Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.2, Apr. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.2.311

관심영역 암호화 시 발생하는 H.264 영상의 비트레이트 오버헤드 최소화 방법 연구*

손 동 열,[1†] 김 지 민,[2] 지 청 민,[2] 김 강 석,[3] 김 기 형,[3] 홍 만 표[3‡]
[1]아주대학교 지식정보공학과, [2]아주대학교 컴퓨터공학과, [3]아주대학교 사이버보안학과

A Study on the Method of Minimizing the Bit-Rate Overhead of H.264 Video when Encrypting the Region of Interest*

Dongyeol Son,[1†] Jimin Kim,[2] Cheongmin Ji,[2] Kangseok Kim,[3]
Kihyung Kim,[3] Manpyo Hong[3‡]
[1]Department of Knowledge Information Engineering, Ajou University,
[2]Department of Computer Engineering, Ajou University,
[3]Department of Cyber Security, Ajou University

요 약

H.264/AVC-MPEG의 JM v10.2 코드 기반에서 QCIF (176x144) 해상도를 가지는 News 샘플 영상을 사용하여 실험을 하였다. 암호화를 하게 될 관심영역(Region of Interest, ROI)이 H.264 표준의 움직임 예측 및 보상의 특성상 연속적으로 각 프레임마다 불필요하게 참조하여 드리프트를 발생시켰다. 드리프트를 완화하기 위해 암호화가 된 I픽처를 특정 주기로 재삽입하는 최신 관련연구의 방법은 추가 연산량 증가로 이어져 영상 전체의 비트레이트 오버헤드가 증가하는 요인이 된다. 따라서 움직임 예측 및 보상 단계에서 각 프레임마다 암호화가 될 관심영역에서의 Block과 Frame의 참조 탐색 범위를 제한하고, 암호화가 되지 않을 비관심영역에서의 참조 탐색 범위는 정상적인 인코딩 효율을 유지하기 위해 제한하지 않는다. 이와 같이 특정 참조 탐색 범위가 제한된 영상 인코딩을 한 후, 영상 속 개인정보 보호를 위해 얼굴과 같이 개인 식별이 가능한 관심영역에 대해 RC4 비트스트림 암호화 하는 방법을 제안한다. 그리고 동일한 환경의 조건에서 암호화되지 않은 원본 영상과 최신 관련연구 방법과 본 연구의 제안 방법을 각각 구현한 후, 실험 결과들을 비교·분석하였다. 최신 관련연구 방법과 다르게 제안방법을 통해 시간상 드리프트를 완화하면서, 제안방법이 적용된 영상 전체의 비트레이트 오버헤드가 원본 영상보다 2.35% 증가되고 최신 관련연구 방법보다 14.93% 감소되었다. 이와 같이 향상된 결과는 본 연구의 실험을 통해 입증하였다.

ABSTRACT

This paper has experimented using News sample video with QCIF (176x144) resolution in JM v10.2 code of H.264/AVC-MPEG. The region of interest (ROI) to be encrypted occurred the drift by unnecessarily referring to each frame continuously in accordance with the characteristics of the motion prediction and compensation of the H.264 standard. In order to mitigate the drift, the latest related research method of re-inserting encrypted I-picture into a certain period leads to

Received(01. 04. 2018), Modified(03. 28. 2018), Accepted(04. 02. 2018)
* 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2015R1D1A1A01060793)
* 본 연구는 과학기술정보통신부 및 정보통신산업진흥원의 "고용계약형 SW 석사과정 지원사업"의 연구결과로 수행되었음
† 주저자, sdy0621@ajou.ac.kr
‡ 교신저자, mphong@ajou.ac.kr(Corresponding author)

an increase in the amount of additional computation that becomes the factor increasing the bit-rate overhead of the entire video. Therefore, the reference search range of the block and the frame in the ROI to be encrypted is restricted in the motion prediction and compensation for each frame, and the reference search range in the non-ROI not to be encrypted is not restricted to maintain the normal encoding efficiency. In this way, after encoding the video with restricted reference search range, this article proposes a method of RC4 bit-stream encryption for the ROI such as the face to be able to identify in order to protect personal information in the video. Also, it is compared and analyzed the experimental results after implementing the unencrypted original video, the latest related research method, and the proposed method in the condition of the same environment. In contrast to the latest related research method, the bit-rate overhead of the proposed method is 2.35% higher than that of the original video and 14.93% lower than that of the latest related method, while mitigating temporal drift through the proposed method. These improved results have verified by experiments of this study.
Keywords: H.264/AVC, Privacy Protection, ROI Encryption, Drift of Video, Minimization of Bit-Rate Overhead

I. 서 론

오늘날, 얼굴인식 관련 컴퓨터 비전과 딥러닝이 발전하면서 지능형 CCTV, 얼굴인식 맞춤형 서비스, 얼굴인식 BlackBox, 얼굴인식 후 사람들의 얼굴 빅데이터 분석을 활용한 타겟 광고 마케팅 등이 산업 속에 등장하게 되었다. 급변하는 산업화와 개발을 위한 영상 빅데이터의 수요에 따라, 영상 속 개인의 동의가 없이 개인정보를 이용하는 침해가 이루어지고 있다. 본 연구는 지능형 CCTV에서 무분별하게 개인을 식별할 수 있는 얼굴을 포함한 영상이 유출됨으로써 개인정보가 침해되지 않도록, 영상 속 얼굴영역을 보호하고자 한다.

지능형 CCTV는 영상처리 기술과 인공지능 기술과 같은 각종 ICT 기술들이 네트워크 기술과 융합되어 급속히 발전해 지능화 되어 가고 있는 CCTV를 말한다[1]. 지능형 CCTV도 기존의 CCTV와 마찬가지로 방범 예방이 주목적을 가진다.

기존의 CCTV는 아날로그 인코딩 방식만을 가졌지만, 지능형 CCTV는 네트워크를 통해 영상이 전송되어야 하므로 네트워크 카메라, IP 카메라, 웹 서버, 웹 카메라 등을 가지는 복합형이다. 즉, CCTV의 영상이 네트워크 환경으로 쉽게 접근이 용이하기 때문에 반대로 외부에서 영상의 유출 위험성과 개인정보 침해의 우려가 증가하고 있다. 실제로, 네트워크 카메라 장비의 IP와 포트번호, ID와 Password만 알더라도 CCTV에 접속해 영상을 획득 및 유포할 수 있다. 그 예로 이러한 허점을 이용하고 무단으로 영상을 획득 후 스트리밍하는 웹 사이트도 존재한다[2][3].

이러한 영상 속 개인정보 침해 상황에 따라, 본 논문에서는 네트워크 전송 전 단계인 CCTV 내부에서 아날로그 영상을 TCP/IP 네트워크를 통하여 전송하게 되는 데이터 패킷인 전체 Video Stream[1] 중 개인의 얼굴 스트림에 대해 암호화를 한다. 이는 개인 정보를 보호함과 동시에, 각종 범죄 수사 또는 법원의 요구와 같은 특수한 상황 때 암호화된 영상을 원본으로 복구 명령에 응할 수 있도록 한다[2].

그리고 일반적으로 영상 데이터를 네트워크로 전송할 때, 대역폭을 줄이기 위해 영상 인코딩 알고리즘을 적용한다. 지능형 CCTV와 같은 네트워크 카메라는 H.264/AVC 표준 인코딩 알고리즘을 가장 많이 사용한다[2].

따라서 본 논문에서는 H.264/AVC-MPEG 표준 인코딩 알고리즘을 사용한 JM v10.2 코드 기반에서 실험을 진행한다. 최신 관련연구[2]와 객관적인 비교·분석의 지표로써, 영상의 비트레이트 오버헤드 발생 이슈와 관련이 없는 부분은 동일한 환경으로 재구현하였다. 인코딩의 시퀀스 타입은 IPPP 순서이고 QCIF (176x144) 해상도를 가지는 News 샘플 영상을 사용하고 공간상 드리프트를 완화시키기 위해 최신 관련연구[2]와 같이 FMO(Flexible Macroblock Ordering)[2][9][10] 기법을 적용하였다. 그리고 본 연구는 암호화로 인해 영상 전체의 비트레이트 오버헤드가 발생되는 원인인 시간상 드리프트의 해결 방법을 최신 관련연구[2]의 I픽처 재삽입하는 방식과는 다르게, H.264/AVC-MPEG의 소프트웨어 기능 분석서인 “JM Reference Software Manual”[4]를 참고하여 본 논문에서 제안하고 있는 방식으로 결과를 도출하였고 비트레이트 오버헤드를 감소시키는 것을 주목적으로 두었다. 본 연구에서 제안하는 방법은 다음과 같다.

H.264의 코덱 인코딩 방식에 의하여 암호화된 관

심영역이 각 프레임마다 관심영역 내에서와 비관심영역까지 불필요하게 계속해서 다시 참조됨으로써 비트레이트 오버헤드가 증가하는 원인을 발견하였다.

이 문제를 최소화하기 위해, 각 프레임마다 관심영역의 슬라이스 그룹 범위 내에서는 참조가 되던 모션 예측의 탐색 범위 설정을 Block, Frame 간의 참조 탐색을 제한시키고 비관심영역의 슬라이스 그룹 범위에서는 본래의 인코딩 효율을 유지하기 위해 Block, Frame 간의 참조 탐색 영역을 제한시키지 않도록 하는 것을 제안한다. 본 제안 방법은 I픽처에서 암호화된 관심영역의 재참조로 인해 발생하는 비트레이트 오버헤드를 현저하게 줄일 수 있다.

이와 같이 관심영역의 Block, Frame 참조 탐색 영역을 제한시키는 인코딩 이후에, 관심영역을 암호화하는 방식으로 실험을 진행하였다.

본 논문에서 제안하는 영상의 비트레이트 오버헤드를 최소화시키는 과정은 본론에서 암호화 하지 않은 원본영상과 관련연구와의 비트레이트 비교 실험을 통해 결과를 입증한다.

본 논문은 2장 배경지식; 3장 관련연구; 4장 제안방법; 5장 제안방법의 실험결과; 6장 결론으로 구성되어 있다.

II. 배경지식

2.1 H.264 비트스트림 구조

H.264/AVC-MPEG은 시작 단계부터 지능형 CCTV처럼 네트워크 전송의 목적에 적합하도록 개발이 되었다. 따라서 효율적인 전송을 위해 인코딩된 H.264의 영상은 비트스트림 구조로 되어 있으며, 네트워크 전송 시에는 Fig. 1.처럼 NAL(Network Abstraction Layer) 단위 구문을 통해 비트스트림을 형식화 한다[5].

NAL은 비트스트림 내 데이터를 형식화하기 위해 사용되고, 다양한 채널이나 저장 미디어에 적합한 전송 방법에 대한 헤더정보를 제공한다. 모든 데이터들은 NAL 안에 있고, 각각은 바이트 정수값을 가지고 있다. NAL은 패킷 지향 시스템과 비트스트림 지향 시스템에서 모두 쓰이기 위한 하나의 일반적인 형식을 갖는다. 패킷 지향전송과 바이트스트림에서 NAL 형식은 바이트스트림 형식에서의 NAL이 접두(prefix)나 여분의 첨가 바이트에 의해 선행된다는 점만 제외하면 일치한다[6].

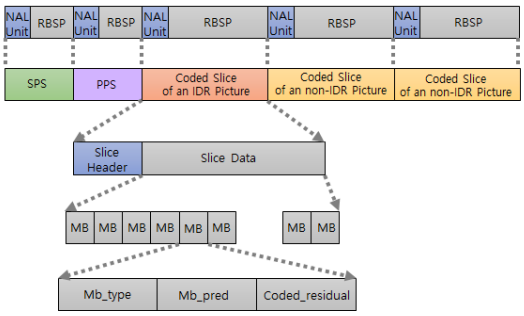

Fig. 1. The analysis of H.264/AVC-MPEG bit-stream structure [5].

한 슬라이스에서 NAL은 헤더인 NAL unit과 데이터를 가지고 있는 Payloads로 구성된다. NAL unit은 데이터의 종류의 지시를 포함하는 구문구조와 실행금지 바이트에 필요한 것으로 산재된 RBSP(Raw Byte Sequence Payload)의 형태의 데이터를 포함하는 바이트이다[7].

NAL unit type은 NAL에 포함되는 RBSP 데이터 구조의 종류를 정의한다. VCL NAL은 이러한 NAL들의 NAL unit type이 1, 2, 3, 4, 5인 NAL들은 VCL NAL로 정의되고, 나머지 NAL들은 비 VCL NAL이라 한다. 세부적으로, H.264 표준에서 NAL unit type별로 NAL 내용과 RBSP 구문구조는 다음과 같다. NAL unit type 0은 정의되어 있지 않고, 1~5는 IDR 화면을 포함하여 부호화된 슬라이스, 6은 추가 향상 정보, 7은 시퀀스 변수 집합, 8은 화면 변수 집합, 9는 화면 경계, 10은 시퀀스의 끝, 11은 스트림의 끝, 12는 채움 문자 데이터로 되어 있다[7].

NAL unit stream은 NAL 단위들의 시퀀스이다. 그리고 바이트 시퀀스 부가정보인 RBSP가 Payload이며, RBSP에는 저장되어 있는 것이 슬라이스 데이터인데, 이 슬라이스의 구성은 Fig. 1.과 같다. RBSP에는 SPS(Sequence Parameter Set), PPS(Picture Parameter Set) 등의 정보와 IDR Picture와 non-IDR Picture에 대한 슬라이스 정보가 구성되어 있다. 그리고 IDR Picture에는 Slice Header와 Slice Data로 이루어져 있으며, 1개의 Slice Data는 여러 개의 Macroblock으로 되어 있다. 한 매크로블록을 복호하기 전에 매크로블록 시작 함수가 매크로블록 복호를 위한 초기화 작업을 수행한다. 복호를 위한 변수들의 초기화가 끝나면, NAL로부터 읽어온 버퍼에서

각 매크로블록의 복호를 위한 값을 읽는다. 이후 각 모드 별로 매크로블록을 복호한다. 여기서 복호된 데이터와 매크로블록 모드와 움직임 벡터를 이용하여 한 매크로블록의 복호 과정을 종료한다[5].

RSBP의 데이터 구조는 RSBP 멈춤 비트와 0개 이상의 부분 순차적인 0의 비트들에 뒤이은 구문요소들을 포함하는 데이터 비트들의 문자열의 형태를 갖거나 null인 NAL 단위에서 캡슐화 된 정수개의 바이트들을 포함하는 의미론적인 데이터 구조이다[7].

RBSP 멈춤 비트(Raw Byte Sequence Payload Stop Bit, 바이트 시퀀스 부가 멈춤 비트)는 데이터 비트들의 문자열 후에 원본 바이트 시퀀스 부가정보에 존재하는 1의 값을 갖는 비트이다. 그리고 RBSP 멈춤 비트에 의해 RBSP의 끝을 알 수 있다. RBSP 내 데이터 비트들의 문자열 끝의 위치는 RBSP 데이터 구조의 종류 순서에 따라 마지막 비트가 0이 아닌 1인 비트일 때 RBSP 멈춤 비트이며 즉, 끝을 의미하기 때문이다[7].

SPS NAL에는 시퀀스 변수들이 저장되는데, 프로파일 ID나 레벨, 참조 프레임의 개수, 화면의 크기 등의 정보가 복호된다. 이는 InterpretSPS 함수가 비트스트림에서 각 변수 정보를 복호하는 과정에서 진행된다[5].

PPS NAL에는 화면 변수들이 SPS NAL 다음으로 저장되는데, 슬라이스 그룹의 개수, 엔트로피 부호화 방식, 디블록킹 필터 사용 여부, 가중치 예측 방법, 8x8 변환 모드 사용 여부 등을 알려주는 정보가 복호된다. 이는 SPS에서와 마찬가지로, 비슷한 방법으로 InterpretPPS 함수가 비트스트림에서 각 변수 정보를 복호하는 과정에서 진행된다[5].

2.2 H.264 비디오 부호기(인코더) 구조

Fig. 2.는 H.264/AVC-MPEG의 비디오 부호기(인코더) 구조를 나타낸 블록 다이어그램이다[6]. H.264/AVC-MPEG의 인코딩 알고리즘에서 비트스트림을 압축한 인코딩 영상을 얻기 위해서는 Fig. 2.와 같은 과정을 거쳐야 한다.

첫 째, Raw Video의 공간 중복성을 줄이기 위한 화면 내 인트라 예측 코딩(Intra Prediction Coding)을 수행한다[8]. 이 과정은 DCT(Discrete Cosine Transform, 이산 코사인 변환)을 통해 영상의 공간영역을 주파수영역으로 변환한다. 주파수 영역은 높은 주파수대인 DC(Direct Current)와 낮은

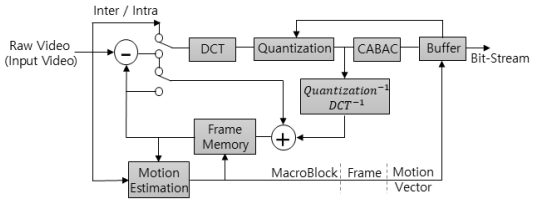

Fig. 2. H.264 Video Encoder Structure. This encoder compresses bit-stream [6].

주파수대인 AC(Alternating Current)으로 나뉘고, 프레임 내에서 서로 인접한 MacroBlock으로 분포하게 된다. 그리고 Quantization(양자화)에 의해 AC와 DC의 표본화 작업을 가지면서 Prediction Mode에 의해 인접해 있는 매크로블록을 통해 참조 예측이 가능할 경우, 가중치를 DC는 낮게, AC는 높게 준다[9]. 이렇게 DCT와 양자화 기법을 이용해, 손실압축을 하게 된다. 이는 이후에 가변 길이 부호화 단계에서 압축효율을 높여준다.

둘 째, Raw Video의 시간 중복성을 줄이기 위한 화면 간 인터 예측 코딩(Inter Prediction Coding)을 수행한다[8]. 연속되는 프레임 간의 중복성을 제거할 수 있도록 모션 예측, 움직임 보상의 개념을 기본으로 하여 이루어진다. 이 단계에서 모션 참조 예측의 범위는 매크로블록, 프레임, 모션벡터로 될 수 있다. 참조되는 예측 과정은 현재 화면의 매크로블록, 프레임, 모션벡터 범위에서 이전 화면의 각각의 범위들과 비교하면서 참조가 이루어진다.

셋 째, 첫 번째 과정과 두 번째 과정을 통해 얻어진 영상의 비트스트림에서 통계적 중복성을 줄이기 위한 엔트로피 코딩(Entropy Coding)을 수행해야 한다[8]. 샤논의 무잡음 소스 부호화 이론에 의하면, 우리가 가능한 한 효율적으로 소스를 부호화할 경우, 평균 비트율은 최소 소스가 갖는 엔트로피와 같거나 그 이상이 된다는 것을 의미한다. 다시 말해, 평균 비트율이 엔트로피에 가까워질수록 압축률이 높다는 것을 의미한다. 즉, 압축률을 높이기 위해 엔트로피 코딩을 통해 무손실 압축을 하고, 이를 통해 영상의 비트스트림에서 통계적 중복성을 줄일 수 있게 되는 것이다.

샘플 영상마다 비트율이 가변적이기 때문에, 비트스트림을 고정된 채널로 전송하기 위해서는 위와 같이 비트율을 제어할 필요가 있다. 따라서 전송 최종단에 버퍼를 두고 버퍼의 상태에 따라서 양자화 과정의 단계 폭을 조절하여 비트율을 제어한다.

2.3 H.264 비디오 복호기(디코더) 구조

Fig. 3.은 H.264/AVC-MPEG의 비디오 복호기(디코더) 구조를 나타낸 블록 다이어그램이다[6]. H.264/AVC-MPEG의 디코딩 알고리즘에서 압축된 비트스트림을 압축 해제한 디코딩 영상을 최종적으로 얻기 위해서는 Fig. 3.과 같은 과정을 거쳐야 한다.

이 과정은 압축된 비트스트림이 복호기의 입력으로 들어오면, 부호기의 역순으로 영상을 복호한다. 역CABAC(Context-based Adaptive Binary Arithmetic Coding) 과정을 거치고 인코딩 때 움직임 예측에 대한 판단 및 보상을 수행한다. 그리고 역양자화와 역DCT룰 한다. 이러한 과정들 이후에는 Frame Memory를 통해 이전 화면에서 저장된 영상과 더해지게 된다. 따라서 디코딩된 영상을 최종적으로 획득할 수 있다.

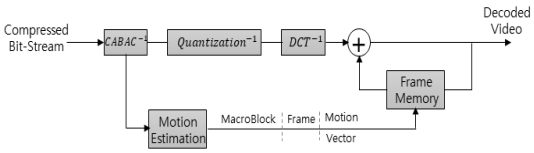

Fig. 3. H.264 Video Decoder Structure. This decoder decompresses bit-stream compressed [6].

2.4 관심영역 범위

영상 속 개인정보를 보호하기 위해 암호화를 진행하려면, 해당 대상의 관심영역 범위를 알아야 한다. 본 연구에서는 관심영역을 얼굴 영역으로 설정하고 해당 얼굴 영역의 매크로블록 좌표 범위를 사전에 직접 알고 암호화를 하였다. 그리고 법원 및 수사기관의 요청이 있을 때 관심영역의 좌표와 키를 관계자에게 알린다고 가정하고 복호화를 진행하는 범위이다.

이는 본 연구의 범위와 기존 연구들의 주안점과 관련이 있다. 본 연구의 범위는 암호화 이후 드리프트를 완화하며 드리프트 해결방법에 대한 전체 영상의 비트레이트 오버헤드를 감소시키는 것이다. 관심영역을 검출하기 위한 방법은 본 연구의 목적, 방향과 다른 부분이 있다. 예를 들어, 일반적으로 적분 영상을 이용해 Haar-like의 특징점 검출[13]을 하는 알고리즘 등의 관심영역(얼굴) 검출 방법은 개인정보를 보호하는 방법과 별개로 검출에 대해 추가 연산량이 증가하게 되는 요인을 발생시킨다.

이는 관련연구[2][9][10]에서도 얼굴 검출 방법을 연구의 범위에 포함시키지 않고, 관심영역의 개인정보를 보호하는 방법과 그로 인해 발생하는 드리프트의 완화 및 비트레이트 오버헤드의 발생에 대한 해결방법에 집중하여 연구하는 흐름에서 알 수 있다. 본 연구에서도 또한 드리프트 완화와 비트레이트 오버헤드 최소화의 해결방법에 집중하며 H.264의 움직임 예측 및 보상 단계의 자체적 특성을 활용해 구현하고, 관련연구[2][9][10]를 재구현하여 비교·분석 하는데 집중하였다. 본 이슈인 드리프트와 비트레이트 오버헤드의 최소화 문제를 해결하고 나서 보다 넓은 범주에서 질적 논문으로 발전시키기 위해, 얼굴 영역과 같은 관심영역의 검출 방법에 대해 향후 연구로 진행할 것이다.

2.5 관심영역 암호화 및 복호화

관련 연구[2]에서는 영상 속 식별이 가능한 개인정보가 담긴 관심영역을 암호화하기 위해, 인코딩 후 실시간 영상 암호화로 네트워크 전송 문제를 줄이는 것에 초점을 두었다.

암호화를 하여 영상 속 개인정보를 보호하더라도, 영상 자체의 인코딩 효율은 유지하는 것이 중요하다. 따라서 영상 데이터 암호화는 인코딩 이후에 하는 것이, H.264/AVC-MPEG 고유의 인코딩 알고리즘에 의한 인코딩 효율성을 보장할 수 있다. 이에 본 연구에서도 인코딩 이후 암호화 및 복호화를 진행한다. 높은 계산 복잡도와 느린 암호화 속도를 가진 AES, DES, RSA 암호화 대신에, 계산 복잡도를 낮추어 빠른 암호화 속도를 가지는 대칭키 암호화 방식인 RC4 스트림 암호화를 채택하였다[2].

RC4 암호화는 대칭키 알고리즘의 특성인 하나의 키를 가지고 평문과 키 값의 랜덤 치환 방식의 빠른 XOR 연산이기에, 실시간으로 영상 스트리밍을 전송하는 네트워크에 적합하다[2]. 또한, H.264/AVC-MPEG의 JM v10.2 코드의 NAL 구조가 비트스트림 형식으로 이루어져 있기 때문에, 대칭키를 키 스트림으로 암호화와 복호화를 수행하는 것이 알맞다. 이와 같이 H.264 영상의 인코딩 포맷 형식을 준수한다. 이렇게 RC4 스트림 암호화를 진행하고, 키 스트림을 인증기관에 전달 후

법원 및 수사기관의 요청이 있을 시 인증기관에서 디코딩 관계자에게 키 스트림을 전달하여 키 스트림과 암호화된 스트림 영역을 XOR 연산을 하면, 암호화 되었던 관심영역에 대해 복호화가 가능하며 원본 영상을 획득할 수 있다[2].

2.6 드리프트 (Drift)

인코딩 이후 암호화를 수행하게 되면 드리프트가 발생한다. H.264 영상의 프레임마다 서로 참조하는 특성 때문에 암호화된 영역이 그 외의 영역으로 점점 번지게 되는 것이 드리프트이다. 즉, 해상도가 흐려지거나 깨지는 왜곡 현상이나 잡음과 같은 현상으로 볼 수 있다.

드리프트는 공간상 드리프트(Spatial Drift)와 시간상 드리프트(Temporal Drift)로 나뉜다. 공간상 드리프트는 화면 내 영상 참조에 의해 발생한 것이고, 시간상 드리프트는 화면 간 영상 참조에 의해 발생한 것을 말한다[2].

III. 관련연구

H.264 영상 기반에서 본 연구와 가장 관련이 깊은 최신 연구는 2016년도에 WASET에 게재된 "H.264 Video Privacy Protection Method Using Regions of Interest Encryption" 이라는 논문이다[2]. 관심영역 암호화를 수행하고 암호화로 인해, 관심영역 외로 발생되는 공간상 드리프트, 시간상 드리프트 문제와 비트레이트 오버헤드를 완화시킨 방법을 다룬 논문으로써, 본 연구와 비교·분석하기에 가장 적합한 연구이다. 따라서 본 논문의 연구 또한, 제안하는 방법을 제외하고 객관적인 비교·분석을 위해 최신 관련 연구[2]의 실험 과정과 동일하게 시간상 드리프트가 발생한 기준까지 같은 조건으로 실험하였다. 본문에서는 서로 다른 대조군들과 제안하는 실험군으로 재구현하고 비교하여 각각의 비트레이트 오버헤드 발생률에 대한 차이점의 분석 결과를 도출할 것이다.

3.1 시간상 드리프트 해결 방법

H.264의 움직임 예측 및 보상 특성상, 각 프레임마다 화면 간 참조 방식에 의해 관심영역 암호화 후

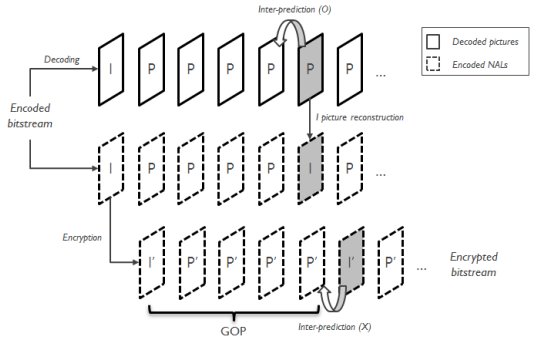

Fig. 4. Re-Insertion method of encrypted I picture(IDR) per 1st, 15th, 30th picture sequence in related research [2].

공간상 드리프트와 시간상 드리프트가 발생하였다. 관련연구[2]에서는 공간상 드리프트는 FMO 방식으로 완화를 시켰으며 이때의 비트레이트 오버헤드 발생율은 저조하였다.

반면, 시간상 드리프트를 완화시키기 위해 인코딩 이후에 I 픽처를 암호화를 하고, Fig. 4.와 같이 암호화한 I 픽처를 1번째 또는 15번째 또는 30번째마다 재삽입하는 방법을 제안하며 시간상 드리프트를 완화하는 실험을 하였다[2]. 그 결과 시간상 드리프트는 완화하였지만, 암호화한 I 픽처를 일정 주기마다 재삽입하기 때문에 영상 전체의 비트레이트는 상당 부분 증가하게 되는 문제점이 발생하였다.

3.2 관련연구의 Bit-rate Overhead 결과

시간상 드리프트를 완화시키기 위해, 관심영역을 암호화한 I 픽처를 1번째 또는 15번째 또는 30번째마다 재삽입한 실험의 영상 전체의 Bit-rate Overhead 결과는 Table 1.과 같다[2].

관련연구[2]의 비트레이트 오버헤드 실험결과는

Table 1. The results of Re-Insertion method of encrypted I picture(IDR) per 1st, 15th, 30th picture sequence in related research [2].

I picture period	Average bit rate	Bit rate overhead	Encoding time increase
no applied	146.94 Kbit/s	0%	0
I picture / 30 frames	161.87 Kbit/s	5.42%	+1.56%
I picture / 15 frames	180.53 Kbit/s	11.56%	+5.37%

샘플 영상과 암호화하는 관심영역의 범위와 H.264의 인코딩 방식 등에 따라 실험 때마다 다르게 나올 수 있다. 관련연구[2]에서의 실험결과는 I 픽처를 삽입하는 픽처 순서에 따라 참조 되어지는 영역의 범위가 다르기 때문에, 15번째에 재삽입시 11.56% 증가하였고 30번째에 재삽입시 5.42%가 증가하였다. 실시간 영상 암호화 후, 영상 스트리밍을 네트워크로 전송할 때 비트레이트 오버헤드가 이정도로 크게 발생하는 것은 전송 시 버퍼 공간의 부족과 추가 시간 소요 등에 많은 요인을 초래할 수 있다. 또한, 재삽입하는 픽처의 순서에 따라 비트레이트 오버헤드 발생율이 일정하지 않고 불규칙하게 발생하는 것은 시스템 자체의 안전성을 보장할 수 없고 위험을 발생시킬 가능성도 존재한다.

따라서 본 논문에서는 관련연구[2]에서 시간상 드리프트를 완화시키기 위하여 제안한 암호화된 I 픽처를 재삽입하는 방법 대신, 다른 제안방법으로 시간상 드리프트와 비트레이트 오버헤드를 줄이고 실시간 영상 스트리밍 전송 시스템에 안정성과 버퍼 공간을 확보할 수 있는 인코딩 방법을 제안한다.

IV. 제안방법

4.1 제안하는 실시간 영상 스트리밍 전송 시나리오

Fig. 5.에서 파란색 블록 영역이 본 논문에서 제안하는 시나리오로, 비트레이트 오버헤드 발생과 관련이 없는 복호화는 제외하고 네트워크를 벗어난 로컬에서 비트레이트 오버헤드 발생과 관련이 있는 암호화만 한 상태에서 제안방법의 실험을 진행하였다.

제안방법은 각 픽처마다 참조하는 움직임 예측 특성에 대해 관심영역 내에서 Block과 Frame의 탐색영역을 제한시키는 인코딩을 한다. 그리고 이후에 사전에 알고 있는 관심영역의 좌표에 대해 RC4 스트림 암호화를 하고 네트워크를 통해 암호화된 영상의 비트스트림을 전송한다. 그리고 움직임 판단 및 보상 특성에 대해 동일하게 Block과 Frame의 참조 탐색 영역을 제한하는 디코딩을 한다.

본 연구는 암호화로 인한 드리프트와 관련연구 대비 드리프트 해결방법의 추가에 따라 연산량이 증가된 비트레이트 오버헤드의 최소화가 목적이다. 따라서 네트워크 이전 단계인 로컬 단계에서 제안방법을 추가하여 암호화만 하고 관련연구 대비 드리프트와 비트레이트 오버헤드의 변화를 비교·분석

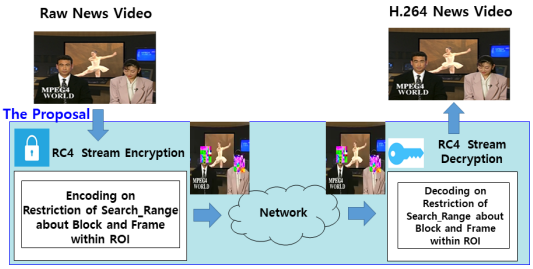

Fig. 5. The Scenario of real-time video streaming transmission to the network with proposed method encrypting ROI(Region of Interest) after the encoding of restriction of search range of block and frame within ROI about the reference of motion estimation per each picture.

한다. 복호화는 원본 영상만 획득하는 것이기 때문에 영상의 비트레이트 오버헤드 발생과는 관련이 없기에, 비트레이트 오버헤드를 최소화하려는 본 연구의 목적에 따라 진행을 하지 않는다. 그러나 법원 및 수사기관 등의 원본 영상 제출의 요청이 있을 시, 로컬 단계에서 암호화를 할 때 사용한 키 스트림과 사전에 알고 있는 암호화가 된 관심영역의 매크로블록 좌표 범위의 비트스트림을 XOR 연산하면, 본래 암호화가 되지 않은 관심영역을 가진 원본 영상을 획득할 수 있다.

이와 같은 시나리오를 통해, 본 연구의 실험 결과는 암호화 때문에 원본 영상보다는 비트레이트 오버헤드가 증가하겠으나 소폭 증가하도록 하며, 관련연구[2]의 비트레이트 오버헤드 보다는 최대한으로 감소시키는 방향을 추구하고자 한다.

4.2 관심영역 암호화 이후 드리프트(공간상, 시간상)

드리프트를 완화하기 위한 기법을 어떠한 것도 적용하지 않고, 인코딩 이후에 H.264 영상의 관심영역의 비트스트림을 RC4 암호화를 하게 되면 Fig. 6.처럼 공간상 드리프트와 시간상 드리프트가 함께 발생하게 된다.

이어서 공간상 드리프트를 완화하기 위해 본 연구와 관련연구[2]는 FMO를 적용하고, 시간상 드리프트를 완화하기 위해 본 연구와 관련연구[2]는 서로 다른 방법을 적용한 실험을 진행한다. 그리고 시간상 드리프트를 완화할 때, 서로 다른 방법에 의해 발생되는 비트레이트 오버헤드를 비교·분석하는 과정의 실험결과를 도출한다.

Fig. 6. The drift (including both spatial and temporal drift) caused by the motion estimation of the ROI encrypted after encoding is occurred to spread to the non-ROI.

4.3 FMO 적용 이후 공간상 드리프트 완화 재구현

Fig. 7.은 관련연구[2]와 같이 원본영상의 관심영역을 암호화한 후에, 단일 화면 내 공간상 드리프트를 완화한 모습을 보여준다. 두 명의 리포터의 얼굴을 관심영역으로 하여 슬라이스 그룹을 2개로 분리하여 나누고, 이외의 배경화면을 비관심영역으로 하여 1개의 슬라이스 그룹으로 분리하여 나눈다. 이렇게 암호화하기 전, 단일 화면 내에서 총 3개의 슬라이스 그룹으로 독립적으로 분리하여 인코딩 과정에 적용한 것을 FMO 방식이라 한다[2][9][10]. FMO 방식을 적용하여 단일 화면 내에서 공간상 드리프트를 해결할 수 있으며 비트레이트 오버헤드 발생율도 원본 영상과 대비 시, 거의 없다고 볼 수 있다.

이후, 시간상 드리프트와 비트레이트 오버헤드를 완화시키는 과정의 해결방법을 객관적으로 비교하기 위해, 공간상 드리프트를 완화하는 방법을 관련연구[2][9][10]과 동일하게 적용하여 재구현하였다.

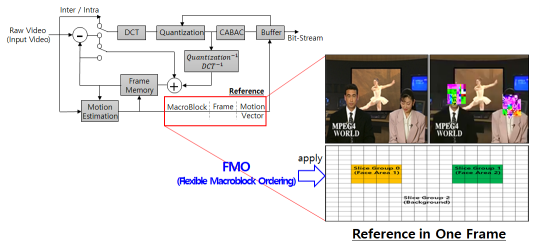

Fig. 7. FMO is applied to mitigate the spatial drift occurred in the encoder process by re-Implementation based on the code of JM v10.2 similar as related research [2][9][10].

4.4 'H.264 영상 개인정보 보호' 전체 시스템 구조도

Fig. 8.은 제안하는 실험과정이 모두 포함된 'H.264 영상 개인정보 보호'을 위한 전체 시스템 구조도이다. 본 구조도에는 Part.1,2,3 과정으로 이루어져 있으며, Part.1은 인코더 구조, Part.2는 인코딩 후의 암호화 구조, Part.3는 디코더 구조이다. Fig. 7.과 같이 인코딩 과정에 FMO 기법이 적용되고 화면 내 공간상 드리프트가 완화되고, 화면 간 시간상 드리프트와 비트레이트 오버헤드를 최소화하기 위해 만들어진 전체 시스템 구조도이다.

H.264의 비트레이트 종류에는 고정 비트레이트와 가변 비트레이트로 구성할 수 있다. 실시간 영상 스트리밍을 전송할 때, 영상데이터 보안을 하려면 본연의 원본 영상의 일정 품질을 유지하는 것이 가장 중요하다. 따라서 H.264의 표준 압축효율에 의해 정해지는 고정 비트레이트는 표준 알고리즘의 효율성을 저해하지 않기 위해 원래의 방식대로 손실 압축(DCT, 양자화)과 무손실 압축(CABAC)을 진행한다. 가변 비트레이트는 H.264의 움직임 예측 및 보상 특성에 따라, 다음 프레임에서 움직임이 발생할 시에 예측 및 보상의 참조가 일어나 가변적으로 비트레이트가 일어나는 것을 말한다[12]. 따라서 영상의 가변 비트레이트를 조율하려면, Encoder_Part.1 구조에서부터 변경을 진행해야 한다.

이에 따라 압축률을 제어하는 구간을 제외하고 가변 비트레이트 변화가 일어날 수 있는 구간으로, 움직임 예측 및 보상이 일어나는 Motion Estimation에서 화면 간 참조 부분(Fig. 8.의 Reference; MacroBlock, Frame, Motion Vector)을 변경한다. 화면 간에서 관심영역의 참조 탐색 영역을 Block과 Frame에 대해 제한함으로써, 비관심영역과 독립적으로 분리시킨다. 이는 인코딩 이후에 관심영역을 암호화하고 나서, 암호화된 관심영역이 관심영역 내에서와 비관심영역으로 불필요하게 참조되는 현상을 막게 된다.

관심영역의 Block과 Frame 참조 탐색 영역을 제외한 이외의 영역에 대한 움직임 예측이 진행되고, 해당 Frame들은 Frame Memory 단계를 통해 저장되는 과정을 가진다. 이렇게 영상의 가변 비트레이트가 결정되면서 Buffer 공간에 고정 비트레이트와 함께 할당되고 최종적으로 손실/무손실 압축이 된 H.264 영상의 비트스트림이 만들어진다.

다음은 인코딩 이후의 관심영역에 대한 RC4 스트림 암호화 구조 단계(Encryption_Part.2)이다. 여기서의 암호화하는 조건 기준은 영상의 각 프레임마다 매크로블록을 확인하면서, 해당 매크로블록이 관심영역의 슬라이스 그룹(남성리포터의 얼굴: 0,

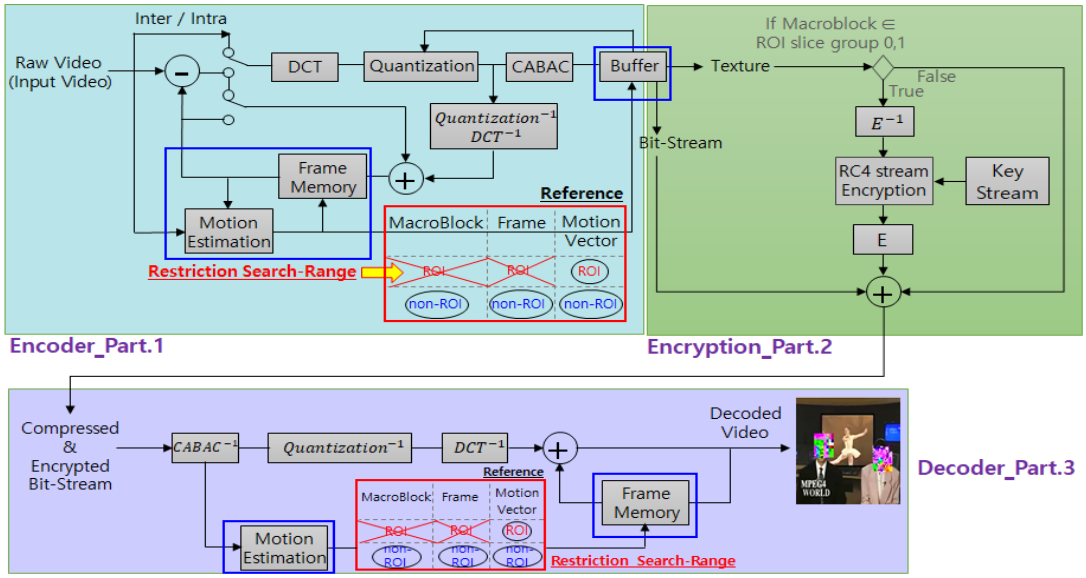

Fig. 8. The overall system structure of the proposed 'H.264 video privacy data protection' (Part.1: Encoding process, Part.2: ROI encryption process, Part.3: Decoding process)

여성리포터의 얼굴: 1)에 포함되어 있는 지에 대한 여부이다. 포함되어 있다면 평문과 키 스트림 값의 XOR연산을 통한 RC4 암호화를 하고, 포함되어 있지 않다면 암호화를 하지 않으면서 순차적으로 영상의 비트스트림들을 더해 나간다[2].

암호화 구조 단계를 거치고 나면, 그 다음은 디코더 구조 단계(Decoder_Part.3)이다. 디코더 구조 단계는 인코더 구조 단계의 역순이다. H.264의 디코더 표준 알고리즘에 의해 CABAC의 무손실 압축 해제를 하고, 인코딩 단계에서 움직임이 예측되었던 부분들에 대해 관심영역의 Block과 Frame 참조 탐색 영역이 제한된 상태에서 움직임 판단 및 보상이 일어난다. 그리고 역양자화, 역DCT를 통해 손실 압축해제를 한 후, IPPP 타입으로 프레임들을 더해 나감으로써 관심영역이 암호화된 H.264 News 영상을 획득한다.

원본영상에 대해서는 법원과 같은 특정 기관의 요청이 있을 시, 키 스트림으로 암호화된 관심영역의 복호화를 하고 이를 통해 원본영상의 획득이 가능하다.

4.5 제안하는 시간상 드리프트 완화 방법

관련연구[2]에서 비트레이트 오버헤드가 증가하게 된 원인인, 움직임 예측 및 보상을 위한 참조는 H.264의 표준 인코딩 방식에서 큰 장점 중 하나이다. 영상에서 모션에 대한 예측을 할 수 있고 보정을 할 수 있기 때문이다.

그러나 영상데이터 보안을 위해 암호화를 한 영역에 대해서는 이전 프레임의 참조가 불필요한 부분일 뿐이다. 이유는 관심영역이 암호화로 랜덤 치환되어 픽셀 값이 흐트러져 있기 때문에, 현재 프레임에서 이전 프레임의 해당 관심영역을 다시 참조할 필요가 없는 것이다.

참조하게 될 시, 랜덤 치환된 픽셀 값이 이동하는 모션 구간을 따라 계속 참조가 이루어져 해당 관심영역 외의 바깥 영역의 픽셀 값도 참조에 의해 점점 흐트러져 번지는 것처럼 보이는 현상이 일어나게 된다. 이 부분이 시간상 드리프트라고 앞서 밝힌 내용이다.

Fig. 9.는 시간상 드리프트를 완화시키면서 비트레이트 오버헤드를 감소시키기 위해, 본 논문에서 제안하는 핵심을 잘 나타내고 있는 매크로블록 단위의 참조 구조 그림이다.

이 움직임 예측 및 보상에 대한 참조 구조의 그림은 먼저, 이전 프레임과 현재 프레임을 표현하였다. 둘째로, 2개의 프레임에 FMO의 슬라이스 그룹별 분리처럼, 각각 ROI(관심영역) 2개와 non-ROI(비관심영역) 1개로 나타냈다. 셋째로, ROI 2개와 non-ROI 1개는 슬라이스 그룹 0,1,2인 총 3개의 슬라이스 그룹으로 분리하였다. 넷째로, 2개의 ROI는 이전 프레임에 대한 Block, Frame의 참조 탐색

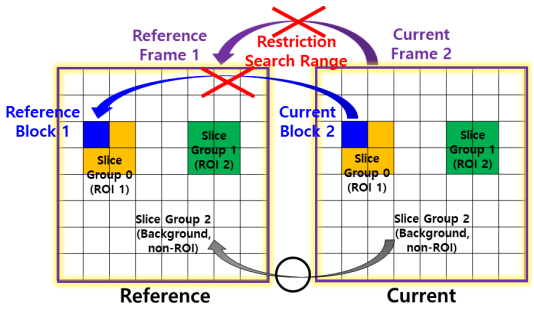

Fig. 9. The proposed method to mitigate temporal drift. This method restricts search range of block, frame about the reference [4][12].

영역을 제한하였다. 그리고 1개의 non-ROI는 이전 프레임에 대한 참조 탐색 영역을 본래의 인코딩 특성대로 모두 허용하였다.

본 제안 방법에 의하여, 인코딩 후 암호화된 I픽처의 관심영역은 모든 프레임마다 해당 관심영역에 대해, 이전-현재 프레임 간에 관심영역과 비관심영역과의 Block, Frame 탐색 범위의 재참조가 일어나지 않게 된다.

V. 제안방법의 실험결과

5.1 각각의 실험방법에 따른 드리프트 결과

동일한 환경에서 암호화를 하지 않은 원본영상과 암호화만 진행한 영상, FMO 기법을 적용한 관련연구[9][10]의 영상과 암호화된 I픽처를 특정 주기로 재삽입하는 최신 관련연구[2]의 영상으로 총 4개의 대조군들과 Fig. 9.와 같은 본 제안방법의 실험군을 구현하여, 시간상 드리프트 완화에 대한 실험을 비교·분석 하였다.

샘플 영상은 2명의 리포터(관심영역 2개)가 나오는 News 영상을 사용하였고 QCIF 해상도를 가지는 본 영상의 크기는 176x144, 관심영역의 크기는 2개의 32x32, 비디오 시퀀스 타입은 IPPP 형식으로 관련연구[2]와 동일한 조건으로 구성하였다.

인코딩, 암호화, 디코딩 과정을 통해, 생성된 프레임은 총 100 프레임이었고, 이를 인코딩 때 모든 프레임에 대한 비트를 나열하면 약 200프레임 정도가 되었다. 여기서 관련연구[2]의 방법대로 30번째마다 암호화된 I픽처를 재삽입 하였는데, 30번째, 60번째, 90번째마다 시간상 드리프트가 원래 처음의 I픽처대로 완화되었지만 이때마다 비트의 주기는 암호화하였던 IDR 픽처만큼 굉장히 크게 치솟게 되었다. 따라서 Fig. 10.은 관련연구[2]에서 시간상 드리프트가 크게 발생하는 지점으로 재삽입 주기의 바로 직전 지점인, 29프레임, 59프레임과 마지막 프레임인 99프레임으로 구성하였다. 59프레임의 경우, 비관심영역에 있는 발레리나의 움직임 범위가 암호화된 관심영역과 겹치게 되면서 나타난 시간상 드리프트는 각각의 모든 실험 방법에서 크게 발생하였다.

Fig. 10.은 좌측부터 드리프트가 없던 원본 영상에서 관심영역을 암호화할 시 발생한 공간상/시간상

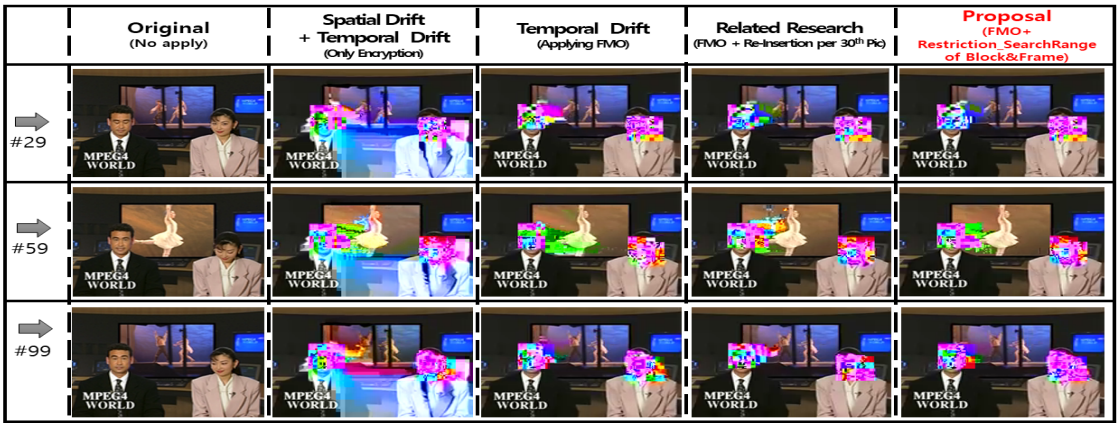

Fig. 10. The analysis of the comparison for the difference of the spatial and temporal drift of each method according to Original(no apply), Spatial Drift and Temporal Drift(only encryption), Temporal Drift(applying FMO)[9][10], Related Research(FMO + re-insertion per 30^{th} picture)[2], Proposal(FMO + restriction search range of block&frame) using News sample of H.264 video.

드리프트 영상을 보여주고 있다. 그리고 이어서 관련 연구[8]의 FMO 기법을 적용하면서 공간상 드리프트를 완화하고, 남아있는 시간상 드리프트에 대해 관련 연구[9][10]의 FMO 기법을 최신 관련연구[2]과 제안방법에도 동일하게 적용하여 실험결과를 도출하였다. 이는 최신 관련연구[2]과 제안방법에 FMO 적용을 통해, 비트레이트 오버헤드 발생 원인이 아닌 공간상 드리프트를 동일하게 완화한 후, 연이어 발생하는 시간상 드리프트의 변화를 최신 관련연구[2]의 방법과 제안방법의 실험 결과에 대해 객관적으로 비교하기 위함이다. 여기서 시간상 드리프트를 완화하기 위해 사용된 각각의 기법들이 가변 비트레이트를 높이게 되었다.

실험 결과, 공간상 드리프트를 완화하는 데 화면 내 적용된 FMO기법[9][10]은 두드러지게 효과적이었다. 그리고 나타나는 문제는 화면 간 시간상 드리프트 발생이었다. 이는 JM v10.2 코드에서 인코더의 암호화된 관심영역과 비관심영역 사이의 움직임 예측과 해당영역에 대한 디코더의 움직임 판단 및 보상 때문에 일어나는 이슈였다.

본 이슈는 Fig. 10.에서 Spatial+Temporal Drift(Only Encryption)와 Temporal Drift[9][10]와 Related Research[2]와 Proposal의 59프레임을 보면 알 수 있다. 59프레임의 비관심영역에 있는 발레리나의 움직임 크기, 범위, 관심영역과의 겹쳐지는 범위가 29프레임과 99프레임에 비해 확연히 다르고 크다. 따라서 시간상 드리프트의 크기 또한 59프레임에서 크게 발생하였다.

반면에, 전체 프레임에서 관심영역과 겹치는 비관심영역의 움직임이 가장 컸던 59프레임을 보면 제안방법이 다른 관련연구[2][9][10] 대비 시간상 드리프트가 발생되는 범위가 시각적으로 줄어드는 것을 알 수 있다. Fig. 10.의 드리프트 변화의 시각적인 결론은 움직임 예측 및 보상 특성을 고려하지 않은 채, 움직임이 발생되는 영역과 암호화된 관심영역이 겹치게 되는 프레임이 존재하면 참조에 의해 화면 간의 시간상 드리프트는 존재할 것이라는 점이다. 이 결과에 따라, 제안방법의 관심영역에 대한 Block, Frame 간 움직임 참조 탐색 영역 제한은 시간상 드리프트를 완화하는데 효과적이라는 것을 확인하였다.

5.2 PSNR(국제표준 화질척도 기준) 결과

Table 2.는 Fig. 10.에서 각각의 방법에 대한 PSNR(Peak Signal-to-Noise Ratio, 최대 신호대 잡음비, 단위: dB) 화질 척도 값이다. PSNR은 국제표준 화질척도 기준으로, 두 영상간의 왜곡률에 대한 화질 비교하여 차이 잡음에 대한 값을 구한 것을 말한다[11]. PSNR을 구하는 과정은 다음과 같다.

Table 2. The difference result of the resolution of PSNR by each method using News video as the sample video.

(Unit: dB)

Experiment Type	PSNR Y	PSNR U	PSNR V
[Original] No applied	36.47	39.72	40.23
[Spatial+Temporal Drift] Only Encryption	7.65	10.86	11.27
[Temporal Drift] Applying FMO [9][10]	7.78	11.04	12.18
[Related Research] Re-Insertion per 30th Picture [2]	8.12	11.70	12.70
[Proposal] Restriction Search-Range of Block & Frame within ROI	8.23	11.87	12.88
[PSNR Difference] The Proposal vs Related Research	0.11	0.17	0.18

$$MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \qquad (1)$$

(1)의 공식은 PSNR을 구하기 전, MSE이라는 평균 제곱 오차를 구하는 식이다[11]. 이는 다음의 PSNR 공식에서 분모 값으로 들어간다. (1)의 공식에서 m과 n은 원본영상과 실험군들, 두 개의 이미지 프레임 모두 QCIF 크기가 176x144 이므로, 각각 176, 144의 값을 가진다. 그리고 서로 비교하는 대상의 두 이미지 프레임들은 흑백으로 비교를 한다. 원본과 실험영상의 왜곡률에 대한 비교를 할 때, 왜곡률 판단의 정확성을 위함이다. I(i, j)는 원본영상에서 176x144 크기의 영상 전체 좌표(i, j)의 픽셀 값이며, K(i, j)는 동일 조건에서 비교 영상에서의 영상 전체 좌표(i, j)의 픽셀 값이다[11][12].

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) = 20 \cdot \log_{10}(255) - 10\log_{10}(MSE_{(176,144)}) \quad (2)$$

(2)의 공식이 PSNR을 구하는 전체 공식이다〔11〕. 원본영상과 각각 다른 방법의 영상들을 비교할 때, 두 이미지 프레임들 간의 색상 채널은 모두 그레이 스케일이므로 MAX_I는 이미지의 가능한 최대 픽셀 값으로 255이다. (2)의 PSNR 계산 과정 중, 마지막 식인 $20\log_{10}(255) - 10\log_{10}(MSE_{(176,144)})$의 $MSE_{(176,144)}$에서 176x144 크기의 각각의 영상 매트릭스 픽셀 값을 대입하여 MSE를 계산해준다〔11〕〔12〕. 이와 같은 과정으로, Table 2.와 같이 각각의 서로 다른 방법들에 대한 최종 PSNR 값들을 도출해낼 수 있다.

Table 2.에서 모든 PSNR의 기준 척도는 원본영상이다. 원본영상과 왜곡률을 대비하여, 발생한 잡음비에 대한 계산된 값이다. Table 2.를 보면 원본영상의 PSNR 값은 36.47~40.23 dB로 높고, 나머지 영상들의 PSNR 값은 7.65~12.88 dB로 굉장히 낮다. 원본영상을 제외하고 나머지 영상들은 관심영역 암호화를 하고 각각의 방법들을 적용하면서 원본영상 대비 PSNR 값이 낮게 측정되었다. 국제표준에서 PSNR 값이 낮다는 것은 왜곡률이 높다는 것을 의미한다〔11〕. 본 실험 결과 또한 그러하다.

관심영역 암호화를 통해 드리프트가 많이 발생할수록 PSNR 값은 더 낮게 떨어졌다. 즉, Fig. 10.에서 움직임 예측 및 보상의 특성에 의해 참조를 많이 할수록 공간상/시간상 드리프트 발생 범위가 증가하는데, 이에 따라 반비례적으로 PSNR 값은 낮아진 것이다. 결과적으로, Table 2.의 PSNR 값을 비교·분석 해보면 제안방법의 PSNR 값이 제일 높으므로 종합적인 드리프트 발생률이 줄어든 것을 수치적으로 알 수 있다. 그리고 FMO 적용 후부터는 공간상 드리프트가 완화되면서, 시간상 드리프트의 발생이 이슈였다.

결론적으로, 움직임 예측 및 판단 대비 프레임별 시각적인 결과는 관련연구〔2〕보다 제안방법이 시간상 드리프트를 완화하였고 PSNR 수치값도 증가하였기에 이를 뒷받침 할 수 있는 객관적인 결과표라고 할 수 있다. 또한, 최신 관련연구〔2〕과 제안방법의 PSNR 차이 값이 0.11~0.18 dB로 굉장히 작다. 그러므로 제안방법은 다른 방법을 사용한 관련연구〔2〕와 시각적으로 구분하기 힘든 미세한 차이값을 가지며 시간상 드리프트를 비슷하게 완화하였고, 다음으로 이에 따른 비트레이트 오버헤드의 결과에 대한 분석이 중요하다.

5.3 영상 전체의 비트레이트 오버헤드 결과

Fig. 10.과 Table. 2를 통해, 제안방법에 의한 시간상 드리프트는 관련연구〔2〕 만큼 비슷하게 완화한 것을 확인하였다. 여기서 시간상 드리프트는 서로 다른 방식에 의해 완화가 되었지만, 제안방법을 적용하였을 때 영상의 비트레이트 오버헤드는 관련연구〔2〕보다 어느 정도 최소화를 시켜주는 지가 관건이었다.

원본영상, 암호화 적용(공간상/시간상 드리프트 발생), FMO 적용(시간상 드리프트 발생)〔9〕〔10〕, FMO적용 후 제안방법(시간상 드리프트 완화), FMO적용 후 암호화한 I픽처 30번째마다 재삽입하는 관련연구(시간상 드리프트 완화)〔2〕의 비트레이트, 오버헤드, 인코딩 시간에 대한 결과 분석표는 Table 3.과 같다. 원본영상을 기준으로, 원본영상의 비트레이트 오버헤드와 인코딩 시간 결과를 0%로 설정하였다. 그리고 서로 다른 방법의 증가한 비트 수와 인코딩 시간을 원본영상 대비 비율로 계산하였다.

실험 결과, 제안방법의 가변 비트레이트는 69.74 kbit/s, 총 비트수 464952 bits로 원본영상 대비 오버헤드가 2.35% 증가하였고, 인코딩 시간은 총 32.561 sec로 41.36% 증가하였다. 암호화만 적용한 후에는 공간상/시간상 드리프트 발생으로 인해, 가변 비트레이트 오버헤드 발생율이 1.24%로 미약하게 증가하였다. 이에 FMO〔9〕〔10〕 적용 후에는 공간상 드리프트를 완화하였지만, FMO에 대한 연산량 증가로 원본영상 대비 비트레이트 오버헤드가 1.91%로 조금 더 미약하게 증가하게 되었다. 관련연구〔2〕는 암호화된 IDR 픽처 재삽입으로 인해, 17.28% 라는 높은 수치의 오버헤드가 발생하였다. 제안방법과 최신 관련연구〔2〕 모두 시간상 드리프트 완화를 위한 추가 방법의 적용으로 인해 추가 연산량이 늘어나면서 비트레이트 오버헤드가 더 증가하게 되었다.

Table 3. The difference result of bit-rate, overhead, and encoding time by each method using News sample video.

Experiment Type	Bit rate	Overhead	Encoding time
〔Original〕 No applied	69.64 kbit/s	0 % Total bits:454248 bits (I 32736, P 421304, NVB 208)	0 %
〔Spatial +Temporal Drift〕 Only Encryption	69.69 kbit/s	+ 1.24 % Total bits:459916 bits (I 96280, P 363428, NVB 208)	+ 40.87 % 32.448 sec (3.08 fps)
〔Temporal Drift〕 Applying FMO 〔9〕〔10〕	69.72 kbit/s	+ 1.91 % Total bits:462944 bits (I 63672, P 399064, NVB 208)	+ 41.06 % 32.492 sec (3.08 fps)
〔Proposal〕 Restriction Search-Range of Block & Frame within ROI	69.74 kbit/s	+ 2.35 % Total bits:464952 bits (I 32736, P 432008, NVB 208)	+ 41.36 % 32.561 sec (3.07 fps)
〔Related Research〕 Re-Insertion per 30th Picture 〔2〕	79.76 kbit/s	+ 17.28 % Total bits:532760 bits (I 128192, P 404360, NVB 208)	+ 41.57 % 32.610 sec (3.07 fps)
〔Difference〕 The Proposal vs Related Research	+ 10.02 kbit/s	+ 14.93 %	+ 0.21 %

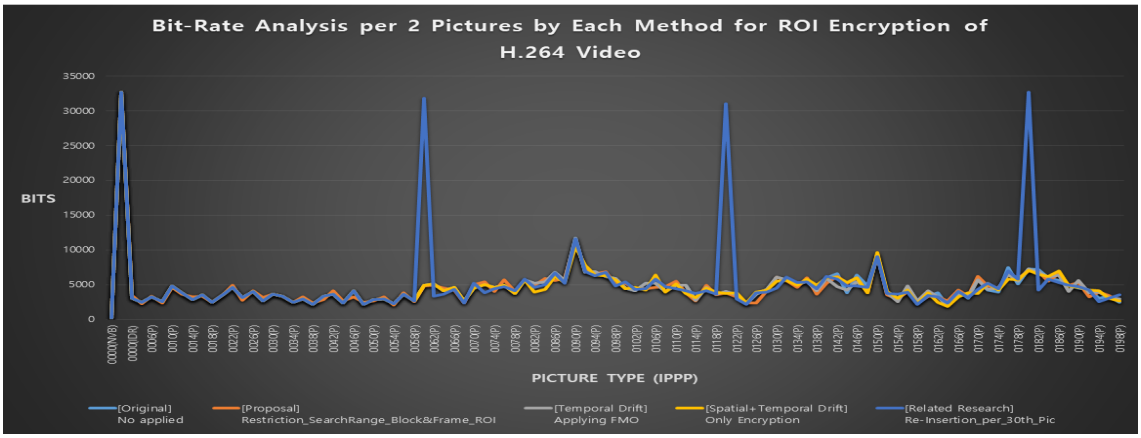

Fig. 11. The analysis of bit-rate overhead per 2 frames by each method for ROI encryption of H.264 video.

결과적으로, 제안방법은 기존에 관심영역과 비관심영역이 경계 없이 본래의 움직임 예측 및 참조 방식에서 관심영역은 Block과 Frame의 참조 탐색 영역을 제한시키고 비관심영역은 제한하지 않는 방법으로 분리하여 추가 연산이 되었기 때문에 가변 비트레이트 오버헤드가 원본영상보다 2.35%, 암호화 적용한 영상보다 1.11%, FMO적용한 관련연구〔9〕〔10〕보다 0.44%로 미약하게 증가하면서, 관련연구〔2〕와 14.93% 차이로 대폭 감소시키는 향상된 결과를 보였다.

VI. 결 론

본 논문에서는 관심영역 암호화 시 발생하는 H.264 영상의 비트레이트 오버헤드를 감소시키기 위한 방법에 대하여 제안하였다. 제안방법에서는 H.264/AVC-MPEG의 인코더와 디코더에서 수행되는 움직임 예측 및 보상의 표준 특성에 대해, 관심영역 내에서 Block과 Frame의 참조 탐색 영역을 제한시켰다. 이에 따라, 시간상 드리프트를 완화하며 관심영역 암호화 후 발생하는 가변 비트레이트 오버

헤드가 관련연구[2] 대비 14.93% 감소하고 원본영상 대비 2.35% 증가하였고, 인코딩 시간은 관련연구[2]보다 0.21% 감소하였다.

이에 대한 비트레이트 오버헤드 전체 변화율은 서로 다른 방법으로 실험하고 모든 프레임들을 그래프로 나열하여 나타낸 Fig. 11.을 통해 객관적인 변화 결과를 확인할 수 있다. 본 변화 그래프에는 IPPP 비디오 시퀀스 타입으로 이루어진 모든 프레임들의 비트를 2개의 프레임마다 나타냈다. 관련연구[2]만 해당 연구의 IDR 픽처 재삽입으로 인해, 재삽입 주기마다 비트레이트 오버헤드가 IDR 픽처의 비트 크기만큼 높게 발생하였고, 본 연구의 제안방법은 원본영상과 FMO만 적용한 관련연구[9][10]와 암호화만 진행한 영상과 비슷한 그래프 변화로 낮게 발생하였다.

따라서 관련연구[2] 대비 영상 전체의 비트레이트 오버헤드가 최소화된 제안방법은 본래의 H.264의 인코딩 효율을 유지하며, 관심영역의 암호화를 통해 개인정보를 보호하고 고화질의 H.264 영상 전송 시 충분한 비트스트림의 버퍼 공간 확보가 가능하며 실시간 스트리밍 시스템 안정성에 기여할 수 있다.

향후 연구에서는 관심영역의 검출 방법을 포함한 암호화와 복호화를 하고 Stefan(테니스 선수의 경기), Hall(정적인 복도에서 사람들 지나가는 영상) 등과 같이 움직임이 많은 샘플 영상 기반에서, 제안방법과 모션벡터의 참조[4][6][12] 제한과 비관심영역과의 독립적 분리에 추가 연구를 진행할 것이다. 이는 움직임이 많은 영상 샘플로 실험 범위를 넓혀 참조 제한에 대한 객관성을 더 높일 수 있을 것이며, 추가 연구를 통해 시간상 드리프트를 더욱 완화하면서 비트레이트 오버헤드를 현재보다 줄여나갈 수 있기를 기대한다.

References

[1] Hyunchan Moon, "Intelligent CCTV technology and market trend," Institute for Information Technology Advancement, Weekly Technology Trends no.1361, Aug. 2008.

[2] Taekyun Doo, Cheongmin Ji and Manpyo Hong, "H.264 Video Privacy Protection Method Using Regions of Interest Encryption," World Academy of Science, Engineering and Technology, International Journal of Computer and Information Engineering, Vol. 10, No. 11, 2016.

[3] Yun Seong Ko, Kwang Hyuk Park and Chang Soo Kim, "Problem Analysis and Countermeasures Research through Security Threat Cases of Physical Security Control Systems," Journal of Korea Multimedia Society, Vol. 19, No. 1, pp. 51-59, 2016.

[4] Alexis Michael Tourapis and Athanasios Leontaris, "JM Reference Software Manual (JVT-AE010)," H.264/14496-10 AVC Reference Software Manual, Joint Video Team (JVT), London, UK, Jul. 2009.

[5] Yoseong Ho and Seunghwan Kim, "H.264 / AVC algorithm understanding and program analysis," Dooyangsa, ISBN: 978-89-7528-225-6 / 93560, Sep. 2008.

[6] Yoseong Ho and Seunghwan Kim, "The source code analysis of H.264/AVC standard," Dooyangsa, ISBN: 89-7528-153-1 / 93560, Jul. 2006.

[7] Yoseong Ho, "Understanding of H.264 standard," Dooyangsa, ISBN: 89-7528-140-X / 93560, Feb. 2006.

[8] T. Wiegand, G.J. Sullivan and G. Bjontegaard, "Overview of the H. 264/AVC video coding standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 560-576, Jul. 2003.

[9] Peng, Fei, Xiao-wen Zhu and Min Long, "An ROI privacy protection scheme for H. 264 video based on FMO and Chaos," IEEE Transactions on Information Forensics and Security, Vol. 8, No. 10, pp. 1688-1699, Oct. 2013.

[10] Datong Chen, Yi Chang, Rong Yan and Jie Yang, "Tools for protecting the privacy of specific individuals in video," EURASIP Journal on Advances in Signal

Processing, Springer International Publishing, (2007): 075427, Online ISSN: 1687-6180, pp. 107-107, Jan. 2007.
〔11〕 Tomas Brandao, and Maria Paula Queluz, "No-reference PSNR estimation algorithm for H.264 encoded video sequences," IEEE, 16th European Signal Processing Conference (EUSIPCO 2008), Apr. 2015.
〔12〕 Wonsang You, "Analysis of H.264/AVC Encoder Reference Software," ResearchGate, Korea Information and Communication University, Apr. 2015.
〔13〕 Dongkeun Kim, "C++ API OpenCV Programming," Kame, ISBN: 978-89-8078-286-4, Dec. 2016.

〈저자소개〉

손 동 열 (Dong-yeol Son) 학생회원
2016년 2월: 원광대학교 전자공학과 전자공학심화전공 공학학사
2018년 2월: 아주대학교 지식정보공학과 IoT전공 정보통신 공학석사
〈관심분야〉 영상데이터 보안, IoT, 임베디드 소프트웨어, 스마트 그리드, 정보보호, 영상처리, 데이터 압축 처리

김 지 민 (Ji-min Kim) 학생회원
2015년 2월: 아주대학교 정보컴퓨터공학과 학사
2015년 3월~현재: 아주대학교 컴퓨터공학과 석·박사 통합과정
〈관심분야〉 임베디드/IoT 보안

지 청 민 (Cheong-min Ji) 학생회원
2012년 2월: 아주대학교 정보컴퓨터공학과 학사
2012년 3월~현재: 아주대학교 컴퓨터공학과 석·박사 통합과정
〈관심분야〉 차량 보안, 임베디드/IoT 보안, 블록체인 보안, 영상데이터 보안

김 강 석 (Kangseok Kim) 정회원
2007년: 인디애나대학교 컴퓨터공학과 박사
2010년~현재: 아주대학교 사이버보안학과 부교수
〈관심분야〉 모바일컴퓨팅, 모바일보안, 데이터마이닝

김 기 형 (Kihyung Kim) 종신회원
1990년: 한양대학교 전자통신공학과 졸업
1992년: KAIST 전자공학과 석사
1996년: KAIST 전자공학과 박사
1997년~2004년: 영남대학교 부교수
2000년~2001년: AdForce, Inc. Senior Engineer
2005년~현재: 아주대학교 정교수
〈관심분야〉 IoT/네트워크 보안, IoT 네트워크, 임베디드 소프트웨어

홍 만 표 (Manpyo Hong) 종신회원
1981년 2월: 서울대학교 계산통계학과 학사
1983년 8월: 서울대학교 계산통계학과 석사
1991년 2월: 서울대학교 전산과학과 박사
1985년 3월~2016년 2월: 아주대학교 정보컴퓨터공학부(과) 교수
2016년 3월~현재: 아주대학교 사이버보안학과 교수
〈관심분야〉 기반시설보안, 차량보안, 임베디드/IoT 보안, 금융보안, 병렬처리