ISSN(Print): 2287-5891

http://dx.doi.org/10.3745/KTCCS.2012.1.3.187

Efficient H.264/AVC Video Scrambling Methods for Digital Rights Management

Soojin Kim† · Geun Park† · Kyeongsoon Cho††

ABSTRACT

This paper describes efficient H.264/AVC video scrambling methods for digital rights management. The proposed scrambling methods are to scramble level and suffix in entropy encoding and MVD in motion estimation of the H.264 video compression process. Other scrambling methods have been proposed but they degrade the compression efficiency or make it difficult to achieve real-time processing due to the large amount of computational efforts. Since the proposed scrambling methods resolve the drawbacks of other approaches, they do not cause image distortion and the original compression efficiency is maintained. We verified our scrambling methods and evaluated the performance by conducting several experiments with H.264 reference program. Finally, we implemented video player system using USB dongle in order to apply the proposed scrambling/descrambling methods to H.264 video compression.

Keywords : H.264/AVC, Scrambling/Descrambling, Digital Rights Management, Video player system

디지털 저작권 관리를 위한 효율적인 H.264/AVC 비디오 스크램블링 방법

김 수 진† · 박 균† · 조 경 순††

요 약

본 논문은 디지털 저작권 관리를 위한 효율적인 H.264/AVC 비디오 스크램블링 방법을 제안한다. 제안하는 방법은 H.264 동영상 압축 방법에서 엔트로피 부호화에서 사용되는 레벨 및 suffix와 움직임 예측에서 사용되는 MVD에 스크램블링 방법을 적용하는 것이다. 다른 논문들에서 제안된 방법들은 데이터의 압축 효율을 감소시키거나 많은 연산량으로 인해 실시간 처리가 불가능하다는 문제점이 있다. 본 논문에서 제안하는 스크램블링/디스크램블링 방법은 다른 논문에서 제안한 방식들의 문제점을 개선시켜 복원된 영상에 왜곡을 일으키지 않을 뿐만 아니라 압축 효율을 원래의 압축 방법 그대로 유지한다. H.264 레퍼런스 프로그램을 이용한 실험을 통해 제안하는 방법의 성능 및 동작을 검증하였으며, USB 동글을 이용하여 제안하는 스크램블링/디스크램블링 방식을 H.264 비디오 압축에 적용할 수 있는 동영상 재생 시스템을 구현하였다.

키워드 : H.264/AVC, 스크램블링/디스크램블링, 디지털 저작권 관리, 동영상 재생 시스템

1. 서 론

영상 데이터가 디지털화되고 통신 기술이 발전함에 따라 고화질의 영상 데이터 전송이 보편화 되었으며, 유선망을 이용한 인터넷과 3G(3rd Generation), WiBro(Wireless Broadband Internet) 같은 무선 인터넷 서비스가 보급되고 고속 무선 데이터 패킷통신 규격인 LTE(Long Term Evolution) 방식이 상용화되면서 영상 콘텐츠에 대한 이용이 급증하고 있다. 또한 HD(High Definition)급 화질을 지원하는 디지털 방송이 보급되면서 디지털 영상 콘텐츠에 대한 이용이 급격하게 증가하고 있다. 이에 따라 H.264/AVC (Advanced Video Coding)[1] 동영상 압축 표준을 이용한 IP TV(Internet Protocol Television) 등과 같은 상용화 서비스들의 보급 속도가 크게 증가하고 있다.

영상 매체의 발전은 유료 케이블 방송의 활성화에도 영향을 미치게 되었다. 디지털 셋톱박스에 탑재되어 암호화를 통해 특정 방송물의 수신을 제한하는 수신 제한 시스템, 즉 CAS(Conditional Access System)는 유료 방송 시청 및 서비스를 가능하게 하는 핵심 기술이다. 수신 제한 시스템은 유료 방송 서비스에서 요금을 지불한 정당한 가입자만이 콘

※ 이 논문은 2012년도 한국외국어대학교 교내학술연구비의 지원에 의한 것임.
† 준 회 원 : 한국외국어대학교 전자공학과 박사과정
†† 정 회 원 : 한국외국어대학교 전자공학과 정교수
논문접수 : 2012년 3월 29일
수 정 일 : 1차 2012년 9월 19일
심사완료 : 2012년 9월 20일
* Corresponding Author : Kyeongsoon Cho(kscho@hufs.ac.kr)

텐츠를 시청할 수 있도록 하기 위한 기술로, Fig. 1과 같이 콘텐츠를 스크램블링/디스크램블링하는 기능을 포함한다. 스크램블링은 보호하고자 하는 영상을 약속된 키(key)를 이용하여 변형함으로써 키를 모르는 사용자가 수신된 영상을 정상적으로 복원하지 못하게 하는 방식이며, 디지털 영상 콘텐츠에 대해 허락된 사용자의 권리를 보호하기 위해 다양한 스크램블링 기술들이 연구되어 왔다[2-7].

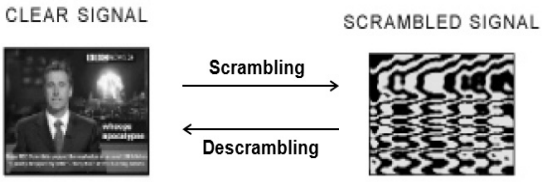


Fig. 1. Video contents protection using scrambling method

유료 콘텐츠를 보호하기 위해서 AES(Advanced Encryption Standard), DES(Data Encryption Standard), SEED 등과 같은 블록 암호화 방법들이 연구되었지만, 압축된 영상 데이터 전체를 암호화해야 하기 때문에 연산량이 많아 데이터를 실시간으로 처리하기 어렵다. DCT(Discrete Cosine Transform) 블록과 움직임 벡터를 스크램블링하는 방법[2]이나 엔트로피 부호화 단계에서 스크램블링하는 방법[3]은 복원된 영상에 왜곡이 발생하거나 스크램블링 후 데이터가 증가하여 압축 효율을 감소시키는 단점이 있다. 인트라 블록을 스크램블링하는 방법[4]은 배경의 변화가 거의 없는 영상을 크게 왜곡시킬 수 없는 단점이 있다.

본 논문에서는 복원된 영상의 왜곡, 압축 효율 감소 등과 같은 다른 스크램블링 방법들의 문제점을 개선하고 영상 데이터를 효과적으로 보호할 수 있는 스크램블링 방식을 제안한다. 제안하는 방식은 현재 많은 모바일 기기에서 사용되고 있는 H.264 동영상 압축 표준의 특징을 이용하였으며, 엔트로피 부호화 과정에서 사용되는 0이 아닌 계수들의 레벨과 이 레벨을 구성하는 suffix에 스크램블링을 적용하는 것과 움직임 추정 과정에서 사용되는 움직임 벡터에 스크램블링을 적용하는 것을 포함한다. 제안하는 스크램블링 방식은 데이터의 크기를 변경시키지 않기 때문에 압축 효율에 손실이 전혀 없으며, 기존 인코딩 시간에 비해 시간 증가율이 작다. 또한 본 논문에서는 USB(Universal Serial Bus) 동글(dongle)을 이용하여 제안하는 방법을 적용할 수 있는 동영상 재생 시스템을 구현하였다.

2. H.264/AVC 동영상 압축 표준

H.264/AVC 동영상 압축은 Fig. 2와 같이 변환, 양자화, 인트라 예측, 움직임 추정, 움직임 보상, 후처리 필터, 엔트로피 부호화 등의 다양한 압축 도구들을 사용하여 이루어진다.

입력 영상 F_n은 매크로블록 단위로 처리되며, 각 매크로블록은 인트라와 인터 모드를 적용하여 압축된다. 인트라 모드는 현재 영상에서 이전 매크로블록을 압축하였다가 복원된 값 uF'_n을 이용하여 현재 매크로블록에 대해 가장 유사한 값들을 탐색한다. 인터 모드는 한 장 이상의 참조 프레임 (F'_{n-1})을 대상으로 현재 영상의 매크로블록에 대해 가장 유사한 블록을 탐색한다. 이와 같은 방법들을 적용하여 탐색된 가장 유사한 데이터 P는 현재 블록의 데이터와 차분 값 D_n을 구한다. D_n은 블록 단위의 변환기 (T)를 이용하여 공간 영역에서 주파수 영역의 데이터들로 변환되고 양자화 과정 (Q)을 거쳐 X 값으로 계산된다. 양자화된 계수 X는 순서를 재정렬(Reorder)하고 엔트로피 부호화 과정(Entropy Encoder)을 거친다. 엔트로피 부호화된 계수들은 디코더에서 복원을 하기 위해 동영상 압축 표준에서 정의된 비트스트림 규격으로 만들어진다. 비트스트림은 각 매크로블록 단위로 예측 모드, 양자화 파라미터, 움직임 벡터 등의 정보를 갖는다. 동영상을 압축하는 과정은 매크로블록 단위로 압축하여 비트스트림을 만드는 과정뿐만 아니라 다음 영상의 예측을 위함 참조 프레임을 생성하는 디코딩 과정이 포함된다. 양자화된 계수 X는 역양자화 과정 (Q^{-1})과 역변환 과정 (T^{-1})을 거쳐 차분 값 D'_n으로 변환된다. D'_n은 예측 과정에서 얻은 P와 합하여 참조 프레임 uF'_n을 만들어낸다. 필터는 블록 단위로 계산된 참조 프레임 uF'_n이 가지고 있는 블록 왜곡 현상을 제거하기 위해 사용된다.

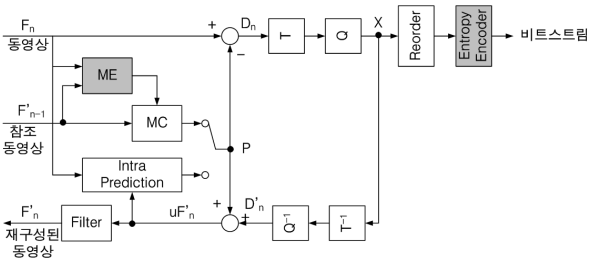


Fig. 2. Proposed method in H.264 video compression

3. 제안하는 스크램블링 방법

H.264 동영상 압축은 2절의 Fig. 2와 같이 다양한 압축 도구들을 사용하여 이루어진다. 디지털 영상 콘텐츠를 보호하기 위해 H.264 압축 표준에 스크램블링 방식을 적용한 다른 알고리즘들은 인코딩 및 디코딩 과정에서 추가적인 시간을 필요로 하거나 압축 효율을 감소시킨다. 논문 [2]에서는 변환 과정에서 DCT 블록의 순서를 바꾸고 DCT 계수의 부호 비트를 바꾸는 방식과 움직임 추정 과정에서 움직임 벡터에 스크램블링을 적용하는 방식을 제안한다. 이 방식은 매우 간단하지만 기존 압축 방식에 비해 연산량이 증가하여 시간이 더 오래 걸리고 압축된 데이터의 크기가 증가한다는 단점이 있다. 엔트로피 부호화 단계에서 사용되는 허프만 코드에 스크램블링을 적용하는 논문[3]의 방식 또한 압축 시간과 압축된 데이터의 크기가 증가한다. 인트라 예측 과정에서 인트라 블록을 스크램블링하는 논문[4]의 방식은 영상

의 공간적 특성만 이용하기 때문에 배경의 변화가 거의 없거나 객체의 움직임이 매우 적은 영상을 크게 왜곡할 수 없다는 단점이 있다.

따라서 본 논문에서는 기존에 발표된 스크램블링 방식들의 단점을 보완할 수 있는 스크램블링 방법을 제안하며, 제안하는 방법들은 Fig. 2에 회색으로 표시된 것과 같이 엔트로피 부호화(Entropy Encoder)와 움직임 추정(ME) 단계에 적용되었다. 제안하는 방식은 압축 효율을 원래의 압축 방법 그대로 유지하며, 배경이나 움직임의 변화가 거의 없는 경우에도 영상을 효율적으로 왜곡시킬 수 있다.

3.1 엔트로피 부호화에 적용한 스크램블링 방법

H.264의 대표적인 엔트로피 부호화 방법으로 CAVLC (Context Adaptive Variable Length Coding)가 있다. CAVLC는 지그재그 스캔으로 재배열된 계수들의 인코딩을 위한 목적으로 사용되며 다음과 같은 특징을 갖는다.

예측, 변환, 그리고 양자화 과정을 거친 블록들은 일반적으로 대부분 0을 포함하는데, CAVLC는 이를 압축하기 위해 run-level 코딩을 사용한다. 먼저 지그재그 스캔 이후 +1 또는 -1 값을 갖는 0이 아닌 계수들에 대해 Trailing Ones를 이용하여 인코딩한다. 0이 아닌 계수들의 개수는 look-up 테이블에 의해 인코딩되며, 이 테이블은 인접한 블록 내의 0이 아닌 계수들의 개수에 따라 선택된다. 0이 아닌 계수들의 레벨(level)은 재배열된 배열의 시작점인 DC 계수 주변에서 보다 큰 영향이 있고 고주파 쪽으로 갈수록 작아진다. CAVLC는 이러한 특징들을 이용하여 최근에 부호화된 레벨의 크기에 따라 VLC look-up 테이블을 달리하여 사용한다[8].

CAVLC 단계는 Fig. 3와 같이 크게 다섯 단계로 나누어진다. 첫 번째 단계에서는 0이 아닌 계수들의 전체 개수와 ±1의 값을 갖는 계수의 개수인 Trailing Ones를 인코딩하며, 두 번째 단계에서는 이 Trailing Ones의 부호를 인코딩한다. 세 번째 단계에서는 나머지 0이 아닌 계수들의 레벨을 인코딩하며 네 번째와 다섯 번째 단계에서는 마지막 계수 이전의 전체 0의 개수 및 각 0이 아닌 계수 앞의 0의 개수를 인코딩한다.

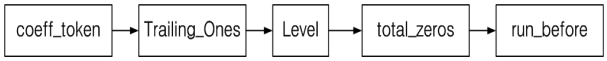


Fig. 3. CAVLC steps in H.264

본 논문에서 제안하는 스크램블링 방법은 0이 아닌 계수들의 레벨을 인코딩하는 CAVLC의 세 번째 단계에서 적용하였다. 제안하는 방식은 먼저 레벨 값의 부호를 변경하는 것이다. CAVLC에서 레벨은 음수 또는 양수의 값을 갖는 0이 아닌 계수이다. 따라서 레벨의 부호를 변경하게 되면 전체적인 화면을 효과적으로 왜곡시킬 수 있다. 두 번째 방식은 레벨을 구성하는 prefix와 suffix 중에서 suffix의 값과 암호화키 값에 대해 XOR(Exclusive OR) 연산을 하는 것이다. Prefix와 suffix는 인코딩된 영상 데이터를 디코딩할 때 원본 데이터를 복원하기 위해 사용되는 값이다. Prefix는 원본 데이터의 계수 값에 직접적인 영향을 주는 정보를 갖고 있기 때문에 이 값을 변경하면 원본 데이터를 복원할 수 없다. 따라서 제안하는 방식에서는 suffix 값만을 이용하여 스크램블링을 하며, 키와 XOR 연산을 하기 위한 비트 위치는 suffix의 값에 따라 다르게 정하였다.

Table 1. Scrambling method using suffix length

Suffix length	Positions of XOR operation
1	0x0100
2	0x0030
3	0x0070
4	0x0f00
5	0x01f0
6	0x03f0

Table 1은 suffix의 길이에 따라 키 값과 XOR 연산을 하기 위한 비트 위치의 예를 나타낸다. 각 suffix의 길이에 따라 스크램블링을 위한 XOR 연산을 하는 비트 위치가 다르기 때문에 허가받지 못한 사용자가 암호화키를 알게 되더라도 원본 데이터로 복원하는 것이 불가능하다. 엔트로피 부호화 단계에 적용하기 위해 제안하는 스크램블링 방식은 단순히 레벨의 부호와 suffix의 값을 변경하기 때문에 압축된 데이터의 크기에 영향을 미치지 않고 영상을 효과적으로 스크램블링 할 수 있다. 디스크램블링에서는 스크램블링에 적용한 키와 XOR 연산 위치를 그대로 적용하여 원래의 동영상 데이터로 복원할 수 있다.

3.2 움직임 추정에 적용한 스크램블링 방법

연속된 비디오 프레임 사이에는 여러 요소들로 인해 인접한 프레임 간에 데이터 변화가 발생하게 되며, 대부분의 오차는 프레임 사이의 객체, 즉 픽셀의 움직임에 해당하는 변화로 인해 발생한다. 이러한 객체의 움직임을 분석하여 현재 프레임과 이전 프레임 간의 유사성을 확인하여 영상 데이터를 효율적으로 압축하기 위해 움직임 추정 방식이 사용된다.

움직임 추정 방식은 이웃한 비디오 프레임들 간의 시간적 중복성을 제거하여 높은 압축율을 얻는 방식이며, Fig. 4는 다중 참조 프레임을 이용한 움직임 추정 과정을 나타낸다.

움직임 추정 방식에서는 현재 프레임의 블록과 이전 프레임들의 동일 위치에 있는 블록 주위의 탐색 영역 내에서 유사한 값을 갖는 블록이 있는지를 판단하고, 현재 프레임의

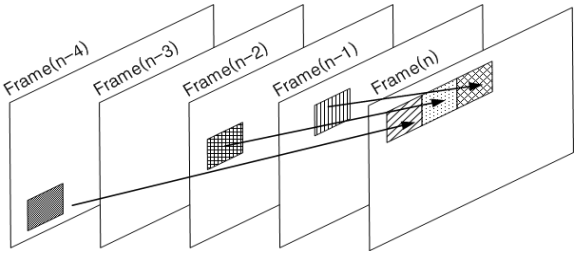
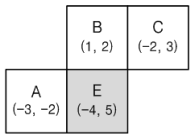


Fig. 4. Motion estimation using multiple reference frames

블록과 유사한 블록이 이전 프레임들의 탐색 영역 내에 존재한다면 두 블록 간의 위치 오차 정보인 움직임 벡터 MV (Motion Vector)를 압축한다. MV의 값이 클 경우 압축할 데이터의 양이 많아지게 되며, 주변 블록의 움직임은 거의 비슷하게 형성되기 때문에 연속적으로 큰 값들을 압축해야 한다. 하지만 일반적인 상황에서 객체의 움직임은 일정한 속도와 방향을 갖는 경우가 많기 때문에 주변 블록의 MV도 비슷한 값을 갖게 되는 경우가 많다. 따라서 현재 블록에 대한 MV의 값을 근처 블록의 MV와의 차이만을 이용하여 압축하면 각각의 MV를 그대로 압축하는 것보다 압축량이 훨씬 더 줄어들게 된다.

예측 움직임 벡터 MVP (Motion Vector Prediction)는 이전에 계산된 MV에 의해 생성되며, 현재의 MV와 MVP 간의 차이인 MVD(Motion Vector Difference)가 압축되어 전송된다. 예를 들어 Fig. 5에서 현재 매크로블록 E에 대한 MV인 (-4, 5)는 그대로 압축되지 않고 주변 MV에 대한 정보를 이용하여 계산된 MVD가 압축되어 전송된다. 현재 매크로블록 E에 대한 MVD를 계산하는 방법은 다음과 같다.



$E: MVP_X = Median(MVA_X, MVB_X, MVC_X) = Median(-3, 1, -2) = -2$

$E: MVD_X = MVE_X - MVP_X = -4 - (-2) = -2$

$E: MVP_Y = Median(MVA_Y, MVB_Y, MVC_Y) = Median(-2, 2, 3) = 2$

$E: MVD_Y = MVE_Y - MVP_Y = 5 - 2 = 3$

Fig. 5. Examples of MVD calculation

현재 매크로블록 E에 대한 MVP는 MVP_X와 MVP_Y로 나누어 표현될 수 있고, 주변 참조 블록 A~C의 MV_A~MV_C에 대한 Median 값으로 계산된다. 여기에서 Median은 나열된 수들 중에서 중간 값을 취하는 것이며, 과도하게 적거나 큰 값으로 인한 평균값의 오류를 막기 위해 이용된다. 매크로블록 E에 대한 MVP_X와 MVP_Y는 Fig. 5와 같이 (-2, 2)로 계산되고, 최종 압축될 MVD는 원래의 MV와 MVP의 차이 값으로 계산된다. 따라서 현재 매크로블록 E에 대해 압축될 움직임 정보는 (-2, 3)이다. 결국 압축되어야 할 데이터가 (-4, 5)에서 (-2, 3)이 되므로 압축해야 할 데이터의 양을 줄일 수 있게 된다. MVD를 압축하여 전송하는 방법은 MVP가 원래 MV보다 큰 특정 경우에는 비효율적이지만, 일반적인 경우에는 데이터 압축율을 높일 수 있는 효율적인 방법이 된다.

본 논문에서 제안하는 두 번째 스크램블링 방법은 H.264의 움직임 추정 과정에서 사용되는 MVD에 스크램블링을 적용하는 것이다. MVD에 적용하는 스크램블링에 대해 본 논문에서 제안하는 첫 번째 방식은 MVD의 부호를 바꾸는 것이다. 이때 모든 MVD의 부호를 바꾸지 않고 특정 조건을 만족하는 경우에만 부호를 변경하였다. Table 2는 MVD의 절대 값에 따라 적용되는 키의 비트 위치를 나타낸다. MVD의 절대 값에 따라 결정되는 키의 비트 위치에 해당하는 값이 1이면 해당 MVD의 부호를 바꾼다. Fig. 5의 예에서 결정된 MVD는 (-2, 3)이고 MVD의 절대 값은 (2, 3)이 된다. 따라서 MVD_X와 MVD_Y에 적용될 키의 비트 위치는 모두 0x0010이 된다. 따라서 암호화키의 0x0010 번째 비트가 1이면 MVD의 부호를 변경하여 (2, -3)으로 바꾸고 0이면 기존 MVD의 값을 유지시킨다. 이러한 방식은 MVD와 키의 값에 따라 부호를 변경하는 조건이 달라지기 때문에 허가받지 못한 사용자가 원본 영상을 복원하는 것을 방지할 수 있다.

Table 2. Conditions for sign bit substitution using MVD

Absolute value of MVD	Positions of key bit
1	0x0001
2~3	0x0010
4~7	0x0002
≥ 8	0x0020

본 논문에서 MVD에 스크램블링을 적용하기 위해 제안하는 두 번째 방법은 MVD의 값을 Y축으로 대칭시키는 것이다. Table 3는 본 논문에서 적용한 Y축 대칭 방법을 나타낸다. 표를 통해 알 수 있듯이 Y축으로 대칭시키는 비트 수는 MVD의 비트 수보다 작다. 이는 스크램블링된 데이터를 제대로 복원시키기 위해 MSB(Most Significant Bit)를 제외한 나머지 비트들을 Y축 대칭시키기 때문이다. 예를 들어 MVD의 절대 값이 8(이진수: 1000)인 경우 모든 비트를 Y축 대칭시키면 1이 되고, MVD의 절대 값이 2(이진수: 10)이거나 4(이진수: 100)인 경우에도 Y축 대칭의 결과는 모두 1이 된다. 따라서 MVD의 모든 비트를 Y축 대칭시키면 디스크램블링할 때 원래의 데이터로 복원시키는 것이 불가능하다. 이러한 경우를 방지하기 위해서 본 논문에서는 MSB를 제외한 비트들을 Y축 대칭시킨다. MVD의 절대 값이 6(이진수: 110)인 경우 스크램블링된 결과는 '101'이 되며, 디스크램블링을 할 때에도 MSB를 제외한 나머지 비트들을 Y축 대칭시켜 '110'으로 복원한다. 이 방법은 MVD의 절대 값이 2의 제곱수인 경우에는 영향을 미치지 않지만 대부분의 MVD 값이 2의 제곱수가 아닌 경우가 더 많기 때문에 효과적인 스크램블링 방법이 될 수 있다. MVD에 적용하기 위해 본 논문에서 제안된 스크램블링 방식은 MVD의 부호를 변경하거나 Y축 대칭을 통해 단순히 값을 바꾸는 것이기 때문에 압축된 데이터의 크기에 영향을 미치지 않고 영상을 효과적으로 스크램블링할 수 있다. 디스크램블링에서는 스크램블링에 적용한 키와 부호 반전 및 Y축 대칭 방법을 거꾸로 적용하여 원래의 동영상 데이터로 복원할 수 있다.

Table 3. Conditions for y-axis symmetricity using MVD

MVD		# of y-axis symmetricity bits
Absolute value	# of bits	
0	0	0
1	1	0
2~3	2	0
4~7	3	2
≥ 8	≥ 4	≥ 3

4. 제안하는 방법을 이용한 동영상 재생 시스템

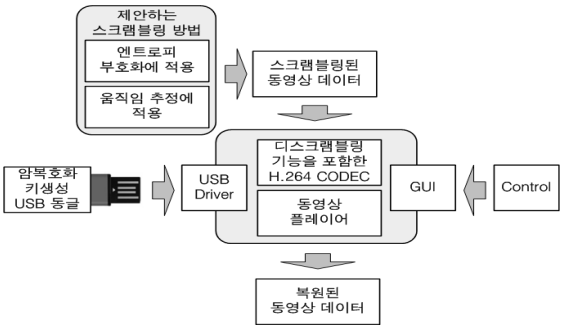


Fig. 6. Proposed video player system

Table 4. Environments for experiments

CPU	Inter Core2 T7200 (2.0 GHz)
RAM	2 GB
VGA	ATI Mobility Radeon X1350
OS	Windows 7 Professional K 32-bit
Profile type	Baseline Profile
Frame rate	30 fps
QP	28

Fig. 6는 본 논문에서 제안하는 스크램블링 및 디스크램블링 방법을 적용한 동영상 재생 시스템의 구조를 나타내며, Table 4는 시스템을 동작시키기 위해 사용한 실험 환경을 나타낸다. 본 논문에서 제안하는 스크램블링 방식들을 적용하여 압축된 동영상 데이터를 디스크램블링하여 원본 영상을 복원시키기 위해서는 Fig. 6와 같이 USB 동글을 통해 인증된 키를 입력받아야 한다. USB 동글을 통해 인증된 키가 입력되면 스크램블링된 동영상 데이터를 제대로 디스크램블링하여 원본 동영상으로 복원시킬 수 있다. 하지만 USB 동글이 삽입되지 않거나 인증된 키가 입력되지 않으면 제대로 된 디스크램블링을 할 수 없기 때문에 복원된 동영상에 왜곡이 발생한다.

Fig. 7은 USB 동글로부터 인증된 키가 입력되었을 때 디스크램블링되어 복원된 동영상의 재생 결과를 나타내며, Fig. 8은 USB 동글이 삽입되지 않았을 때의 동영상 재생

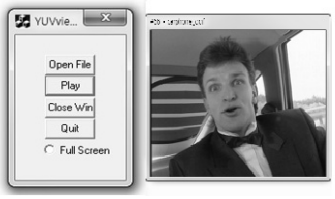


Fig. 7. Case with authenticated key

Fig. 8. Case without authenticated key

결과를 나타낸다. Fig. 7과 Fig. 8을 통해 알 수 있듯이 USB 동글로부터 인증된 키가 입력되면 동영상을 제대로 복원시킬 수 있지만, USB 동글이 삽입되지 않은 경우에는 복원된 동영상에 왜곡이 발생하며 키가 발견되지 않았다는 메시지가 출력된다.

5. 실험 결과

본 논문에서는 디지털 저작권 관리를 위해 H.264/AVC에 적용할 수 있는 효율적인 스크램블링/디스크램블링 방식을 제안하였으며, 제안하는 방법을 구현하고 검증하기 위해 H.264 레퍼런스 프로그램인 JM11과 YUV Viewer의 오픈소스 프로그램을 이용하였다. 실험을 위해 사용한 동영상 데이터는 애리조나 대학교에서 제공하는 foreman, akiyo, carphone, mother-daughter이며 각 동영상은 177x144 화소의 QCIF(Quarter Common Intermediate Format) 영상 300장 이상으로 구성되어 있다. Fig. 9의 네 개의 영상은 각각 foreman과 akiyo 영상의 100번째 프레임과 carphone과 mother-daughter 영상의 108번째 프레임을 나타낸다.

Fig. 9. Original video images

Fig. 10. Scrambled images using proposed methods

Fig. 10은 본 논문에서 제안하는 두 가지 스크램블링 방법인 엔트로피 부호화에서 사용되는 레벨 및 suffix와 움직임 추정에서 사용되는 MVD에 스크램블링을 적용한 결과를 나타낸다. 스크램블링된 결과를 통해 알 수 있듯이 본 논문에서 제안하는 첫 번째 스크램블링 방법은 레벨과 suffix의 값을 변화시키기 때문에 픽셀 값 자체에 영향을 끼쳐 원본 영상의 색깔을 변화시키는데 큰 영향을 미친다. 두 번째 스크램블링 방법은 MVD 값을 변화시켰기 때문에 움직임이 많은 영상을 왜곡시키는데 큰 영향을 미치게 된다.

Table 5. Comparison of encoding time (time unit: second)

	foreman	carphone	mother-daughter	akiyo
Case 1	170.95	218.34	160.08	161.57
Case 2	171.30	218.86	160.33	161.76
Case 3	172.28	220.05	162.00	163.02
Increase ratio 1	0.21%	0.24%	0.16%	0.12%
Increase ratio 2	0.77%	0.78%	1.18%	0.89%

Table 5는 각 실험 영상에 대해 스크램블링 방식이 적용되지 않은 경우(Case 1)와 제안하는 스크램블링 방식이 적용된 경우(Case 2), 그리고 경우 2에서 USB 동글로부터 키 값을 전달받는 시간을 함께 고려한 경우(Case 3)의 인코딩 시간을 나타낸다. 표에 나타낸 Increase ratio 1과 Increase ratio 2는 각각 Case 1에 대한 Case 2 및 Case 1에 대한 Case 3의 인코딩 시간 증가율을 의미한다. Table 5를 통해 제안하는 스크램블링 방법을 적용해도 기존 인코딩 시간에 비해 증가량이 작은 것을 확인할 수 있다.

(a) (b) (c)

Fig. 11. Comparison of image distortion

Fig. 11은 본 논문에서 제안하는 방식을 적용한 foreman의 영상(a)와 원본 영상(b), 그리고 논문 [6]에서 제안한 방식을 적용한 영상(c)의 왜곡 정도를 나타낸다. 그림을 통해 알 수 있듯이 본 논문에서 제안하는 스크램블링 방법은 논문 [6]에서 제안하는 방법에 비해 화면 전체를 효과적으로 왜곡시킬 수 있다.

Table 6. Comparison of average bit increase ratio (Kbit/s)

	[6]	[7]	Proposed
foreman	7.33%	51.52%	0%
carphone	6.43%	121.63%	0%

Table 6는 본 논문에서 제안하는 스크램블링 방식과 논문 [6], [7]에서 제안한 방식을 적용했을 때의 원본 대비 평균 비트율 증가량을 나타낸 것이다. 본 논문에서 제안하는 방식은 데이터의 크기를 변경시키지 않기 때문에 비트율 증가율이 0%이고, 따라서 압축 효율에 손실이 없다.

6. 결 론

본 논문에서는 디지털 저작권 관리를 위한 H.264/AVC 비디오 스크램블링 방법 및 이를 적용한 동영상 재생 시스템의 구현을 제안한다. 제안하는 방식은 H.264 동영상 압축의 엔트로피 부호화와 움직임 추정 과정에서 스크램블링을 적용하는 것이며, 다른 논문에서 제안된 방식들과는 달리 복원된 영상에 왜곡을 일으키지 않을 뿐만 아니라 압축 효율에 손실을 가져오지 않는다. 따라서 본 논문에서 제안하는 효율적인 스크램블링 및 디스크램블링 방법은 H.264 동영상 압축을 사용하는 다양한 애플리케이션 및 수신 제한 시스템과 같은 유료 디지털 방송 기술에서도 적용되어 사용될 수 있다. 향후 연구 계획으로 본 논문에서 제안한 스크램블링 방식의 외부 공격에 대한 강인한 정도를 측정하는 실험을 통해 제안하는 방식으로 스크램블링된 영상 데이터의 기밀성을 확인하고자 한다. 또한 제안하는 방식을 적용하였을 때 인코딩 시간을 원래 압축 방식 그대로 유지시킬 수 있는 방안에 대해서도 연구할 계획이다.

참 고 문 헌

[1] ITU-T, “Recommendation and International Standard of Joint Video Specification,” ITU-T Rec. H.264/ISO/IEC 14497-10 AVC, October, 2004.

[2] W. Zeng and S. Lei, “Efficient Frequency Domain Selective Scrambling of Digital Video,” IEEE Trans. on Multimedia, Vol.5, No.1, pp.118-129, March, 2003.

[3] M. S. Kankanhalli and T. T. Guan, “Compressed -domain Scrambler/Descrambler for Digital Video,” IEEE Trans. on Consumer Electronics, Vol.48, No.2, pp.356-365, May, 2002.

[4] J. Ahn, *et al.*, “Digital Video Scrambling Method using Intra Prediction Mode,” in Proceedings of Pacific-Rim Conference on Multimedia, Vol.3333, pp.386-393, November, 2004.

[5] J. Ahn, B. Jeon, “Digital Video Scrambling Method using Motion Vector and Intra Prediction Mode” Journal of the Institute of Electronics Engineers of Korea, Vol.42, No.4, pp.133-142, July, 2007.

[6] L. Tong, *et al.*, “Prediction Restricted H.264/AVC Video Scrambling for Privacy Protection,” IET Electronics Letters, Vol.46, No.1, pp.47-49, January, 2010.

[7] F. Dufaux and T. Ebrahimi, “H.264/AVC Video Scrambling for Privacy Protection,” in proceedings of IEEE International Conference on Image Processing, pp.1688-1691, October, 2008.

[8] Iain E. G. Richardson, “H.264 and MPEG-4 Video Compression(Video Coding for Next Generation Multimedia),” John Wiley & Sons, 2003.

김 수 진

e-mail : ksjsky9888@hufs.ac.kr
2007년 한국외국어대학교 전자공학과(학사)
2009년 한국외국어대학교 전자공학과 (공학석사)
2009년~현 재 한국외국어대학교 전자공학과 박사과정
관심분야 : SoC 설계 등

박 균

e-mail : datame@hufs.ac.kr
2008년 한국외국어대학교 전자공학과(학사)
2010년 한국외국어대학교 전자공학과 (공학석사)
2010년~현 재 한국외국어대학교 전자공학과 박사과정
관심분야 : SoC 설계 등

조 경 순

e-mail : kscho@hufs.ac.kr
1982년 서울대학교 전자공학과(학사)
1984년 서울대학교 전자공학과(공학석사)
1988년 미국 Carnegie Mellon University 대학원 전기 및 컴퓨터 공학과 (공학박사)
1988년~1994년 삼성전자(주) 반도체 총괄 선임, 수석연구원
1994년~현 재 한국외국어대학교 전자공학과 정교수
관심분야 : SoC 설계 등