인코더의 문제점인 PSNR이 감소되는 원인과 이를 해결하기 위한 방법을 제시하였고 내부 인코더로 전송되는 프레임 전송량을 감소시키는 방법을 제시하는 동시에 이를 구현하기 위한 방법을 제시하였다.

II. Polyphase 인코더

본 연구에서 제안한 Polyphase 인코더의 구조는 <그림 1>과 같다. 분배기는 영상 소스에서 출력되는 연속적인 영상 프레임을 순차적으로 내부 인코더 Encoder 1, 2, ..., N 에 분배한다. 내부 인코더 Encoder 1, 2, ..., N 은 분배된 프레임을 매크로 블록 단위로 압축한다. 내부 인코더에서 발생한 비트는 파일에 저장되거나 유/무선으로 전송된다.

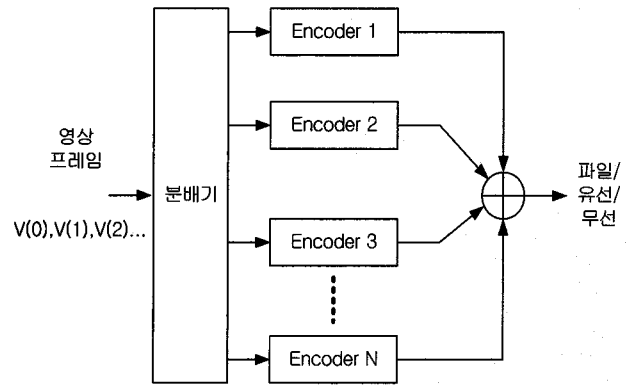


그림 1. Polyphase 인코더 구성
Fig. 1. Configuration of the Polyphase encoder.

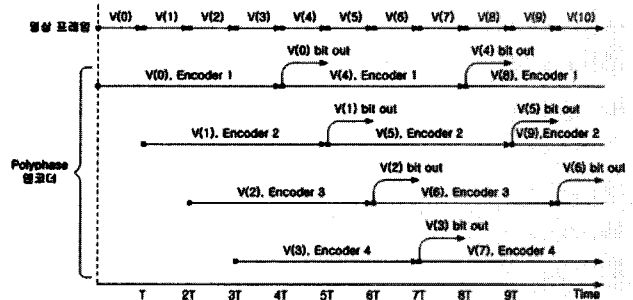


그림 2. Polyphase 인코더의 동작 순서
Fig. 2. The working sequence of polyphase encoder.

<그림 1>의 구성에 대하여 N = 4일 때 내부 인코더의 동작 순서는 <그림 2>와 같다. 영상 소스로부터 영상 데이터 V(0), V(1), V(2), ... 가 생성되는 시간 간격을 T라고 가정한다. t = 0에서 영상 프레임 V(0)이 Polyphase 인코더로 입력되면 분배기에 의해 이 프레임은 내부 인코더 Encoder 1로 입력되고 압축 과정을 거쳐서 t = 4T에 압축된 비트를 출력한다. t = T에서 Encoder 2는 프레임 V(1)을 압축하고 t = 5T에 압축된 비트를 출력한다.

이와 같은 방식으로 Encoder 3, 4는 입력된 프레임 V(2), V(3)을 압축하고 비트를 출력한다. t = 4T에서 Encoder 1은 입력된 프레임 V(4)를 압축하면서 동시에 V(0)의 압축된 비트를 출력한다. t = 5T에서 Encoder 2는 입력된 프레임 V(5)를 압축하면서 동시에 V(1)의 압축된 비트를 출력한다. 이와 같은 방식으로 Encoder 3, 4는 입력된 프레임 V(6), V(7)을 압축하고 비트를 출력한다. <그림 2>를 보면 내부 인코더는 순차적으로 동작하면서 병렬로 동작하는 것을 알 수 있다. <그림 1>과 같이 구성하고 <그림 2>와 같이 동작하면 Polyphase 인코더의 내부 인코더 Encoder 1, 2, ..., N으로 입력되는 프레임의 샘플링 속도는 단일 인코더를 사용할 경우 샘플링 속도의 1/N로 감소한다. 따라서 내부

인코더의 개수를 N으로 증가시키면 일반 영상 프레임보다 N배 고속으로 입력되거나 N배 데이터 량이 많은 영상에 대해서도 추가적인 고속 알고리즘 개발 없이 압축할 수 있다.

그러나 이와 같은 방식을 사용할 경우 단일 인코더를 사용하는 경우보다 PSNR이 감소되는 단점을 갖는다. 그 이유를 알아보기 위하여 <그림 2>를 살펴보면 단일 인코더를 사용하는 경우 입력 프레임이 V(0), V(1), V(2),...의 순서이지만 Polyphase 인코더의 내부 인코더 Encoder 1이 사용하는 입력 프레임은 V(0), V(4), V(8),...의 순서이다. 내부 인코더에 입력되는 프레임은 다운 샘플링(down sampling)된 프레임이므로 프레임 사이의 유사성이 단일 인코더로 입력되는 프레임의 유사성보다 감소하게 된다. 일반적으로 프레임 사이의 유사성이 감소하면 움직임 추정을 사용하는 매크로 블록의 개수가 감소하고 해당 매크로 블록을 인트라로 설정하여 오직 이산 코사인 변환 및 양자화에 의해 압축하는 매크로 블록의 개수가 증가하게 된다. 인트라로 처리하는 매크로 블록이 증가하면 발생하는 비트 양이 증가한다. 인코더는 증가된 비트 양을 제어하기 위하여 양자화 스케일의 크기를 증가하게 되므로 PSNR이 감소하게 된다.^[18~19] <그림 3>은 Polyphase 인코더의 내부 인코더 개수 N에 대한 Y 성분의 평균 PSNR을 비교한 것이다. 평균 PSNR 계산을 위해 사용된 시험 영상은 8개이며 서로 다른 움직임 특성을 갖는다.

<그림 3>에서 N = 1의 PSNR은 단일 인코더의 PSNR을 의미하며 단일 인코더에 비해 Polyphase 인코더의 PSNR이 최대 약 4 dB 감소한다는 사실을 알 수 있다. 내부 인코더 개수 N이 증가함에 따라 Polyphase 인코더의 PSNR이 감소하고 N이 4 이상이면 PSNR이 감소하지 않고 일정한 값을 갖는다. 이는 N이 4 이상

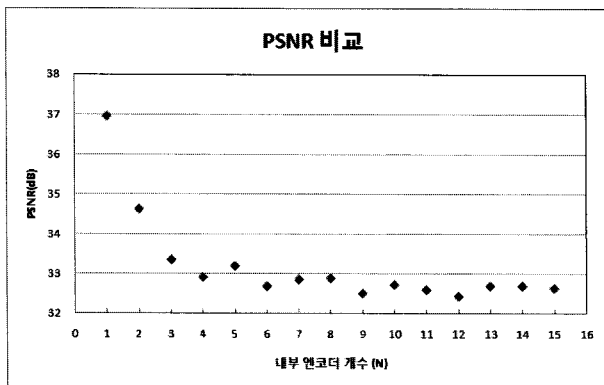


그림 3. 내부 인코더 개수에 대한 PSNR 비교
Fig. 3. PSNR vs Number of internal Encoder.

인 경우 내부 인코더에서 사용하는 프레임 사이의 유사성이 더 이상 감소하지 않고 일정하여 움직임 추정을 사용하지 않고 인트라로 처리하는 매크로 블록의 개수가 증가하지 않음을 의미한다. 본 연구에서는 Polyphase 인코더의 PSNR을 감소시키지 않고 압축하기 위하여 슬라이스 단위로 프레임을 분해하여 압축하는 방식을 제안하며 이에 대한 자세한 설명은 다음 절에서 설명한다.

III. Polyphase 인코더의 PSNR 향상 기법

앞 절에서 설명한 바와 같이 Polyphase 인코더로 압축하는 경우 PSNR이 감소하게 된다. 이 단점을 해결하기 위하여 Polyphase 인코더는 <그림 1>,<그림 2>와 같은 구조와 동작 순서를 갖지만 <그림 4>와 같은 압축 기법을 사용한다. 내부 인코더의 개수가 N이고 입력되는 프레임의 슬라이스 개수가 L 일 때 <그림 4>와 같이 Encoder 1은 프레임 V(0)에 대한 압축을 시작한다. Encoder 1이 프레임 V(0)의 L/N 번째 슬라이스(slice)에 대한 압축이 종료되고 프레임 V(1)이 입력되면 Encoder 2는 V(1)의 1번째 슬라이스에 대한 압축을 시작한다. Encoder 2가 프레임 V(1)의 L/N 번째 슬라이스에 대한 압축을 종료하고 프레임 V(2)가 입력되면 Encoder 3은 V(2)의 1번째 슬라이스에 대한 압축을 시작하는 방식으로 이 과정을 Encoder N까지 반복한다. 이때 Encoder 2 부터는 움직임 추정을 사용하지므로 움직임 벡터의 한계 값을 ± 15로 설정하면 L/N은 식(1)을 만족하여야 한다.

$$\frac{L}{N} \geq 2 \tag{1}$$

식(1)은 $1 < n \leq N$ 일 때 Encoder n-1 이 최소한 2

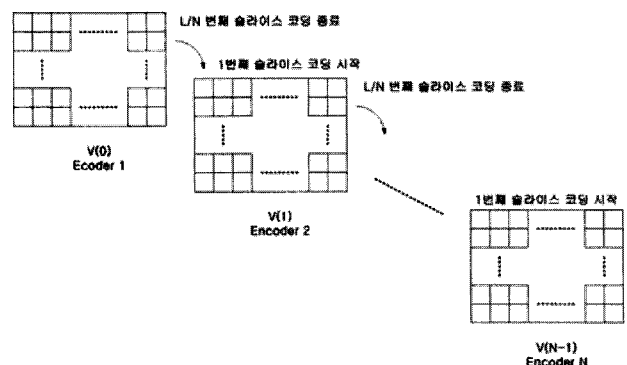


그림 4. Polyphase 인코더 압축 방식
Fig. 4. Coding scheme of the Polyphase encoder.

번째 슬라이스에 대한 압축을 종료하여야 Encoder n이 압축을 시작할 수 있다는 의미이다. 식(1)에서 Polyphase 인코더의 내부 인코더 개수 N은 최대 L/2 개 이다. Encoder 1이 원본 프레임 V(0)을 사용하여 압축을 수행하면 인코더 내부의 디코딩 과정에 의해 재구성된 프레임 $\hat{V}(0)$ 을 매크로 블록 단위로 저장한다. Encoder 1이 최소 2번째 슬라이스에 대한 압축을 종료하는 시점에서 Encoder 2는 V(1)에 대한 움직임 추정을 포함한 압축을 수행한다. 이 때 Encoder 2는 원본 프레임 V(0)과 재구성된 프레임 $\hat{V}(0)$ 을 참조하여 움직임 추정을 수행한다. 따라서 Encoder 1이 실행되는 동안 Encoder 2가 동시에 실행되어도 움직임 벡터의 한계 값이 ± 15 이므로 단일 인코더와 동일한 움직임 추정을 수행할 수 있다. 이 과정은 Encoder N에 대해서도 동일하게 적용된다. 이와 같이 프레임을 슬라이스 단위로 분해하여 압축하는 기법을 사용하면 움직임 추정이 수행될 때 Polyphase 인코더의 프레임 간 유사성이 단일 인코더와 동일하게 유지되므로 압축할 때 인트라로 처리하는 매크로 블록이 증가하지 않게 된다. 그러므로 발생하는 비트 양을 증가하지 않고 양자화 스케일의 크기가 증가하지 않으므로 Polyphase 인코더의 PSNR을 단일 인코더의 PSNR과 동일하게 유지할 수 있게 된다.

IV. Phase의 반복성을 이용한 프레임 전송량 감소 방법

앞 절에서 설명한 슬라이스 단위로 분해하여 압축하는 기법을 구현하기 위하여 GOP (Group of Pictures) 내의 프레임 개수 15, I/P 프레임과 P 프레임 사이의 간격 $M = 3$, Polyphase 인코더의 내부 인코더 개수 $N = 2$ 일 경우 현재 프레임에 대한 P 프레임의 순방향 참조 프레임과 B 프레임에서의 순방향 및 역방향 참조 프레임을 <그림 5>와 같이 정리하였다. 압축되는 프레임의 종류와 순서는 I(0), P(3), B(1), B(2), P(6), B(4), B(5), ... 이므로 <그림 5>의 사각형에 현재 프레임을 표시하였고 순방향 참조 프레임은 사각형 위에 타원으로 표시하였으며 역방향 참조 프레임은 사각형 아래에 타원으로 표시하였다. 그리고 이전 phase에서 사용되거나 저장되어 추가로 전송받을 필요가 없는 프레임은 점선 타원으로 표시하였다. <그림 5>의 phase 0에서 Encoder 1은 입력되는 프레임을 I 프레임으로 압축하고 저장한다. Encoder 1의 I(0)은 P 프레임을 압축하는 Encoder

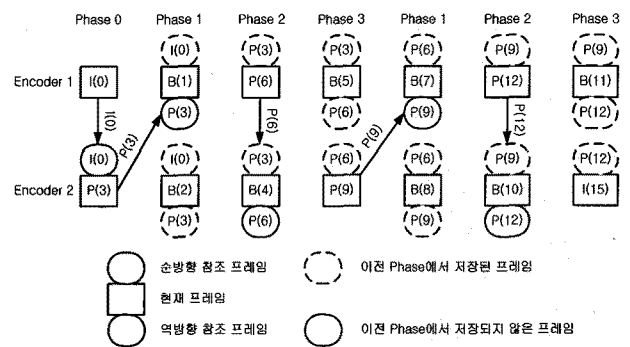


그림 5. Phase와 전송 경로의 반복 (N = 2)
 Fig. 5. The repetition of the phase and the path. (N = 2)

2에서 순방향 참조에 사용되므로 Encoder 2로 전송된다. Encoder 2는 전송된 I(0) 프레임을 순방향 참조 프레임으로 사용하여 압축하고 phase 1의 Encoder 1로 전송한다.

phase 1에서는 Encoder 1, 2가 모두 B 프레임으로 압축한다. phase 1에서 Encoder 1은 phase 0에서 이미 저장된 I(0)을 순방향 참조 프레임으로 사용하고 phase 0의 Encoder 2가 전송한 P(3)을 역방향 참조 프레임으로 사용하여 압축한다. phase 1에서 Encoder 2는 phase 0에서 사용된 I(0)을 순방향 참조 프레임으로 사용하고 P(3)을 역방향 참조 프레임으로 사용하여 B 프레임을 압축한다. 따라서 phase 1에서 Encoder 1이 전송받는 프레임은 I(0), B(1), P(3)이 아니라 B(1), P(3) 프레임으로 감소하고 Encoder 2가 전송받는 프레임은 I(0), B(2), P(3)이 아니라 B(2) 프레임으로 감소된다. phase 2에서 Encoder 1은 phase 1에서 역방향 참조 프레임으로 사용한 P(3)을 순방향 참조 프레임을 사용하여 P(6)을 압축하고 Encoder 2로 전송한다. phase 2에서 Encoder 2는 이전 phase에서 역방향 참조 프레임으로 사용한 P(3)을 순방향 참조 프레임으로 사용하고 P(6)을 역방향 참조 프레임으로 사용하여 B(4)를 압축한다. phase 2에서 Encoder 1이 전송받는 프레임은 P(3), P(6)이 아니라 P(6) 프레임으로 감소하고 Encoder 2가 전송받는 프레임은 P(3), B(4), P(6)이 아니라 B(4) 프레임으로 감소된다. 이 과정을 B(11)까지 수행하였을 때 전송받는 프레임 개수는 58개에서 22개로 감소되어 약 62%의 데이터 전송량을 감소시킬 수 있다. <그림 5>를 관찰하면 phase 0을 제외하고 phase 1, 2, 3이 반복되는 것을 알 수 있다. 즉, phase 1에서 Encoder 1, 2가 압축하는 프레임은 모두 B 프레임이고 이전 phase에서 저장된 프레임을 참조용 프레임으로 사용한다.

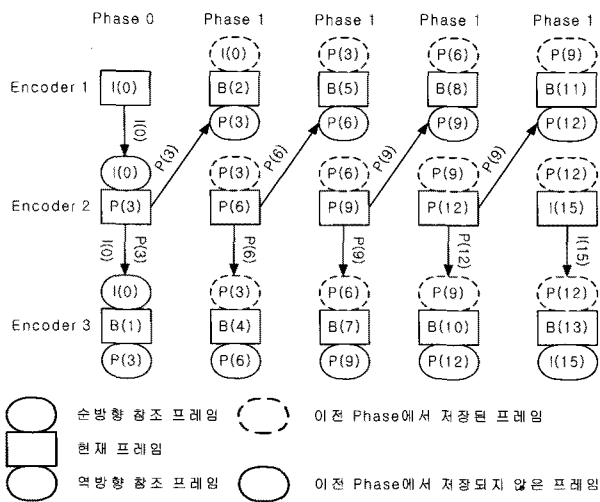


그림 6. Phase와 전송 경로의 반복 (N = 3)
 Fig. 6. The repetition of the phase and the path. (N = 3)

phase 2에서는 Encoder 1, 2가 압축하는 프레임이 각각 P, B 프레임이고 이전 phase에서 저장된 프레임을 참조 프레임으로 사용한다. phase 3에서는 Encoder 1, 2가 압축하는 프레임이 각각 B, P 프레임이고 이전 phase에서 저장된 프레임을 참조 프레임으로 사용한다. 이 과정은 무한히 반복되며 다만 phase 3에서 I/P 프레임이 서로 다른 것을 볼 수 있는데 이는 각 Encoder에서 순번에 의해 현재 입력된 프레임이 I 프레임인지 P 프레임인지 알 수 있으므로 현재 프레임이 I 프레임이면 움직임 추정을 수행하지 않고 P 프레임이면 움직임 추정을 수행하게 되므로 동일한 패턴이 될 수 있다. 지금까지 설명한 내용을 N = 3에 대하여 적용하면 <그림 6>과 같다. N = 3인 경우 phase 0을 제외하고 phase 1이 반복됨을 알 수 있고 전송받는 프레임 개수도 감소시킴을 알 수 있다. GOP내 프레임 개수와 I 프레임과 P 프레임의 간격이 결정되면 나머지는 모두 B 프레임이므로 프레임 패턴은 반복성을 갖는다. 따라서 I/P 프레임과 P 프레임의 간격이 M인 경우 내부 인코더의 개수 N을 증가하여도 생성되는 phase는 0, 1, 2, ... M으로 제한된다. Polyphase 인코더의 내부 인코더 Encoder 1, 2, 3, ... N이 압축하는 I, P, B 프레임이 시간에 따라 변화하고 순방향 및 역방향으로 참조하는 프레임이 변경되어 다수의 phase가 발생하지만 지금까지 설명한 phase의 반복성을 이용하면 이전 phase에서 사용된 프레임을 참조 프레임으로 사용할 수 있으므로 각 인코더로 전송되는 데이터량을 감소시킬 수 있다.

V. 시뮬레이션 결과

Polyphase 인코더의 PSNR 성능을 검증하기 위하여 여러 시험용 영상에 대한 시뮬레이션을 수행하였으며 이 때 입력되는 영상 프레임의 크기는 352×288의 CIF (Common Intermediate Format)이지만 속도는 일반적으로 사용되는 영상 소스의 속도보다 N 배 빠른 고속 카메라에서 출력되는 조건을 적용하였다. 또한 앞 절에서 설명한 phase의 반복성을 이용하여 시뮬레이션을 구현하였다. Polyphase 인코더에서 발생하는 총 비트량을 단일 인코더와 동일하게 설정하여 시뮬레이션을 수행하였다. 그리고 다중 스레드(multi-thread) 기법을 사용하여 N개의 인코더가 동시에 작동하도록 시뮬레이션을 수행하였다. <그림 3>에서 보는 바와 같이 내부 인코더 개수 N이 4 이상이면 PSNR이 더 이상 감소하지 않고 일정한 값을 유지하므로 최소 PSNR 값을 갖는 N = 4에 대한 시뮬레이션을 수행하였다. 여러 시험용 영상에 대한 시뮬레이션을 수행하여 PSNR을 비교하였으며 비트율 1Mbps에 대한 평균 PSNR은 <표 1>과 같다. <표 1>에서 보는 바와 같이 1개 인코더를 사용하는 단일 인코더와 Polyphase 인코더의 PSNR이 동일함을

표 1. 평균 PSNR 비교 (N = 4)
 Table 1. Comparison of average PSNR (N = 4).

	단일 인코더			Polyphase 인코더		
	Y	U	V	Y	U	V
Football	28.5	33.5	36.0	28.5	33.5	36.0
Coastguard	35.2	46.7	48.1	35.2	46.7	48.1
Container	39.1	45.2	45.1	39.1	45.2	45.1
Foreman	37.1	43.3	44.2	37.1	43.3	44.2
Garden	27.3	31.3	33.3	27.3	31.3	33.3
Hall Monitor	39.9	43.7	45.9	39.9	43.7	45.9
Mobile	26.8	30.7	30.4	26.8	30.7	30.4
Table Tennis	31.2	39.0	37.7	31.2	39.0	37.7
평균	33.1	39.2	40.1	33.1	39.2	40.1

표 2. 전송되는 프레임 개수
 Table 2. Number of the transmitted frames.

내부 인코더 개수	프레임 전송 회수	감소율
2	22	62%
3	32	44%
4	41	29%

알 수 있다.

또한 Polyphase 인코더의 내부 인코더 개수에 따른 프레임 전송 회수를 <표 2>에 정리하였다.

Polyphase 인코더 방식을 사용하지 않고 여러 개의 인코더를 병렬로 압축하는 방식을 사용할 경우 각 인코더가 압축을 위해 필요한 총 프레임 전송 회수는 GOP 내에서 58회이다. 그러나 Polyphase 인코더 방식을 사용하는 경우 전송 회수가 <표 2>에서 보는 바와 같이 감소함을 알 수 있다.

VI. 결 론

본 연구에서는 고속 카메라에서 출력되는 영상 데이터 또는 대용량 영상 데이터를 압축하기 위한 Polyphase 인코더를 제안하였다. Polyphase 인코더는 분배기에 의해 1/N 만큼 다운샘플링 (downsampling)된 영상 프레임을 내부 인코더가 압축하기 때문에 대용량의 영상 데이터를 처리할 수 있으나 프레임 간의 유사성이 감소하여 PSNR 이 감소하고 각 인코더로 참조 프레임을 전송하기 위한 데이터 량이 증가하는 단점을 갖는다. 본 연구에서는 그 문제점을 분석하였고 단일 인코더와 동일한 PSNR을 유지할 수 있도록 참조 프레임의 진행 상황에 따라 현재 프레임을 압축하는 방법을 제시하였다. 이와 같이 프레임을 순차적으로 분배하면 비트 스트림의 동기화가 필요 없으며 슬라이스 단위로 압축하는 경우보다 인코더로 전송되는 데이터 량을 감소시킬 수 있으며 디코더로 출력되는 화면에 경계면이 보이지 않게 된다. 또한 압축률을 높이기 위하여 B 프레임 사용이 가능하다. 마지막으로 여러 시험용 영상에 대한 시뮬레이션을 수행하여 Polyphase 인코더의 PSNR이 단일 인코더와 동일하고 프레임의 전송량이 감소됨을 보였다.

참 고 문 헌

- [1] Dzung, Jeffrey, "Efficient Algorithms for MPEG Video Compression", John Wiley & Sons, pp41-69, 2002
- [2] Detlev Marpe, Thomas Wiegand, "The H.264/MPEG-4 Advanced Video Coding Standard and its Applications", IEEE Comm. Magazine, Aug. 2006, pp134-143.
- [3] Shan Zhu, Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Trans. Image. vol. 9, No. 2, Feb. 2000, pp287-290.
- [4] R. Li, B. Zeng, and M. Liou, "A new three-step search algorithm for block motion estimation", IEEE Trans. CASVT, vol. 4, no. 4, Aug. 1994, pp 438-442.
- [5] Jun-Yup Kang, Sandeep Gupta, "An Efficient Data-Distribution Mechanism in a Processor-In-Memory(PIM) Architecture Applied to Motion Estimation", IEEE, Trans. Computers, vol 57, No. 3, Mar 2008, pp375-388
- [6] Shahriar M, Akramullah, "Performance of software-based mpeg-2 video encoder on parallel and distributed systems", IEEE Trans. CASVT, vol 7, No. 4, Aug 1997, pp687-695.
- [7] Zhuo Zhao, Ping Liang, "Data Partition for Wavefront Parallelization of H.264 Video Encoder", IEEE International Symposium on Circuits and Systems, 2006, pp2669-2672.
- [8] Mattias O'Nils, Benny Thornberg, "Data Partitioning for Parallel Implementation of Real-Time Video Processing Systems", Proceedings of the 2005 European Conference on Circuit theory and Design, pp213-216.
- [9] N. H. C. Yung, K. C. Chu, "Fast and Parallel Video Encoding by Workload Balancing", IEEE International Conference on System, Man, and Cybernetics, 1998, pp4642-4647.
- [10] Xun He, Xiang zhong Fang, Ci Wang, "Parallel HD Encoding on CELL", IEEE International Symposium on Circuits and Systems, 2009, pp1065-1068.
- [11] Jonghan Park, Soonhoi Ha, "Performance Analysis of Parallel Execution of H.264 Encoder on the Cell Processor", IEEE Workshop on Embedded Systems for Real-Time Multimedia, 2007, pp27-32.
- [12] Yong He, Ishfaq Ahmad, Ming L. Liou, "A Software-Based MPEG-4 Video Encoder Using Parallel Processing", IEEE Trans. CASVT, vol 8, No. 7, Nov 1998, pp909-920.
- [13] Ping Li, Bharadwaj Veeravalli, "Design and Implementation of Parallel Video Encoding Strategies Using Divisible Load Analysis", IEEE Trans. CASVT, vol 15, No. 9, SEP 2005, pp1098-1112.
- [14] Nogaki, S, "A Study on Rate Control Method for MP@HL Encoder with Parallel Encoder Architecture Using Picture Partitioning", IEEE International Conference on Image Processing, 1999, pp261-265.
- [15] Nelson H. C Yung, Kwong-Keung Leung,

“Spatial and Temporal Data Parallelization of the H.261 Video Coding Algorithm”, IEEE Trans. CASVT, vol 11, No. 1, Jan 2001, pp91-104.

[16] Changying Luo, Jinguang Sun, Zhiyong Tao, “The Research of H.264/AVC Video Encoding Parallel Algorithm”, Sencond International Symposium on Intelligent Technology Applications , 2008, pp201-205.

[17] Alan V. Oppenheim, Ronald W. Schafer, “Discrete -Time Signal Processing”, Prentice Hall, 2nd, pp 179-184.

[18] Keesman, G and Shah, “Bit-rate control for MPEG encoders”, EURASIP Signal Processing : Image Communication.

[19] Yoo-Sok Saw, “Rate-Quality Optimized Video Coding”, Kluwer Academic Publishers, pp66-68, 1999.

저 자 소 개



조 동 식(정회원)
 1990년 단국대학교 전기공학과
 학사 졸업.
 1992년 단국대학교 전기공학과
 석사 졸업.
 1993년~현재 국방과학연구소
 선임연구원.

<주관심분야 : 통신, 신호처리, 영상 처리>



나 성 응(정회원)
 1976년 서울대학교 전기공학과
 학사 졸업.
 1978년 한국과학기술원 전기및
 전자공학과 석사 졸업.
 1992년 한국과학기술원 전기및
 전자공학과 박사 졸업.

1980년~현재 충남대학교 전기정보통신공학부
 교수

<주관심분야 : 통신, 디지털 신호처리, 영상처리>