논문 2005-42SP-2-7

H.264 인트라 예측 모드를 이용한 디지털 비디오 스크램블링 방법

(Digital Video Scrambling Method using Intra Prediction Mode of H.264)

안 진 행*, 전 병 우**

(Jinhaeng Ahn and Byeungwoo Jeon)

요 약

최근 아날로그 데이터가 디지털화 되면서 디지털 멀티미디어의 사용이 급증하고 있다. 그러나 이러한 디지털 멀티미디어 컨텐츠는 인터넷과 같은 누구나 접근이 용이한 채널을 통해 불법 복제 및 유통이 자유롭다는 단점을 가지고 있다. 따라서 디지털 멀티미디어에 대한 보안의 필요성이 매우 커지고 있다. 이에 따라, 최근 몇 년 동안 디지털 컨텐츠 보호 기술에 대한 여러 가지 연구가 이루어져 왔으며, 비디오 스크램블링은 이러한 기술 중 하나로써 현재 아날로그 방송에서도 사용되고 있다. 본 논문에서는 차세대 부호화 기술인 H.264의 인트라 예측 모드를 이용한 간단하고 효율적인 디지털 비디오 스크램블링 방법을 제안한다. H.264는 다양한 인트라 예측 모드를 사용하므로, 인트라 예측 모드의 간단한 변경만으로 효율적으로 스크램블링 할 수 있다. 뿐만 아니라 제안된 알고리듬은 스크램블링 후 비트양이 전혀 증가하지 않으며, 간단한 인트라 블록의 스크램블링 만으로 인터 블록까지 왜곡시킬 수 있는 장점을 가지고 있다. 본 논문에서는 새로운 디지털 비디오 스크램블링 방법을 제안하고, 이에 대한 실험 결과를 통해 제안된 알고리듬의 효율성을 보인다.

Abstract

The amount of digitalized contents has been rapidly increased, but the main distribution channel of them is Internet which is easily accessible. Therefore 'security' necessarily arises as one of the most important issues and the method of protecting contents becomes a major research topic as much as data coding techniques. In recent years, many developers have studied on techniques that allow only authorized person to access contents. Among them, the scrambling method is one of well-known security techniques. In this paper, we propose a simple and effective digital video scrambling method which utilizes the intra block properties of a recent video coding technique, H.264. Since intra prediction modes are adopted in H.264 standard, it is easy to scramble a video sequence with modification of the intra prediction modes. In addition to its simplicity, the proposed method does not increase bit rate after scrambling. The inter blocks are also distorted by scrambling intra blocks only. This paper introduces a new digital video scrambling method and verifies its effectiveness through simulation.

Keywords : Scrambling, Intra prediction mode, Variable length coding, Fixed length coding

I. 서 론

디지털 화의 추세에 발맞추어 디지털 멀티미디어의 사용이 급증하고 있다. 뿐만 아니라 기존의 아날로그 데이터들이 디지털화 되면서 각종 멀티미디어 매체 또한 기존의 아날로그 방식에서 디지털 방식으로 전환되어 가고 있다. 그러나 대부분의 디지털 데이터는 불법 복제 및 유포가 매우 용이하다는 단점을 가지고 있다. 특히 대부분의 디지털 비디오의 경우, 위성이나 케이블, 인터넷 망과 같은 공개적인 경로를 통해 전송되므로 불

* 학생회원, ** 정회원, 성균관대학교 전자전기공학과
(School of Electronic Electrical Engineering, Sungkyunkwan University)
접수일자: 2004년10월5일, 수정완료일: 2004년12월10일

법 복제 및 무단 서비스 이용의 위험성이 더욱 크다. 따라서 디지털 멀티미디어 데이터에 대한 보호의 중요성이 커지고 있으며, 이에 따라 여러 가지 디지털 비디오 보호 기술에 대한 연구가 이루어지고 있다. 스크램블링은 이와 같은 디지털 비디오 보호 기술 중 하나로써 원래의 영상 데이터를 특정한 키(key)에 의해 변형 또는 암호화하여 전송함으로써, 특정한 키를 가진 수신자만이 정상적으로 영상을 복원할 수 있도록 하는 기술이다. 따라서 원 영상을 복원할 수 있는 키를 가진 인증된 가입자들만이 스크램블링으로 인해 왜곡된 영상을 정상적으로 복구할 수 있고, 허가되지 않은 수신자의 경우 수신된 영상을 복호화 하더라도, 원 영상이 아닌 스크램블링 과정으로 인해 왜곡된 영상을 보게 됨으로써 정당한 수신자의 권리를 보호할 수 있다.

최근 몇 년 동안 이와 같은 스크램블링 기술에 대한 연구가 이루어져왔다[1]-[5]. 기존의 제안된 스크램블링 방법 중 움직임 벡터를 이용한 스크램블링 방법[5]은 전송 블록 형태 값(Coded Block Pattern)과 움직임 벡터 값을 이용한 방법으로 다음과 같다. 각 매크로블록의 전송 블록 형태 값을 구한다음, 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 떨어져있는 부호를 이용하여 부호화한다. 그러나 이와 같은 방법은 전송 블록 형태 값을 모듈러 연산한 결과 만큼 이동한 후의 코드 길이가 기존의 코드 길이 보다 늘어 날 수 있으므로 스크램블링 후 비트의 양이 증가 할 수 있다는 단점이 있다.

혹은 DES나 AES와 같은 암호화 알고리듬을 사용하여 스크램블링 하는 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, 영상 신호 전부를 암호화 할 경우 복잡도가 매우 크게 증가하게 된다. 이를 보완하기 위해 영상의 중요도에 따라 크게 네 가지 레벨로 나누어 선택적으로 스크램블링 하기도 한다. 이를 위해 매우 중요한 영상의 경우 모든 영상 신호를 암호화하고, 이보다 한 단계 낮은 중요도를 갖는 영상의 경우에는 움직임 벡터 또는 DCT 변환 계수 등과 같은 중요한 파라미터 값 들을 암호화 한다. 그 다음 단계의 중요도를 갖는 영상의 경우에는 인트라 프레임 또는 인트라 블록만을 암호화 하며, 가장 낮은 중요도를 갖는 영상의 경우에는 모든 헤더 정보를 암호화 하여 전송하게 된다. 그러나 아무리 적은 양의 영상 신호를 암호화 한다 하더라도 암호화 알고리듬 자체의 계산량이 매우 크며, 앞서 설명한 움직임 벡터를 사용한 스크램블링 방법과 마찬가지로 스크램블링 후 비트의 양이 증가하게 된다. 뿐만 아니라 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 존재한다[1]. 웨이블릿 기반의 스크램블링 방법은 웨이블릿 변환 계수들을 블록 별로 섞는 것이다. 웨이블릿 변환으로 인해 생성된 부대역(Subband)내에서 블록을 나누어, 특정한 테이블을 기준으로 블록내의 계수값들을 섞어 영상에 왜곡을 가한다. 또는 비트 플레인(Bit Plane)에서 중요한 비트(Significant Bit)나 부호 비트를 선택적으로 섞는 방법이 있다. 하지만 MPEG-2나 H.264와 같은 대부분의 비디오 영상 압축 알고리듬에는 웨이블릿 변환이 포함되지 않으므로 일반적인 비디오 영상 신호에 적용하는 데는 어려움이 있다. 주파수 영역에서의 또 다른 스크램블링 방법인 DCT 기반의 스크램블링 방법은 웨이블릿 기반의 스크램블링 방법과 마찬가지로 변환 계수들을 섞어 영상을 왜곡 시킨다. 예를 들어, 한 프레임내의 DC 계수값만을 모아 특정한 테이블을 통해 DC 계수값들을 섞거나, 동일 주파수 위치의 계수값들을 모아 웨이블릿 변환에서의 부대역 개념과 유사한 주파수 층을 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 스크램블링 방법을 제안한다. 제안된 알고리듬은 H.264 비디오 압축 기술의 인트라 예측 모드를 이용한 방식으로, 단순히 인트라 예측 모드만을 변경함으로써 효율적으로 원 영상을 왜곡시킨다. H.264 비디오 압축 기술은 압축 효율을 높이기 위해 기존의 MPEG-1,2,

H.263과 달리 인터 블록 뿐만 아니라 인트라 블록도 예측 부호화 한다. 따라서 H.264 비디오 압축 기술은 인트라 블록을 예측 부호화하기 위해 인트라 예측 모드를 사용하며, 부호화 시 사용된 각 인트라 블록의 예측 모드를 전송한다. 본 논문은 이와 같이 전송되는 인트라 예측 모드를 이용한 방법으로, MPEG-1,2, H.263과 같은 비디오 압축 기술과 달리 인트라 예측 부호화 기술을 사용하는 H.264 비디오 압축 기술에 적합한 방법이다. 이와 같은 H.264 비디오 압축 기술의 인트라 예측 부호화에는 인트라 16x16 부호화와 인트라 4x4 부호화 방법이 존재하며, 각 방법에 따른 구체적인 스크램블링 과정은 2장에서 서술한다. 3장에서는 제안된 알고리듬을 사용한 실험 결과를 설명하고, 끝으로 4장에서는 본 방식에 대한 결론을 내린다.

II. H.264 인트라 예측 모드를 이용한 비디오 스크램블링

H.264 비디오 압축 기술에서는 인트라 프레임 부호화 시, 공간방향의 중복성을 없애기 위해 인트라 블록의 예측값을 추정한다. 인트라 예측 방향을 결정하는 것은 정해진 방법이 있는 것이 아니라, 부호화기에 따라 최적의 방법을 나름대로 선택할 수 있으나 통상적으로 정해진 방향(모드)별로 현재 블록의 화소값들과 예측값과의 SAD(Sum of Absolute Difference) 값을 계산하여 가장 작은 SAD 값을 갖는 예측값의 방향을 인트라 예측 모드로 한다. 인트라 예측 모드가 정해지면, 결정된 모드에 해당하는 방향의 예측값과 현재 블록의 화소값들과의 감산 연산을 통해 잔여값(Residual)을 구한다. 이렇게 생성된 잔여값과 예측 모드를 가변 길이 부호화 하여 수신측에 전송하게 된다. 수신측에서는 전송받은 모드를 통해 예측값을 구한 후, 수신된 잔여값과 앞서 구한 예측값을 서로 더하여 원 영상을 복원 할 수 있다. 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 변경함으로써 원래의 모드를 알지 못하는 수신자의 경우 원 영상을 복원 할 수 없도록 한다. 앞서 설명한 바와 같이 H.264 비디오 압축 기술은 인트라 4x4 부호화와 인트라 16x16 부호화 방법이 존재하며 각 방법에 따른 스크램블링 알고리듬은 다음과 같다.

1. 인트라 4x4 부호화의 스크램블링

인트라 4x4 예측 모드는 그림 1과 같이 8가지 방향의 모드와 DC 모드로 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 1의 경우 수평모드에 해당하며, a, b, c, d 화소의 예측값으로 I 화소값을, e, f, g, h 화소의 예측값으로는 J 화소값을, i, j, k, l 화소의 예측값으로는 K 화소값을, m, n, o, p 화소의 예측값으로는 L 화소값을 사용한다. 이와 유사한 방식으로 그림 1에 나타난 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.

하나의 모드를 정하기 위해 모드 0부터 모드 8까지 각 모드에 따라 결정되는 예측값들을 이용하여 9가지

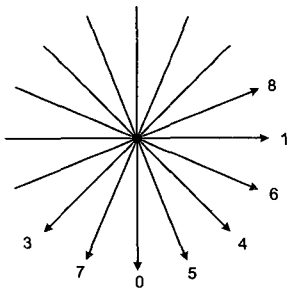

그림 1. H.264 부호화의 인트라 4x4 예측 모드 방향
Fig. 1. Intra 4x4 prediction mode directions of H.264.

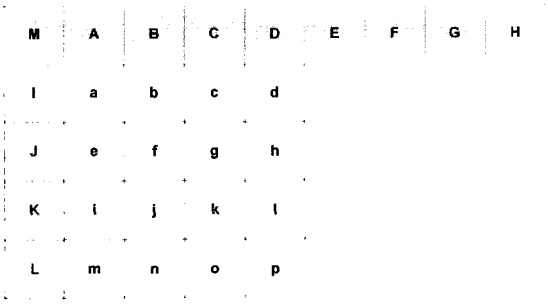

그림 2. 인트라 4x4 예측에서 사용되는 화소와 주변 화소
Fig. 2. Pixels used in the intra 4x4 prediction and its neighbor pixels.

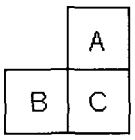

그림 3. C블록의 예측 모드 부호화 시 사용되는 주변 블록 A, B
Fig. 3. Neighboring block A and B used for encoding prediction mode of block C.

경우의 예측 에러값인 SAD 값을 계산 한다. 그 중 가장 작은 SAD 값을 갖는 모드가 인트라 4x4 예측 모드로 결정된다. 이렇게 결정된 예측 모드와, 현재 블록에서 예측값을 뺀 잔여값을 각각 부호화 하여 전송한다. 이때, 예측 모드는 고정 길이 부호화(Fixed Length Coding) 하며, 잔여값은 가변 길이 부호화 하여 전송한다. 이 때 발생 가능한 예측 모드가 총 9 가지이므로 예측 모드를 고정 길이 부호화 하기 위해서는 총 4 비트가 필요하다. 하지만 인트라 4x4 부호화의 기본 블록 단위인 4x4 블록 당 하나의 예측 모드를 전송해야 하므로 이는 비교적 많은 비트량을 필요로 하게 된다. 뿐만 아니라, 만약 발생 가능한 예측 모드가 8 가지라면 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 부호화 효율을 높인다. 플래그 비트는 주변 블록의 모드와의 관계를 나타내는 비트로, 그림 3에서와 같이 현재 블록인 C의 모드 정보를 부호화하기 위해서 좌측 블록인 B와 상위 블록인 A의 모드 정보를 이용한다.

주변 블록 A와 B의 예측 모드 가운데 값이 적은 것을 '최우선모드(Most Probable Mode)'로 결정한다. 최우선모드와 현재 블록 C의 예측 모드를 비교하여, 두 모드가 서로 같으면 앞서 서술한 플래그 비트를 '1'로 하여 전송하고 같지 않으면 '0'으로 하여 전송한 다음, 추가로 나머지 8가지 경우를 가리키는 3비트의 부호어를 고정 길이 부호화하여 전송한다. 따라서 만약 최우선모드와 현재 블록의 예측 모드가 같은 경우 플래그 비트를 통해 1 비트만으로 전송하고자 하는 모드를 표현할 수 있으며, 최우선모드와 현재 블록의 예측 모드가 다른 경우 플래그 비트와 최우선모드를 제외한 나머지 8가지 모드를 표현 할 수 있는 3 비트를 사용해 원하는 모드 정보를 전송할 수 있다. 단, 최우선모드와 현재 블록의 예측 모드가 일치하지 않아 추가로 전송될 3

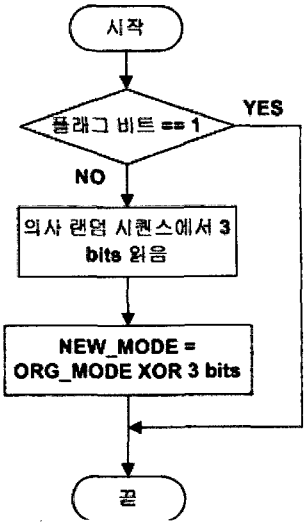

그림 4. 인트라 4x4 부호화의 스크램블링 흐름도
Fig. 4. Scrambling flowchart of intra 4x4 coding.

비트 고정길이 부호어는 최우선모드와 비교하여 현재 모드가 같거나 클 경우, 현재 모드에 1을 뺀 값을 부호화한다. 예를 들어, 주변 블록 A의 예측 모드가 3이고 B의 예측 모드가 4이며 현재 블록 C의 예측 모드는 4라고 가정하면, 최우선모드는 3과 4의 최소값인, 3이 되어 현재 블록의 예측 모드와 같지 않게 된다. 이 경우, 플래그 비트를 '0'으로 하여 보내게 되고 추가로 현재 블록의 예측 모드 4가 최우선모드 3보다 크므로 4에서 1을 뺀 3을 고정 길이 부호화하여 '011'을 전송하게 되는 것이다.

제안하고자 하는 스크램블링 알고리듬은 위와 같은 과정을 거쳐 최종적으로 전송하게 되는 3비트의 고정길이 부호어를 변경함으로써 왜곡을 가한다. 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 같으며 이에 대한 스크램블링 흐름도는 그림 4에 나타나있다.

우선 플래그 비트를 사용해서 예측 모드의 전송 여부를 확인한다. 만약 플래그 비트가 '1'이라면 예측 모드를 전송하지 않고 플래그 비트만 전송하는 경우이므로 예측 모드를 변경할 수 없다. 따라서 예측 모드가 전송되는 플래그 비트가 '0'인 경우에만 예측 모드 변경이 가능하다. 예측 모드 변경을 위해 우선 특정한 키에 의해 의사 랜덤 시퀀스(Pseudo Random Sequence)를 발생시킨다. 그 다음, 전송할 모드 정보인 3비트 고정길이

표 1. H.264의 인트라 16x16 예측 모드 부호화 테이블
Table 1. Intra 16x16 prediction mode coding table of H.264.

mb_type	Name of mb_type	mb_part_pred_mode	Intra16x16PredMode	CodedBlockPatternChroma	CodedACPatternLuma	Code Length
1	I_16x16_0_0_0	Intra_16x16	0	0	0	3
2	I_16x16_1_0_0	Intra_16x16	1	0	0	3
3	I_16x16_2_0_0	Intra_16x16	2	0	0	5
4	I_16x16_3_0_0	Intra_16x16	3	0	0	5
5	I_16x16_0_1_0	Intra_16x16	0	1	0	5
6	I_16x16_1_1_0	Intra_16x16	1	1	0	5
7	I_16x16_2_1_0	Intra_16x16	2	1	0	7
8	I_16x16_3_1_0	Intra_16x16	3	1	0	7
9	I_16x16_0_2_0	Intra_16x16	0	2	0	7
10	I_16x16_1_2_0	Intra_16x16	1	2	0	7
11	I_16x16_2_2_0	Intra_16x16	2	2	0	7
12	I_16x16_3_2_0	Intra_16x16	3	2	0	7
13	I_16x16_0_0_1	Intra_16x16	0	0	1	7
14	I_16x16_1_0_1	Intra_16x16	1	0	1	7
15	I_16x16_2_0_1	Intra_16x16	2	0	1	9
16	I_16x16_3_0_1	Intra_16x16	3	0	1	9
17	I_16x16_0_1_1	Intra_16x16	0	1	1	9
18	I_16x16_1_1_1	Intra_16x16	1	1	1	9
19	I_16x16_2_1_1	Intra_16x16	2	1	1	9
20	I_16x16_3_1_1	Intra_16x16	3	1	1	9
21	I_16x16_0_2_1	Intra_16x16	0	2	1	9
22	I_16x16_1_2_1	Intra_16x16	1	2	1	9
23	I_16x16_2_2_1	Intra_16x16	2	2	1	9
24	I_16x16_3_2_1	Intra_16x16	3	2	1	9

부호어와 앞서 발생시킨 의사 랜덤 시퀀스에서 순서대로 읽어들인 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 '011'이고 의사 랜덤 시퀀스에서 읽어 들인 3비트가 '101'이라면 새로운 모드 정보는 '110'이 된다. 따라서 새로운 모드 또한 3비트로 0에서 7까지의 값을 갖게 되며, 일반적인 복호화 과정에는 전혀 지장을 주지 않게 된다. 따라서 원래의 예측 모드를 모르는 수신자의 경우 일반적인 복호화는 가능하지만, 잘못된 예측값으로 인해 왜곡된 영상을 보게 된다. 또한 XOR연산의 특성에 의해 동일한 의사 랜덤 시퀀스를 가진 사용자만이 원래 모드를 복원할 수 있다. 따라서 스크램블링 과정에서 사용된 의사 랜덤 시퀀스와 동일한 의사 랜덤 시퀀스를 발생시킬 수 있는 특정한 키를 가진 허가된 사용자만이 원래의 예측 모드를 얻어 원 영상을 복원할 수 있게 된다. 그리고 만약 발생 시킨 의사 랜덤 시퀀스를 모두 사용했으면 다시 첫 번째 값부터 순환시켜 사용한다. 이와 같은 과정을 통한 스크램블링은 변경된 모드 정보의 비트 길이가 변경 전의 비트 길이와 같기 때문에 비트량의 증가가 전혀 없다. 또한 스크램블링 시 사용되는 XOR 연산은 구현이 쉬우며 계산량이 적기 때문에, 이로 인한 복잡도의 증가가 매우 적다.

2. 인트라 16x16 부호화의 스크램블링

인트라 16x16 부호화를 위해 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 SAD를 이용한다. 모드 0부터 모드 3까지 각 모드에 따라 결정되는 예측값들을 이용하여 4가지 경우의 예측 에러값인 SAD 값을 계산한다. 그 중 가장 작은 SAD 값을 갖는 모드가 인트라 16x16 예측

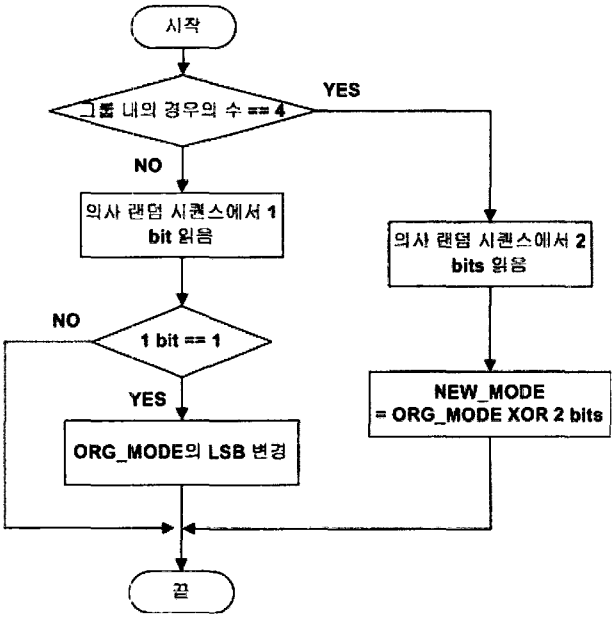

그림 5. 인트라 16x16 부호화의 스크램블링 흐름도
Fig. 5. Scrambling flowchart of intra 16x16 coding.

모드로 결정된다. 이렇게 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 같이 공동 부호화(Joint Coding)한다. 따라서 인트라 4x4의 경우와 같이 원래의 모드 정보를 모드 0부터 모드 3까지 중 랜덤하게 변경 할 경우 공동 부호화되는 휘도 성분과 색차 성분의 전송 블록 형태값이 바뀔 수 있어 일반적인 복호화기를 통해 영상을 복호화 할 수 없는 경우가 발생한다. 따라서 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 길이가 변경 전의 코드 길이와 같아야 한다. 따라서 표 1에서 볼 수 있듯이 예측 모드는 다르지만 코드 길이(Code Length)와 휘도 성분 그리고 색차 성분의 전송 블록 형태값이 같은 그룹을 만들어 해당되는 그룹 내에서 모드를 변경해야 한다. 예를 들어, 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. 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 수가 2인 경우와 4인 경우 두 가지로 나뉠 수 있다. 각 경우에 따른 구체적인 스크램브링 방법은 다음과 같으며 이에 대한 스크램블링 흐름도는 그림 5에 나타나있다.

우선 부호화하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 몇 개인지 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 변경 되는 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 원래의 모드가 1로 '01'의 비트 시퀀스를 갖는다면, 모드 정보의 마지막 비트인 '1'을 '0'으로 교환하여 '00'의 비트 시퀀스를 갖도록 하여 모드 0으로 변경 시킨다. 또한 부호화 하고자 하는 현재 블록의 mb_type이 속한 그룹 내의 경우의 수가 4인 경우, 의사 랜덤 시퀀스로부터 두 비트를 읽어 예측 모드 정보의 표현 비트수인 2 비트와 XOR을 하여 모드 정보를 변경 한다. 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 블록 당 한 비트

또는 두 비트씩 읽어 들여, 읽은 비트와 현재 모드 정보와의 XOR 연산이나 LSB의 변경을 통해 본래 모드 정보를 다른 값으로 변경시킨다. 그리고 인트라 4x4 부호화 과정에서의 스크램블링 방법과 마찬가지로, 발생시킨 의사 랜덤 시퀀스를 모두 사용했으면 다시 첫 번째 값부터 순환시켜 사용한다. 이와 같은 방법으로 모드 정보를 바꾸게 되면, 변경 후의 코드 길이가 동일하므로 스크램블링으로 인한 비트량의 증가가 전혀 없다. 또한 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 어떠한 모드로 변경되었는지에 대한 정보를 알 수 없으므로 원 영상을 복원할 수 없다.

3. 인트라 4x4 부호화의 디스크램블링

인트라 4x4 블록의 디스크램블링 과정은 스크램블링 과정과 매우 유사하다. 복호화 하고자 하는 인트라 4x4 블록 플래그 비트가 '1'일 때 변경된 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 고정길이 부호어와 앞서 발생시킨 의사 랜덤 시퀀스에서 순서대로 읽어들인 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.

4. 인트라 16x16 부호화의 디스크램블링

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, 의사 랜덤 시퀀스에서 두 비트를 읽어 들인 후, 전송받은 모드 정보와 의사 랜덤 시퀀스로부터 읽어 들인 두 비트를 XOR연산하여 새로운 모드 정보를 얻는다. 이렇게 얻은 모드 정보가 본래의 모드 정보가 된다. 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 만약 스크램블링 과정에서 사용된 것과 동일한 의사 랜덤 시퀀스를 알지 못하는 수신자의 경우 어떠한 인트라 16x16 블록의 모드가 변경되었는지 알 수 없으므로 원 영상을 복원할 수 없다.

Ⅲ. 실 험

본 실험에서는 JM(Joint Model) 8.1a의 참조 소프트웨어와 3가지 CIF 영상인 'paris', 'mother and daughter', 'foreman'영상을 사용하였다. 그림 6은 위 실험 조건을 사용하여 본 논문에서 제안한 스크램블링 알고리듬을 적용한 결과이다. 인트라 블록의 왜곡으로 인터 블록에 미치는 에러 전파 효과를 확인하기 위해 첫 번째 프레임과 100번째 프레임을 비교하였다. 본 실험에서는 첫 번째 프레임만을 인트라 프레임으로 부호화하고 나머지 프레임은 인터 프레임으로 부호화하였다. 즉, 모드 영상의 첫 번째 프레임은 인트라 프레임으로 영상 전체가 인트라 블록으로 이루어져 있다. 따라서 영상 전체가 본 논문에서 제안된 인트라 예측 모드 정보의 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 왜곡된

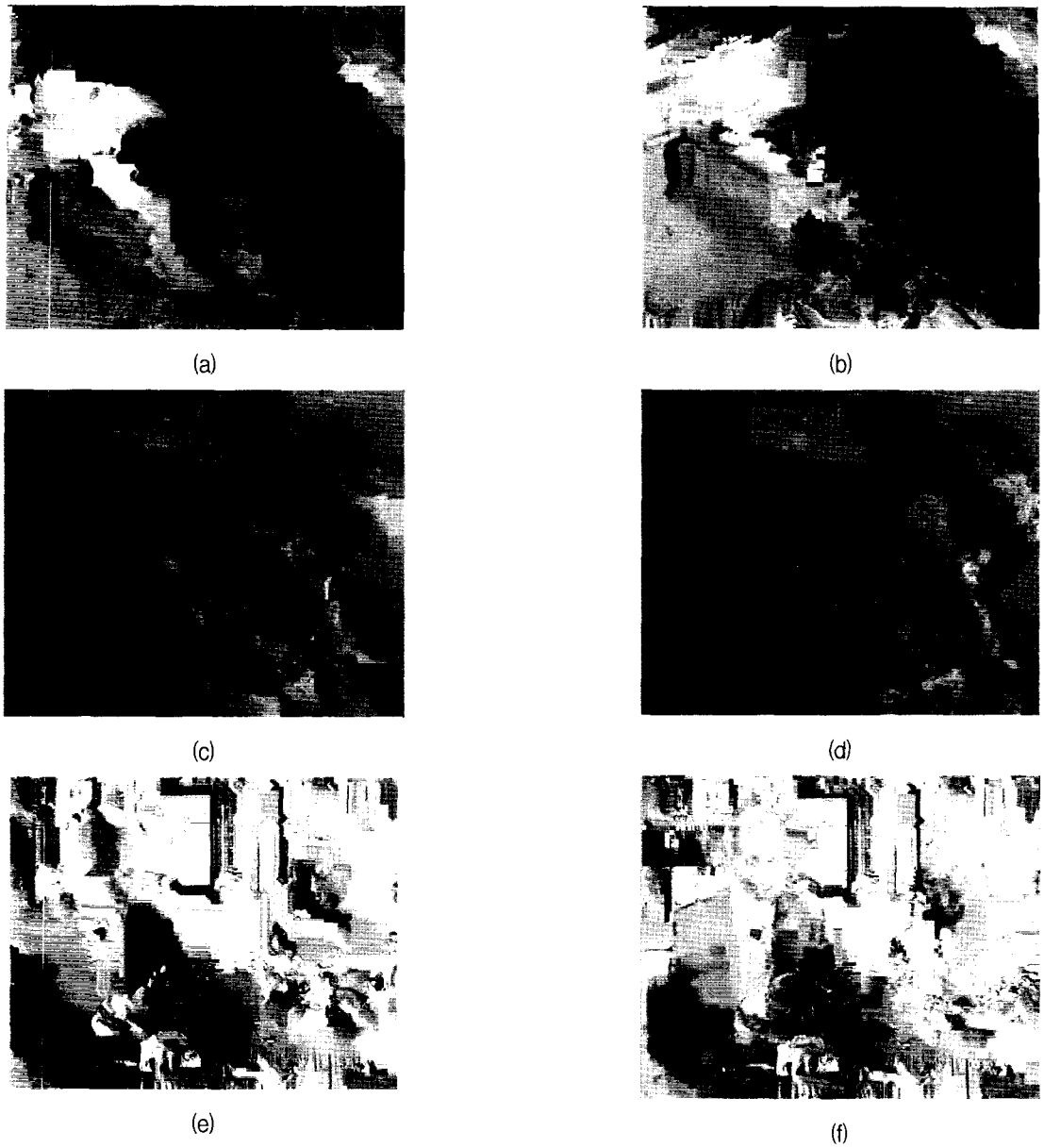

그림 6. 제안된 알고리듬을 사용하여 스크램블링한 첫 번째 프레임과 백 번째 프레임: (a) 'foreman' 영상의 첫 번째 프레임, (b) 'foreman' 영상의 백 번째 프레임, (c) 'mother and daughter' 영상의 첫 번째 프레임, (d) 'mother and daughter' 영상의 백 번째 프레임, (e) 'paris' 영상의 첫 번째 프레임, (f) 'paris' 영상의 백 번째 프레임

Fig. 6. The first and 100th frames using proposed 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.

다. 따라서 프레임 내에 직접적인 왜곡은 많지 않다. 그러나 그림 6 (b), (d), (f)에서 볼 수 있듯이 첫 번째 프레임의 왜곡이 전파되는 효과로 인해, 영상 전체가 스크램블링 된 것과 같은 유사한 효과를 갖는다.

'Foreman' 영상의 경우 카메라의 움직임이 있는 반면 'paris'와 'mother and daughter' 영상의 경우 카메라의 움직임이나 객체의 움직임이 거의 없으므로 배경과 같은 정적인 부분은 대부분 이전 프레임의 값을 그대로 사용하게 된다. 따라서 'paris'나 'mother and daughter'의 경우 왜곡된 첫 프레임의 배경이 그대로 100번째 프레임까지 전파되며 'foreman'영상의 경우에도 에러 전파 효과로 인해 왜곡이 100번째 프레임까지 전해지는 것을 볼 수 있다. 즉, 실험 결과에서 알 수 있듯이 인트라 블록의 스크램블링 만으로 인터 블록의 스크램블링

효과를 얻을 수 있다.

Ⅳ. 결 론

본 논문에서는 H.264의 인트라 부호화 과정에서 생성되는 인트라 예측 모드를 변경함으로써 영상을 스크램블링하는 알고리듬을 제안하였다. H.264의 인트라 부호화는 부호화되는 기본 블록 단위에 따라 인트라 4x4와 인트라 16x16으로 나뉜다. 제안된 알고리듬은 각 인트라 부호화 방법에 따라, 스크램블링으로 인해 비트량의 변화가 발생하지 않는 범위 내에서 인트라 예측 모드를 변경한다. 이와 같이 인트라 예측 모드의 변경을 통해 영상을 스크램블링 할 경우, 인트라 예측 모드를 사용하지 않는 인터 블록은 직접적으로 왜곡시킬 수 없게 된다. 그러나 인터 프레임은 부호화 시 이전 프레임을 참조하여 부호화하므로, 만약 이전 프레임이 왜곡되었다면 그 영상을 참조하는 인터 프레임에 이전 프레임의 왜곡이 그대로 전파된다. 따라서 인트라 부호화되는 첫 번째 프레임에 가해진 왜곡이 인터 부호화되는 두 번째 프레임에 전파되는 효과를 갖는다. 이전 프레임을 참조하는 인터 부호화 방식의 이러한 특성으로 인해 인트라 블록의 왜곡 효과가 비디오 영상 전체에 전파된다. 따라서 제안된 방법은 인트라 블록의 왜곡만으로 인터 블록을 왜곡 시키는 장점을 갖는다. 반면 기존의 움직임 벡터를 이용한 방법의 경우 인터 블록만을 왜곡하므로 모두 인트라 부호화 되는 첫 번째 프레임의 경우 전혀 왜곡할 수 없다. 또한 앞서 설명한 바와 같이 기존의 대부분의 스크램블링 방법은 압축 효율이 떨어지는 단점을 가지고 있다. 하지만 본 논문에서 제안한 방법은 기존 방법과 달리 스크램블링으로 인한 비트량의 증가가 전혀 없다. 또한, XOR 연산이나 LSB의 변환과 같은 단순한 연산을 통해 예측 모드를 변경하므로 구현이 용이하며 계산량의 증가가 거의 없는 장점을 가진다.

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, Boston, pp.219-229, 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, Orlando, pp.285-294, 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 졸업.

2004년 성균관대학교 전자전기 공학과 석사 과정.

<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>