논문 2005-42SP-4-16

움직임 벡터와 인트라 예측 모드를 이용한 디지털 비디오 스크램블링 방법

(Digital Video Scrambling Methods using Motion Vector and Intra Prediction Mode)

안 진 행*, 전 병 우**

(Jinhaeng Ahn and Byeungwoo Jeon)

요 약

본 논문에서는 디지털 컨텐츠 보호 기술 중 하나인 두 가지 디지털 비디오 스크램블링 방법을 제안한다. 그 중 한 가지는 움직임 벡터를 이용하여 인터 블록을 왜곡하는 스크램블링 방법이며, 다른 한 가지는 H.264 비디오 압축 기술의 인트라 예측 모드를 이용하여 인트라 블록을 왜곡하는 스크램블링 방법이다. 움직임 벡터를 이용한 스크램블링 방법은 움직임 벡터의 수평값과 수직값을 교환하는 것으로 MPEG-1, 2, 4, H.264와 같은 대부분의 비디오 압축 기술에 적용가능하다. 인트라 예측 모드를 이용한 방법은 H.264 비디오 압축 기술의 특징인 인트라 예측 부호화를 이용한 것으로, 인트라 예측 부호화시 발생하는 인트라 예측 모드를 통상적인 복호화가 가능하며 비트율의 변화가 없는 법위 내에서 랜덤하게 변경하는 것이다. 두 가지 방법 모두 스크램블링으로 인한 압축 효율의 저하가 전혀 없으며, XOR과 같은 매우 간단한 연산만으로 구현이 가능하므로 계산량의 증가가 적다. 뿐만 아니라, 인트라 블록 스크램블링의 경우 인터 블록에 대한 직접적인 왜곡 없이 에러 전파 효과로 인해 간접적으로 인터 블록을 왜곡할 수 있는 장점을 갖고 있다. 본 논문에서는 이와 같은 두 가지 새로운 디지털 비디오 스크램블링 방법을 제안하고, 이에 대한 실험 결과를 통해 제안된 알고리듬의 효율성을 보인다.

Abstract

In this paper, two digital video scrambling methods are proposed as simple means of the digital content protection techniques. One is inter block scrambling using motion vector, the other is intra block scrambling using intra prediction mode. The proposed inter block scrambling method distorts the original sequences by swapping horizontal and vertical components of motion vector. This method can be applied on most common video coding techniques such as MPEG-1, 2 ,4, H.264, etc. The proposed intra block scrambling method distorts the original sequences by modifying intra prediction mode that is property of H.254 video coding technique. Both methods do not cause any bit rate increase after scrambling. Moreover, they have low complexity because they need only simple operation like XOR. Especially, the proposed intra block scrambling does not distort inter blocks directly. But inter blocks are distorted by error propagation effect as much as intra blocks. This paper introduces two new digital video scrambling method and verifies its effectiveness through simulation.

Keywords : Scrambling, Motion Vector, Intra prediction mode

I. 서 론

디지털 콘텐츠의 사용이 급증함에 따라 이에 대한 보호의 중요성이 커지고 있다. 특히 위성이나 케이블, 인터넷 망과 같은 공개적인 경로를 통해 유포되는 디지털 비디오의 경우, 이를 위한 보호 기술의 필요성이 더욱 강조되고 있다. 디지털 비디오 스크램블링은 이러한 디지털 비디오 보호 기술 중 하나로, 보호 하고자 하는 영상 데이터를 특정한 키(key)에 의해 변형 또는 암호화하여 전송함으로써 이를 복구할 수 있는 키를 가진 수신자만이 정상적으로 영상을 복원할 수 있도록 하는 기

* 학생회원, ** 정회원, 성균관대학교 전자전기공학과
(School of Electronic Electrical Engineering, Sungkyunkwan University)

※ 본 논문은 2003년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2003-041-D20405)
접수일자: 2005년1월13일, 수정완료일: 2005년5월6일

술이다. 즉, 허가되지 않은 수신자의 경우 수신된 영상을 복호화 하더라도, 원 영상이 아닌 스크램블링 과정으로 인해 왜곡된 영상을 보게 됨으로써 정당한 수신자의 권리를 보호할 수 있도록 한다. 이와 같은 스크램블링 기술은 현재 아날로그 방송에서 실제 사용되고 있으며, 최근 들어 디지털 방송이 상용화 되어감에 따라 디지털 비디오를 위한 스크램블링 기술에 대한 연구가 이루어지고 있다[1]-[5].

기존의 제안된 디지털 비디오 스크램블링 방법 중 움직임 벡터를 이용한 스크램블링 방법[5]은 전송 블록 형태 값(Coded Block Pattern)과 움직임 벡터값을 이용한 방법으로 다음과 같다. 각 매크로블록의 전송 블록 형태 값을 구한 다음, 현 매크로블록에서 추정된 움직임 벡터와 이전 매크로 블록에서 추정된 예측 움직임 벡터(Predictive Motion Vector)와의 차이를 통해 차동 움직임 벡터(Differential Motion Vector)를 구한다. 앞서 구한 차동 움직임 벡터를 가변 길이 부호화(Variable Length Coding)하기 전에 전송 블록 형태 값을 모듈러 33을 취한 후, 부호화 테이블 상에서 그 값만큼 떨어져 있는 부호어를 이용하여 부호화한다. 그러나 이와 같은 방법은 전송 블록 형태 값을 모듈러 연산한 결과만큼 이동한 후의 부호어의 길이가 기존의 부호어의 길이 보다 늘어 날 수 있으므로 스크램블링 후 비트의 양이 증가 할 수 있다는 단점이 있다.

혹은 DES나 AES와 같은 암호화 알고리듬을 이용한 스크램블링 방법이 있다[1]. 즉, DES나 AES와 같은 알고리듬을 사용하여 압축 영상 신호 자체를 암호화 하는 것이다. 하지만 DES나 AES와 같은 암호화 알고리듬은 계산량이 매우 많아, 압축 영상 신호 전부를 암호화 할 경우 스크램블링으로 인한 계산량이 매우 크게 증가하게 된다. 이를 보완하기 위해 영상의 중요도에 따라 크게 네 가지 레벨로 나누어 압축 영상 신호를 선택적으로 스크램블링 하기도 한다. 가장 중요한 영상의 경우 모든 영상 신호를 암호화하고, 이보다 한 단계 낮은 중요도를 갖는 영상의 경우에는 인트라 프레임 또는 인트라 블록만을 암호화한다. 인트라 프레임 또는 인트라 블록을 복호화 할 수 없다면, 인터 블록 또한 복호화 할 수 없기 때문이다. 그 다음 단계의 중요도를 갖는 영상의 경우에는 움직임 벡터 또는 DCT 변환 계수 등과 같은 중요한 파라미터 값 들을 암호화하며, 가장 낮은 중요도를 갖는 영상의 경우에는 모든 헤더 정보를 암호화한다. 그러나 아무리 적은 양의 영상 신호를 암호화한다 하더라도 암호화 알고리듬 자체의 계산량이 매우 크며, DES나 AES에 의한 암호화를 복호화 할 수 있는 키를 가지고 있지 않은 수신자의 경우 수신된 영상 신호를 전혀 복호화 할 수 없게 된다.

그 밖에 웨이블릿 변환이나 DCT 변환을 통한 주파수 영역에서의 여러 가지 스크램블링 방법이 있다[1]. 웨이블릿 기반의 스크램블링 방법은 웨이블릿 변환으로 인해 생성된 부대역(Subband)내에서 블록을 나누어, 특정한 테이블을 기준으로 블록내의 계수값들을 섞어 영상을 왜곡한다. 또는 비트 플레인(Bit Plane)에서 중요한 비트(Significant Bit)나 부호 비트를 선택적으로 섞는 방법이 있다. 하지만 MPEG-1, 2, 4나 H.264와 같은 대부분의 비디오 영상 압축 알고리듬에는 웨이블릿 변환이 포함되지 않으므로 일반적인 비디오 압축 기술에 적용하는 데는 어려움이 있다. 주파수 영역에서의 또 다른 스크램블링 방법인 DCT 기반의 스크램블링 방법은 웨이블릿 기반의 스크램블링 방법과 마찬가지로 변환 계수들을 섞어 영상을 왜곡한다. 예를 들어, 한 프레임 내의 DC 계수값만을 모아 특정한 테이블을 통해 DC 계수값들을 섞거나, 동일 주파수 위치의 계수값들을 모아 웨이블릿 변환에서의 부대역 개념과 유사하게 주파수층을 만들어 동일 주파수 내에서 계수값들을 섞는다. 이렇게 주파수 영역에서 변환 계수값을 섞는 방법은 앞서 설명한 움직임 벡터를 이용한 방법이나 DES와 같은 일반적인 암호화 알고리듬을 이용한 방식과 마찬가지로 스크램블링으로 인해 압축 효율이 떨어 질 수 있다는 단점을 가지고 있다.

본 논문은 이와 같은 기존 방식의 문제점을 개선하여 스크램블링 후 압축 비트 스트림의 양이 전혀 증가 하지 않는 두 가지 스크램블링 방법을 제안한다. 제안된 알고리듬은 움직임 벡터를 이용한 인터 블록 스크램블링 방법과 H.264 비디오 압축 기술의 인트라 예측 모드를 이용한 인트라 블록 스크램블링 방법으로, 두 가지 방법 모두 간단하며 효율적으로 원 영상을 왜곡한다. 움직임 벡터를 이용한 방법은 차동 움직임 벡터의 수평 성분과 수직 성분의 교환을 통해 영상을 왜곡하는 방법으로 구체적인 스크램블링 과정은 Ⅱ장에서 서술한다. 인트라 예측 모드를 이용한 방법은 H.264 비디오 압축 기술의 특징인 인트라 예측 부호화를 이용한 방법으로, 인트라 예측 부호화시 사용되는 인트라 예측 모드를 변경하여 영상을 왜곡한다. H.264 비디오 압축 기술에는 인트라 16x16 부호화와 인트라 4x4 부호화 방법이 존재하며 각 방법에 따른 구체적인 스크램블링 과정은 Ⅲ장에서 서술한다. Ⅳ장에서는 제안된 알고리듬을 사용

한 실험 결과를 설명하고, 끝으로 V장에서는 이에 대한 결론을 내린다.

Ⅱ. 움직임 벡터를 이용한 인터 블록 스크램블링 및 디스크램블링 방법

일반적인 비디오 압축 부호화 기술은 움직임 벡터를 추정한 후, 추정된 움직임 벡터를 바로 전송하지 않고 예측 움직임 벡터와의 차이값인 차동 움직임 벡터를 가변 길이 부호화하여 전송한다. 제안하고자 하는 방법은 차동 움직임 벡터의 수평 성분과 수직 성분 교환하여 전송함으로써, 이에 대한 교환 여부를 알지 못하는 수신자의 경우 올바른 움직임 벡터를 복원 할 수 없도록 한다. 따라서 허가되지 않은 수신자의 경우 잘못된 움직임 벡터 정보로 인해 왜곡된 영상을 보게 된다. 구체적인 스크램블링 과정은 다음과 같다.

우선 특정한 키를 사용하여 의사 랜덤 시퀀스를 발생시킨다. 그 다음, 각 인터 블록마다 차동 움직임 벡터를 부호화하기 전에, 의사 랜덤 시퀀스로부터 한 비트를 읽어 들인다. 만약 읽은 비트가 '1'이면 차동 움직임 벡터의 수평 성분과 수직 성분을 교환하고, '0'이면 아무런 변화 없이 차동 움직임 벡터를 부호화한다. 즉, 의사 랜덤 시퀀스에서 순차적으로 비트를 읽어, 읽은 비트가 '1'일 경우에만 움직임 벡터의 수평 성분과 수직 성분을 교환한다.

디스크램블링 과정은 스크램블링 과정과 매우 유사하다. 우선 특정한 키를 사용하여 스크램블링시 사용한 것과 동일한 의사 랜덤 시퀀스를 발생시킨다. 각 인터 블록의 차동 움직임 벡터를 복호화하기 전에, 의사 랜덤 시퀀스로부터 한 비트를 읽는다. 만약 읽은 비트가 '1'이면 수신된 차동 움직임 벡터의 수평 성분과 수직 성분을 교환하고, '0'이면 아무런 변화를 주지 않는다. 즉, 스크램블링 시 교환한 차동 움직임 백터의 수평 성분과 수직 성분을 다시 역 교환함으로써 본래의 차동 움직임 벡터로 복원하는 것이다. 이때 사용되는 의사 랜덤 시퀀스는 허가된 수신자만이 알 수 있도록 하여, 합법적인 수신자만이 원 영상을 복원할 수 있도록 한다.

이와 같은 방법은 차동 움직임 벡터의 수평 성분과 수직 성분을 교환함으로써, 비트량의 증가 없이 매우 효과적으로 스크램블링 할 수 있다. 뿐만 아니라 복잡한 연산이 필요 없으므로 계산량 증가가 거의 없다. 또한 움직임 벡터는 MPEG-1, 2, 3, H.264와 같은 일반적인 비디오 압축 기술에서 인터 예측 부호화를 위해 사용하는 파라미터이므로, 제안된 알고리듬은 대부분의 비디오 압축 기술에 적용 될 수 있다. 그리고 임의의 공격자가 모든 인터 블록의 움직임 벡터의 수평 성분과 수직 성분을 교환하여 본래의 영상을 복원하는 시도를 한다 해도, 쉽게 원 영상을 복원할 수는 없을 것이다. 왜냐하면, 움직임 벡터의 수평 성분과 수직 성분이 스크램블링 과정에 의해 교환되었는지 여부를 판단할 수 있는 척도가 정해져 있지 않기 때문이다. 교환 여부를 판단하는 가장 좋은 방법은 사람의 눈으로 식별하는 것인데, 각 블록마다 눈으로 영상의 왜곡 여부를 판단하기는 매우 힘들 것이다.

Ⅲ. H.264 인트라 예측 모드를 이용한 인트라 블록 스크램블링 및 디스크램블링 방법

MPEG 1, 2, 4와 같은 비디오 압축 기술과 달리 H.264 비디오 압축 기술에서는 인트라 예측 부호화 시, 압축 효율을 높이기 위해 인트라 블록의 예측값을 추정하여 사용한다. 인트라 블록 예측값을 구하기 위한 인트라 예측 방향을 결정하는 것은 특정한 방법이 정해진 것이 아니라, 부호화기에 따라 최적의 방법을 선택하여 사용 할 수 있다. 일반적인 인트라 예측 모드 결정 방법은 정해진 방향(모드)별로 현재 블록의 화소값과 예측값과의 에러 값을 계산하여 가장 작은 에러 값을 갖는 예측값의 방향을 인트라 예측 모드로 결정한다. 인트라 예측 모드가 정해지면, 결정된 모드에 해당하는 규칙을 통해 구한 예측값과 현재 블록의 화소값들과의 감산 연산을 통해 잔여값(Residual)을 구한다. 이렇게 생성된 잔여값과 예측 모드를 부호화하여 수신측에 전송하게 된다. 수신측에서는 전송받은 모드를 통해 부호화 시 사용된 것과 동일한 예측값을 구한 후, 수신된 잔여값과 앞서 구한 예측값을 서로 더하여 원 영상을 복원한다. 따라서 잘못된 모드를 전송 받았을 경우, 일반적인 복호화는 가능하지만 부호화 과정에서 사용된 예측값과 다른 값을 사용하게 되므로 영상이 왜곡된다. 제안된 방법은 이와 같은 특징을 이용하여, 인트라 블록 부호화 과정에서 사용되는 인트라 예측 모드를 변경하여 전송함으로써 원래의 모드를 알지 못하는 수신자의 경우 원 영상을 복원 할 수 없도록 한다. H.264 비디오 압축 기술의 인트라 부호화 방법에는 인트라 4x4 부호화와 인트라 16x16 부호화 두 가지 방법이 존재하며, 각 방법에 따른 구체적인 스크램블링 및 디스크램블링 알고

리듬은 다음과 같다.

1. 인트라 4x4 부호화

인트라 4x4 예측 모드는 그림 1과 같이 8가지 방향의 모드와 모드 2인 DC 모드를 포함하여 총 9가지 모드가 있다.

각 모드에 따라 그림 2와 같이 4x4 블록 내의 16개의 화소(a~p)를 좌측 및 상위 블록의 화소(A~M)를 이용하여 예측한다. 예를 들어 모드 0의 경우 수직 모드에 해당하며, a, c, I, m 화소의 예측값으로 A 화소값을, b, f, j, n 화소의 예측값으로는 B 화소값을, c, g, k, o 화소의 예측값으로는 C 화소값을, d, h, l, p 화소의 예측값으로는 D 화소값을 사용한다. 이와 유사한 방식으로 각 모드에 따른 방향에 해당하는 주변블록을 사용하여 예측값을 정하게 된다. 단, 모드 2의 경우는 DC 모드로서 유효한 좌측 또는 상위 주변 화소들의 평균값을 사용하게 된다. 만약 모든 주변 화소들이 유효하지 않다면 예측값으로 128을 사용한다.

9가지 모드 중 최적의 모드를 정하기 위해 모드 0부터 모드 8까지 각 모드에 따라 결정되는 예측값들을 이용하여 9가지 경우의 예측 에러 값을 계산 한다. 그 중

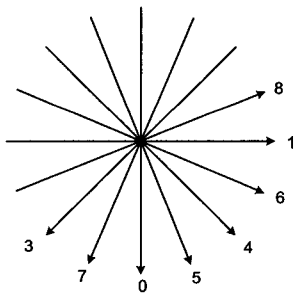


그림 1. H.264 부호화의 인트라 4x4 예측 모드 방향
Fig. 1. Intra 4x4 prediction mode directions of H.264.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

그림 2. 인트라 4x4 예측에서 사용되는 화소와 주변 화소
Fig. 2. Pixels used in the intra 4x4 prediction and its neighbor pixels.

가장 작은 에러 값을 갖는 모드가 인트라 4x4 예측 모드로 결정된다. 이렇게 결정된 예측 모드는 고정 길이 부호화(Fixed Length Coding)하여 전송되며, 현재 블록에서 예측값을 뺀 잔여값은 가변 길이 부호화하여 전송된다. 이 때, 발생 가능한 예측 모드는 총 9 가지이므로 예측 모드를 고정 길이 부호화하기 위해서는 총 4 비트가 필요하다. 하지만 만약 발생 가능한 예측 모드가 8가지라면 3 비트만으로 모든 경우의 예측 모드를 표현할 수 있으므로 예측 모드의 부호화를 위해 보다 적은 양의 비트를 사용할 수 있게 된다. 이를 위해 H.264의 인트라 4x4 부호화에서는 플래그 비트를 사용하여 압축 효율을 높인다. 플래그 비트는 주변 블록의 모드와의 관계를 나타내는 비트로, 현재 블록의 모드 정보를 부호화하기 위해서 좌측 블록과 상위 블록의 모드 정보를 이용한다. 좌측 블록과 상위 블록의 예측 모드 가운데 값이 적은 모드를 '최우선모드(Most Probable Mode)'로 결정한다. 최우선모드와 현재 블록의 예측 모드를 비교하여, 두 모드가 서로 같으면 앞서 서술한 플래그 비트를 '1'로 하여 전송한다. 만약 같지 않으면 '0'으로 하여 전송한 다음, 추가로 나머지 8가지 경우 중 현재 모드를 3비트의 부호어로 고정 길이 부호화하여 전송한다. 단, 전송되는 현재 모드는 최우선모드와 비교하여 현재 모드가 같거나 클 경우, 현재 모드에 1을 뺀 값을 부호화한다. 예를 들어, 좌측 블록의 예측 모드가 2이고 상위 블록의 예측 모드가 5이며 현재 블록의 예측 모드는 3이라고 가정하자. 이때, 최우선모드는 2과 5의 최소값인, 2가 된다. 2는 현재 블록의 예측 모드인 3과 같지 않다. 따라서 플래그 비트를 '0'으로 하여 보내고, 추가로 현재 블록의 예측 모드 전송한다. 단, 현재 블록의 예측모드는 3으로 최우선모드 2보다 크다. 따라서 3에서 1을 뺀 2을 고정 길이 부호화하여 '010'을 전송하게 되는 것이다.

본 논문에서 제안하고자 하는 스크램블링 방법은 위와 같은 과정을 거쳐 최종적으로 전송하게 되는 3비트의 고정길이 부호어를 변경함으로써 원 영상을 왜곡한다. 모드 정보를 표현하는 3비트의 고정길이 부호어는 0부터 7까지의 수를 표현하므로, 원래 모드 정보와 다른 0부터 7까지의 임의의 수로 변경하면 원래의 모드를 알지 못하는 수신측은 원 영상을 복원할 수 없게 되는 것이다. 구체적인 스크램블링 과정은 다음과 같다.

우선 플래그 비트를 사용해서 예측 모드의 전송 여부를 확인한다. 만약 플래그 비트가 '1'이라면 예측 모드를 전송하지 않고 플래그 비트만 전송하는 경우이므로

스크램블링 할 수 없다. 따라서 예측 모드가 전송되는 플래그 비트가 '0'인 경우에만 예측 모드 변경이 가능하다. 예측 모드 변경을 위해 우선 특정한 키를 사용해 의사 랜덤 시퀀스(Pseudo Random Sequence)를 발생시킨다. 그 다음, 전송할 모드 정보인 3비트 고정길이 부호어와 앞서 발생시킨 의사 랜덤 시퀀스에서 순서대로 읽어 들인 3비트를 수식 (1)과 같이 XOR (Exclusive OR) 연산을 통해 새로운 모드를 얻어 전송한다.

$$Mode_{new} = Mode_{org} \oplus 3\,bit\ random\ sequence \quad (1)$$

여기서 ⊕은 XOR 연산을 의미한다. 예를 들어, 전송해야할 3비트 모드 정보가 '010'이고 의사 랜덤 시퀀스에서 읽어 들인 3비트가 '110'이라면 새로운 모드 정보는 '100'이 된다. 따라서 XOR연산의 특성에 의해 동일한 의사 랜덤 시퀀스를 가진 사용자만이 원래 모드를 복원할 수 있으며, 새로운 모드 또한 3비트이므로 비트량이 전혀 증가하지 않는다. 또한 일반적인 복호화 과정에 전혀 지장을 주지 않기 때문에 원래의 예측 모드를 모르는 수신자의 경우 일반적인 복호화는 가능하지만, 잘못된 예측값으로 인해 왜곡된 영상을 보게 된다. 이와 같은 방법은 앞서 설명한 바와 같이 비트량의 증가가 전혀 없으며, 스크램블링 시 사용되는 XOR 연산은 구현이 쉬우며 계산량이 적기 때문에 이로 인한 복잡도의 증가가 매우 적다.

인트라 4x4 블록의 디스크램블링 과정은 스크램블링 과정과 매우 유사하다. 복호화 하고자 하는 인트라 4x4 블록 플래그 비트가 '1'일 때만 모드가 전송되므로 이 경우에만 디스크램블링 과정이 적용된다. 구체적인 디스크램블링 과정은 다음과 같다.

우선 특정한 키를 이용해 스크램블링 시 사용한 의사 랜덤 시퀀스와 동일한 의사 랜덤 시퀀스를 발생시킨다. 전송받은 모드 정보인 3비트 고정 길이 부호어와 앞서 발생시킨 의사 랜덤 시퀀스에서 순서대로 읽어 들인 3비트를 수식 (2)와 같이 XOR 연산을 통해 원래의 모드 정보를 복원한다.

$$Mode_{org} = Mode_{new} \oplus 3\,bit\ random\ sequence \quad (2)$$

XOR 연산의 특성상 의사 랜덤 시퀀스가 스크램블링 할 때 사용한 것과 동일하다면 본래의 모드 정보를 복원할 수 있다. 만약 이와 동일한 의사 랜덤 시퀀스를 알지 못한다면, 본래 모드를 복원할 수 없으므로 원 영상을 제대로 복원할 수 없게 된다.

2. 인트라 16x16 부호화

H.264 비디오 압축 기술의 인트라 16x16 부호화에는 총 4가지 종류의 예측 모드가 존재한다. 모드 0은 수직, 모드 1은 수평, 모드 2는 DC, 모드 3은 Plane 예측에 해당하며 16x16 블록 단위로 가능한 4가지 모드중 하나의 최종 모드를 선택한다. 최종 모드를 선택하는 과정은 인트라 4x4와 유사하게 일반적으로 에러 값을 사용한다. 모드 0부터 모드 3까지 각 모드에 따라 결정되는 예측값들을 이용하여 4가지 경우의 예측 에러 값을 계산한다. 그 중 가장 작은 에러 값을 갖는 모드가 인트라 16x16 예측 모드로 결정된다. 그 다음, 인트라 4x4와 마찬가지로 상기 결정된 예측 모드와, 현재 블록에서의 예측값을 뺀 잔여값을 각각 부호화하여 전송한다. 하지만 인트라 16x16은 인트라 4x4와 달리 예측 모드 정보(Intra16x16PredMode)를 표 1에서 볼 수 있듯이, 휘도 성분의 전송 블록 형태값(CodeBlockPatternLuma)과 색차 성분의 전송 블록 형태값(CodeBlockPatternChroma)과 함께 공동 부호화(Joint Coding)한다. 따라서 스크램블링을 위한 인트라 예측 모드 변경 과정에 의해 휘도 성분과 색차 성분의 전송 블록 형태값이 바뀐다면 일반적인 복호화기를 통해 영상을 복호화 할 수 없게 된다. 그러므로 인트라 16x16 부호화의 경우 휘도 성분과 색차 성분의 전송 블록 형태값이 바뀌지 않는 범위 내에서 모드 정보를 변경해야 한다. 또한 스크램블링으로 인한 비트량의 증가를 막기 위해 모드를 변경한 후의 코드 길이(Code Length)가 변경 전의 코드 길이와 같아야 한다. 따라서 예측 모드는 다르지만 코드 길이와 휘도 성분 그리고 색차 성분의 전송 블록 형태값이 같은 그룹을 만들어 해당되는 그룹 내에서 모드를 변경해야 한다. 예를 들어, 표 1에서 mb_type이 1인 경우와 mb_type이 2인 경우가 하나의 그룹이 될 수 있다. 두 가지 경우 모두 코드 길이가 3으로 동일하며 휘도 성분과 색차 성분의 전송 블록 형태 값이 0으로 동일하지만 예측 모드는 각각 0과 1로 서로 다른 값을 갖는다. 여기서 "my_type"은 표 1에서 나타나있는 것과 같이 공동 부호화 되는 예측 모드와 전송 블록 형태의 종류를 나타내는 값으로 1부터 24까지의 값을 가질 수 있다. 이와 같이 일반적인 복호화가 가능한 조건을 만족하는 그룹을 생성하여 해당 그룹 내에서 모드 정보를 변경하여야 한다. 표 1에서 볼 수 있듯이 이러한 조건을 만족하는 그룹은 mb_type이 {1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10, 11, 12}, {13, 14}, {15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24}인 경우 등이며, 그룹 내의 가능한 mb_type의

표 1. H.264의 인트라 16x16 예측 모드 부호화 테이블

Table 1. Intra 16x16 prediction mode coding table of H.264.

mb_type	Intra16x16 PredMode	CodedBlock PatternChroma	CodedAC PatternLuma	Code Length
1	0	0	0	3
2	1	0	0	3
3	2	0	0	5
4	3	0	0	5
5	0	1	0	5
6	1	1	0	5
7	2	1	0	7
8	3	1	0	7
9	0	2	0	7
10	1	2	0	7
11	2	2	0	7
12	3	2	0	7
13	0	0	1	7
14	1	0	1	7
15	2	0	1	9
16	3	0	1	9
17	0	1	1	9
18	1	1	1	9
19	2	1	1	9
20	3	1	1	9
21	0	2	1	9
22	1	2	1	9
23	2	2	1	9
24	3	2	1	9

수가 2인 경우와 4인 경우 두 가지로 나뉠 수 있다. 각 경우에 따른 구체적인 스크램브링 방법은 다음과 같다.

우선 부호화하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수를 확인한다. 만약 부호화하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 2인 경우, 의사 랜덤 시퀀스로부터 한 비트를 읽어, 읽어 들인 비트가 '0'일 경우에는 원래 모드 정보를 그대로 전송하고, '1'일 경우에는 원래 모드 정보의 LSB(Least Significant Bit)를 바꾸게 된다. 즉, 모드 0과 모드 1이 하나의 쌍을 이루고, 모드 2와 모드 3이 또 하나의 쌍을 이루어, 모드 0은 모드 1로, 모드 1은 모드 0으로, 모드 2는 모드 3으로, 모드 3은 모드 2로 변경되는 것이다. 이렇게 모드 0과 모드 1, 모드 2와 모드 3이 짝을 이루어 서로 바뀔 경우, 변경 가능한 그룹 내에서 짝을 이루는 모드로 변환하게 되므로 앞서 설명한 모든 조건을 만족시키게 된다. 예를 들어, 의사 랜덤 시퀀스에서 읽은 비트가 '1'이며 현재 블록의 mb-type이 2로 원래의 모드가 1로써 '01'의 비트 시퀀스를 갖는다면, 모드 정보의 마지막 비트인 '1'을 '0'으로 교환하여 '00'의 비트 시퀀스를 갖도록 하여 코드 길이와 전송 블록 형태값은 동일하지만 모드 0을 갖는 mb-type 1로 변경 시킨다. 또한 부호화 하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 4인 경우, 식 (3)과 같이 의사 랜덤 시퀀스로부터 두 비트를 읽어 예측 모드 정보의 표현 비트수인 2 비트와 XOR을 하여 모드 정보를 변경 한다.

$$Mode_{new} = Mode_{org} \oplus 2bit\ random\ sequence \qquad (3)$$

예를 들어, 의사 랜덤 시퀀스에서 읽은 비트가 '10'이며 부호화 하고자 하는 현재 블록의 mb_type이 18일 경우 원래 모드 정보가 1로 '01'의 비트 시퀀스를 가지므로, '10'과 '01'의 XOR연산 결과인 '11'이 새로운 모드 정보의 비트 시퀀스가 된다. 따라서 모드 정보가 3으로 변경되며 이에 따라 mb_type이 20이 된다. 이와 같은 방법으로 인트라 16x16 예측 모드 정보를 변경할 경우 앞서 서술한 모든 조건을 만족시키게 된다. 즉, 코드 길이, 휘도 성분의 전송 블록 형태 값, 색차 성분의 전송 블록 형태 값은 동일하지만 모드 정보만 다른 값으로 변경된다. 따라서 의사 랜덤 시퀀스에서 인트라 16x16 블록당 한 비트 또는 두 비트씩 읽어 들여, 읽은 비트와 현재 모드 정보와의 XOR 연산이나 LSB의 변경을 통해 본래 모드 정보를 다른 값으로 변경시킨다. 이와 같은 방법으로 모드 정보를 변경하면, 변경 후의 코드 길이가 동일하므로 스크램블링으로 인한 비트량의 증가가 전혀 없으며, 본래 모드 정보를 간단한 XOR 연산이나 LSB의 변경을 통해 스크램블링하게 되므로 계산량의 증가가 거의 없다. 뿐만 아니라 인트라 4x4 경우와 마찬가지로, 스크램블링 과정에서 사용된 의사 랜덤 시퀀스를 모르는 수신자의 경우 모드 변경에 대한 정보를 알 수 없으므로 원 영상을 복원할 수 없다.

인트라 16x16 블록의 디스크램블링 과정 역시 스크램블링 과정과 매우 유사하다. 구체적인 디스크램블링 과정은 다음과 같다.

우선 특정한 키를 이용해 스크램블링 과정에서 사용된 의사 랜덤 시퀀스와 동일한 의사 랜덤 시퀀스를 발생시킨다. 복호화 하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 2일 경우, 앞서 발생시킨 의사 랜덤 시퀀스에서 한 비트를 읽어 들인다. 만약 읽은 비트가 '1' 이라면 전송받은 모드 정보의 LSB를 변경하고, '0' 이라면 수신된 모드 정보를 그대로 사용한다. 스크램블링 과정에서 의사 랜덤 시퀀스로부터 읽은 한 비트가 '1'일 경우 LSB를 변경하여 전송하였으므로,

디스크램블링 과정에서 이를 다시 변경시킴으로서 원래 모드 정보를 복원하는 것이다. 또한 복호화 하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 4일 경우, 식 (4)와 같이 의사 랜덤 시퀀스에서 두 비트를 읽어 들인 후, 전송받은 모드 정보와 의사 랜덤 시퀀스로부터 읽어 들인 두 비트를 XOR연산하여 본래의 모드 정보를 얻는다.

$$Mode_{org} = Mode_{new} \oplus 2bit\ random\ sequence \quad (4)$$

스크램블링 과정에서 원래 모드 정보와 의사 랜덤 시퀀스에서 읽은 비트를 XOR 연산한 결과를 전송하였으므로, 디스크램블링 과정에서 전송 받은 모드 정보를 다시 동일한 의사 랜덤 시퀀트 비트와 XOR 연산을 통해 변경시킴으로써 원래 모드 정보를 복원하는 것이다. 따라서 스크램블링 과정에서 사용된 것과 동일한 의사 랜덤 시퀀스를 알지 못하는 수신자의 경우 본래의 인트라 16x16 블록의 모드를 알 수 없으므로 원 영상을 복원할 수 없다.

본 논문에서 제안한 인트라 예측 모드를 이용한 스크램블링은 스크램블링의 알고리듬을 알고 있는 임의의 공격자가 모드 변경을 통해 원래의 영상을 복원하려 한다 해도 복원 하기 힘들 것이다. 왜냐하면 각 블록마다 변경 가능한 모든 모드로 변경해본다 해도, 본래의 모드를 찾는 척도가 존재하지 않으며 이를 일일이 눈으로 확인 할 수 없기 때문이다. 또한 앞서 설명한 바와 같이 16x16 블록의 경우 4가지 모드가 존재하며 4x4 블록의 경우 9가지 모드가 존재 한다. 따라서 CIF(352 x 288) 영상의 경우 모든 블록이 인트라 16x16 일 경우 총 396개의 16x16 블록이 존재하며 총 1583번의 모드 변환 및 비교를 해 보아야 한다. 또한 모든 블록이 인트라 4x4 일 경우 총 6336개의 4x4 블록이 존재하며 총 57024번의 모드 변환 및 비교를 해 보아야 한다. 보든 프레마다 이러한 연산은 수행하여 비교한다는 것은 쉬운 일은 아닐 것이다. 따라서 본 논문에서 제안한 방법은 비교적 강인한 스크램블링 방법이라 할 수 있다.

(a) (b)

그림 3. 제안된 움직임 벡터를 이용한 스크램블링 방법을 적용한 백 번째 프레임: (a) 'foreman' 영상의 백 번째 프레임, (b) 'mother and daughter' 영상의 백 번째 프레임

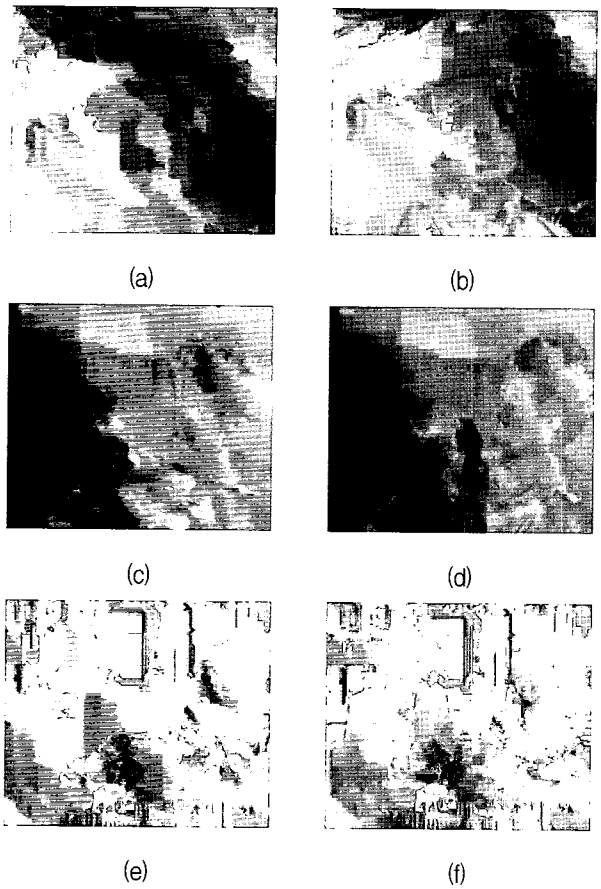
Fig. 3. The 100th decoded frames using the proposed motion vector scrambling method: (a) 100th frame of 'foreman' sequence, (b) 100th frame of 'mother and daughter' sequence.

III. 실 험

본 실험에서는 JM 8.1a 참조 소프트웨어를 사용하여 3가지 CIF 영상인 'foreman', 'paris', 'mother and daughter'영상에 대하여 각각의 스크램블링 알고리듬을 적용하였다. 'foreman'영상은 카메라의 움직임이 있는 영상으로 배경의 변화가 있으며 움직임이 비교적 많은 반면, 'paris'과 'mother and daughter'영상은 배경의 변화가 전혀 없으며 움직임 또한 비교적 적은 영상이다.

그림 3은 'foreman'영상과 'mother and daughter'영상에 본 논문에서 제안한 움직임 벡터를 이용한 인터 블록 스크램블링 방법을 적용한 100번째 프레임이다. 제안된 인터 블록 스크램블링 방법은 인트라 블록을 왜곡하지 못한다. 따라서 모든 블록이 인트라 부호화 되는 첫 번째 프레임의 경우 전혀 왜곡할 수 없다. 하지만 그림 4를 통해 알 수 있듯이 첫 번째 프레임 이후의 영상에 대해서는 매우 효과임을 확인 할 수 있다. 특히 'foreman' 영상의 경우 카메라의 움직임으로 인해 대부분의 인터 블록의 움직임 벡터 값이 어느 정도 큰 값을 갖게 되므로 움직임 벡터의 변경으로 인해 영상 전체가 심하게 왜곡된것을 확인할 수 있다. 하지만 'mother and daughter' 영상의 경우 배경이 전혀 변하지 않으므로 이전 영상의 값을 그대로 사용하게 된다. 따라서 그림 3 (b)를 통해 알 수 있듯이 영상의 배경이 전혀 왜곡되지 않음을 확인 수 있다. 뿐만 아니라, 주요 객체(mother, daughter)의 움직임 또한 매우 적기 때문에 'foreman'영상에 비해 비교적 덜 왜곡된 것을 확인 수 있다.

움직임 벡터를 이용한 스크램블링 방법의 경우, MPEG-1, 2, 4, H.264와 같은 대부분의 비디오 압축 기술에 적용할 수 있는 방법으로, 그림 3을 통해 알 수 있듯이 매우 효과적으로 영상을 왜곡할 수 있다. 뿐만 아니라 본 논문에서 제안된 방법은 연산량이 매우 적기 때문에 복잡도의 증가가 거의 없으며, 추정된 움직임 벡터가 아닌 차동 움직임 벡터를 교환하므로 스크램블링으로 인한 비트량의 증가가 전혀 없다. 그러나 모든



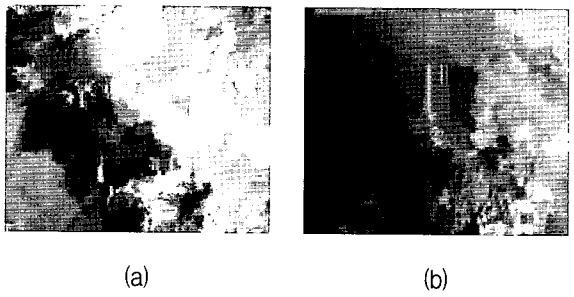
(a) (b) (c) (d) (e) (f)

그림 4. 제안된 H.264의 인트라 예측 모드 스크램블링 방법을 적용한 첫 번째 프레임과 백 번째 프레임: (a) 'foreman' 영상의 첫 번째 프레임, (b) 'foreman' 영상의 백 번째 프레임, (c) 'mother and daughter' 영상의 첫 번째 프레임, (d) 'mother and daughter' 영상의 백 번째 프레임, (e) 'paris' 영상의 첫 번째 프레임, (f) 'paris' 영상의 백 번째 프레임

Fig. 4. The first and 100th decoded frames using the proposed H.264 intra prediction mode scrambling method: (a) first frame of 'foreman' sequence, (b) 100th frame of 'foreman' sequence, (c) first frame of 'mother and daughter' sequence, (d) 100th frame of 'mother and daughter' sequence, (e) first frame of 'paris' sequence, (f) 100th frame of 'paris' sequence.

블록이 인트라 부호화되는 첫 번째 프레임과, 배경의 변화가 전혀 없거나 객체의 움직임이 매우 적은 영상의 경우 영상을 크게 왜곡할 수 없는 단점이 있다.

그림 4는 'foreman', 'paris', 'mother and daughter' 세 가지 영상에 대해 본 논문에서 제안한 H.264의 인트라 예측 모드를 이용한 인트라 블록 스크램블링을 적용한 결과이다. 인트라 블록의 왜곡이 인터 블록에 미치는 에러 전파 효과를 확인하기 위해 첫 번째 프레임과 100번째 프레임을 비교하였다. 본 실험에서는 첫 번째



(a) (b)

그림 5. 제안된 두 가지 스크램블링 방법을 모두 적용한 백 번째 프레임: (a) 'foreman' 영상의 백 번째 프레임, (b) 'mother and daughter' 영상의 백 번째 프레임

Fig. 5. The 100th decoded frames using the two proposed scrambling methods: (a) 100th frame of 'foreman' sequence, (b) 100th frame of 'mother and daughter' sequence.

프레임만을 인트라 프레임으로 부호화하고 나머지 프레임은 인터 프레임으로 부호화하였다. 즉, 모든 영상의 첫 번째 프레임은 모두 인트라 블록으로 이루어져 있으므로, 그림 4 (a), (c), (e)에서 볼 수 있듯이 영상 전체가 인트라 예측 모드의 변경으로 인해 원래 영상을 알아 볼 수 없을 정도로 왜곡된다. 그러나 두 번째 프레임부터는 인터 프레임이므로 프레임 내에 발생 할 수 있는 몇 개의 인트라 블록만이 예측 모드 정보의 변경으로 인해 왜곡된다. 따라서 첫 번째 프레임 이후의 프레임 내에서는 제안된 인트라 예측 모드 변경에 의한 직접적인 왜곡은 많지 않다. 그러나 그림 4 (b), (d), (f)에서 볼 수 있듯이 첫 번째 프레임의 왜곡이 전파되는 효과로 인해, 영상 전체가 직접적으로 스크램블링된 것과 같은 유사한 효과를 갖는다. 뿐만 아니라 카메라의 움직임으로 있는 'foreman'영상과, 배경의 변화가 전혀 없으며 주요 객체의 움직임이 적은 'paris'와 'mother and daughter'영상의 경우 모두 원래 영상을 알아 볼 수 없을 만큼 왜곡되었음을 확인할 수 있다.

그림 5는 'foreman'영상과 'mother and daughter'영상에 본 논문에서 제안한 움직임 벡터를 이용한 스크램블링 방법과 인트라 예측 모드를 이용한 스크램블링 방법을 모두 적용한 결과로 각 영상의 100번째 프레임이다. 그림 5에서 알 수 있듯이 영상의 인터 블록과 인트라 블록이 모두 왜곡 되므로 한 가지 방법만을 적용한 경우보다 더욱 심하게 영상이 왜곡됨을 확인 할 수 있다. 첫 번째 프레임에서 인트라 예측 모드를 이용한 방법에 의해 모든 블록이 왜곡되며, 첫 번째 프레임에서의 왜곡이 에러 전파 효과에 의해 모든 프레임에 영향을 미치게 된다. 뿐만 아니라 움직임 벡터를 이용한 인

터 블록의 스크램블링으로 인해 영상이 더욱 심하게 왜곡되는 것이다.

H.264의 인트라 예측 모드를 이용한 스크램블링 방법의 경우, H.264의 인트라 부호화 특성을 이용한 방법으로, H.264와 같이 인트라 예측 모드를 사용하여 부호화하는 비디오 압축 기술에 적용 가능하다. H.264의 인트라 부호화는 기본 블록 단위에 따라 인트라 4x4 부호화와 인트라 16x16 부호화로 나뉜다. 제안된 알고리듬은 각 인트라 부호화에 따라, 스크램블링으로 인해 비트량의 변화가 발생하지 않으며 통상적인 복호화가 가능한 범위 내에서 인트라 예측 모드를 변경한다. 이 방법은 인트라 블록을 위한 스크램블링이므로 인터 블록을 직접적으로 왜곡할 수 없다. 하지만 그림 4를 통해 알 수 있듯이, 에러 전파의 효과로 인해 인터 블록에 대한 직접적인 왜곡 없이 인트라 블록을 왜곡한 만큼의 유사한 효과를 낼 수 있다. 따라서 이 방법은 인트라 블록의 왜곡만으로 인터 블록을 포함한 비디오 영상 전체를 왜곡할 수 있는 장점을 가지고 있다. 뿐만 아니라, 인트라 예측 모드 변경 시 비트량이 증가하지 않는 범위 내에서 변경하므로 압축 효율에 전혀 영향을 미치지 않는다. 또한 XOR 연산이나 LSB의 변환과 같은 단순한 연산을 통해 예측 모드를 변경하므로 구현이 용이하며 계산량의 증가가 거의 없다.

Ⅳ. 결 론

본 논문에서는 차동 움직임 벡터의 수평 성분과 수직 성분을 교환하여 영상을 왜곡하는 인터 블록 스크램블링 방법과, H.264의 인트라 부호화 과정에서 생성되는 인트라 예측 모드를 변경함으로써 영상을 왜곡하는 인트라 블록 스크램블링 방법을 제안하였다.

본 논문에서 제안된 두 가지 새로운 디지털 비디오 스크램블링 방법은 모두 기존 방법과 달리 스크램블링으로 인한 비트량의 증가가 없으며 간단한 연산으로 구현 가능하다는 장점을 가지고 있다. 또한 두 가지 스크램블링 방법을 각각 따로 사용하여도 효과적이지만, 두 가지 방법을 동시에 사용할 경우 그림 5를 통해 알 수 있듯이 더욱 효과적으로 영상을 왜곡 할 수 있다.

참 고 문 헌

[1] W. Zeng and S. Lei, "*Efficient frequency domain selective scrambling of digital video*," IEEE Transactions on Multimedia, vol. 5, pp.118-129, March 2003.

[2] L. Tang, "*Methods for Encrypting and Decrypting MPEG Video Data Efficiently*," Proc. of the fourth ACM International Conference on Multimedia, pp.219-229, Boston, Nov. 1996.

[3] W. Zeng and S. Lei, "*Efficient frequency domain video scrambling for content access control*," Proc. of the seventh ACM International Conference on Multimedia, pp.285-294, Orlando, Nov. 1999.

[4] N. Katta et al., "*Scrambling apparatus and descramble apparatus*," U.S patent 5377266, Dec. 27, 1994.

[5] J. Jang, "*Digital video scrambling method*," KR patent 0151199, Jun. 18, 1998.

[6] M. Park, "*Estimation and Watermarking of Motion Parameters in Model Based Image Coding*," Proc. of IEEK Conference, pp.1264-1267, Dec. 2002.

저 자 소 개

안 진 행(학생회원)

2003년 성균관대학교 정보통신 공학과 학사.

2005년 성균관대학교 전자전기 공학과 석사.

<주관심분야 : 워터마킹, 영상압축, 신호처리>

전 병 우(정회원)

1985년 서울대학교 전자공학과 학사 졸업.

1987년 서울대학교 전자공학과 석사 졸업.

1992년 Purdue Univ, School of Elec. 박사 졸업.

1993년~1997년 8월 삼성전자 신호처리연구소 수석 연구원.

1997년 9월~현재 성균관대학교 전자전기공학부 부교수.

<주관심분야 : 멀티미디어, 영상압축, 영상인식, 신호처리>