

시간 정보를 활용한 동영상 파일 복원 기법

나기현^{*}, 심규선^{**}, 변준석^{***}, 김은수^{****}, 이 중^{*****}

Recovery Corrupted Video Files using Time Information

Gihyun Na^{*}, Kyu-Sun Shim^{**}, Jun-Seok Byun^{***}, Eun-Soo Kim^{****}, Joong Lee^{*****}

ABSTRACT

In recent crime scene, there is the captured crime scene video at least one. So video files recorded on storage media often provide important evidence. Criminals often attempt to destroy storage saved crime scene video. For this reason recovery of a damaged or deleted video file is important to resolve criminal cases in aspects of digital forensic. In the recent, there is a study to recover video file based on video frames, but it is very poor time efficiency when the connecting video frames. This paper proposed advanced frame-based recovery technique of a damaged video files using time information. We suggest a new connecting algorithm to connect video frames using recorded time information in front of video frame. We also evaluate performance in aspects of time and experiment result shows that proposed method improves performance.

Key words: Video Restoration, Corrupted Video Data, Reassemble Video File

1. 서 론

카메라, CCTV 및 차량용 블랙박스에 범죄 현상이 촬영되면, 동영상 파일¹⁾은 범죄 수사에서 현장을 입증할 수 있는 중요한 증거물이 된다. 하지만, 범죄의 은닉을 위하여 고의로 동영상 파일을 삭제하거나 저장매체를 포맷하는 경우가 많다. 또한, 외부 충격으로 불완전하게 동영상에 생성되어 재생활 수 없으며, 시간이 지남에 따라 자동 삭제되어 일부가 덮어써진다. 범죄를 입증할 수 있는 중요한 증거물들이 있으나, 이러한 이유로 범죄 현장을 확인할 수 없는 경우

가 발생한다. 이러한 이유로 범죄 현장을 입증할 수 있는 증거물을 찾기 위하여 영상에 대한 연구가 활발히 이루지고 있다[1-4].

기존 동영상 복원 기법은 1) 파일시스템을 기반 복원 기법, 2) 파일 시그니처 기반 복원, 3) 영상 프레임 단위 복원기법이 제안되었다. 파일시스템 기반 복원 기법은 파일시스템에 기록되어 있는 파일의 메타 정보인 디렉토리 엔트리를 활용한다[5]. 그러나 삭제된 동영상 데이터가 데이터 영역에는 존재하나 디렉토리 엔트리에 남아있지 않으면 복원이 불가능한 문제점이 있다[5].

* Corresponding Author : Joong Lee, Address: (220-170) Ipchun-ro, Wonju-si, Gangwon-do, Korea, TEL : +82-33-902-5310, FAX : +82-33-902-5921, E-mail : ljfirst@korea.ac.kr

Receipt date : Aug. 25, 2015, Revision date : Oct. 12, 2015
Approval date : Oct. 30, 2015

^{*} Department of Electronic Engineering, Kwang-woon University (E-mail : ghna282@korea.kr)

^{**} Digital Technology & Biometry Division, National Forensic Service (E-mail : blulesks@korea.kr)

^{***} Digital Technology & Biometry Division, National Forensic Service (E-mail : jsbyun77@korea.kr)

^{****} Department of Electronic Engineering, Kwang-woon University (E-mail : eskim@kw.ac.kr)

^{*****} Digital Technology & Biometry Division, National Forensic Service

※ This research was funded by the Forensic Research Program of the National Forensic Service (NFS), Ministry of Government Administration and Home Affairs, Korea.

1) This paper focused on the video data based on the multimedia area and it is also applicable to another types of multimedia.

이러한 문제를 해결하기 위하여 파일의 시작 시그너처(Start Marker)와 끝 시그너처(End Marker)를 찾아 복원하는 기법이 제안되었다[5-7]. 하지만 파일 시그너처 기반으로 하는 복원 기법은 파일의 조각이 많은 경우 복원이 어려운 한계가 존재하며 파일 단위로 복원하기 때문에 파일의 일부가 덮어쓰지면 파일 전체가 복원이 불가능하다[9].

파일의 조각이 많거나 일부가 덮어쓰진 경우에도 남아있는 멀티미디어 데이터를 복원하기 위하여 영상의 프레임 단위로 복원하는 기법이 제안되었다[9]. 또한 복원된 프레임 군들은 프레임 정보를 기록한 파일 메타데이터(STSZ)를 이용하여 연결한다. 만일 STSZ를 발견하지 못하면 복원된 프레임 군들을 1:1 매칭시켜 연결하는데, 이때 시간 복잡도는 $O(n^2)$ 이며, n 은 매칭시켜야할 프레임 군의 개수로 실제 처리 시간이 매우 크다. STSZ는 MP4 파일에 특정하여 존재하는 파일 메타데이터로 만일 복원해야할 멀티미디어 데이터가 MP4 파일 형태로 구성되어 있지 않다면 프레임 단위로 복원한 후에 모든 프레임 군들을 연결하는데 오랜 시간이 소요될 것이다.

본 논문에서는 CCTV 또는 블랙박스에서 영상이 저장될 때 시간 정보가 함께 기록되는 점에 착안하여 시간 정보를 활용하여 동영상을 복원하는 기법을 제안한다. 시간 정보를 활용하면 MP4 파일 형태로 구성되어 있지 않는 동영상도 시간 정보를 기반으로 프레임 군들을 연결하면 정렬 알고리즘의 시간복잡도인 $O(n \log n)$ 로 줄여 효율적으로 연결할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 본 논문의 관련된 관련 연구인 프레임 단위 영상 복원에 대해 설명하고, 3장에서 제안하는 시간 정보를 활용한 동영상 연결 알고리즘에 대해 설명한다. 그리고 4장에서는 제안한 알고리즘의 성능을 평가하고 5장에서 결론을 맺는다.

2. 관련 연구

파일 복원과 관련된 연구는 디지털 포렌식 분야에서 범죄 수사하는데 큰 기여를 하고, 여러 연구에서 디지털 포렌식 분야에서 파일을 복원하는 기술의 발전이 반드시 필요하다고 강조하였다[10-11].

2.1 파일기반 동영상 복원 기법

기존 파일 복원 기법은 저장 매체에 기록되어 있

는 파일시스템의 메타 정보를 기반으로 파일을 복원하였다[10-11]. “Sleuth Kit”은 언급한 방법으로 파일과 디렉토리 등의 정보를 가지고 프로그램으로 만들었다[11]. 하지만 이 기법들은 파일 시스템의 메타 정보를 알 수 없으면 복원이 되지 않는 한계가 존재하여, 파일시스템의 메타 정보를 활용하지 않고 콘텐츠를 기반으로 복원하는 기법이 연구 되었다.

저장 매체에 파일이 저장될 때, 여러 가지 이유로 조각이 나면서 저장이 되는데[6], 콘텐츠를 기반으로 데이터를 복원하는 기법은 조각이 난 파일에 대한 복원에 대하여 한계가 존재한다. 이 문제를 해결하기 위하여 많은 연구가 진행되었다.

콘텐츠를 기반으로 파일을 복원하는 프로그램 중 하나인 Scalpel[7]은 파일시스템의 메타 정보를 활용하지 않고 파일의 시작과 끝의 signature를 기반으로 파일을 복원하는 프로그램이다. 이 기법은 미리 정의된 파일의 시작과 끝 signature에 대하여 연속적인 데이터에 대해서 복원을 한다. 하지만 앞에서 언급하였듯이 저장 매체에는 파일이 조각나서 저장되는 경우가 많기 때문에 Scalpel은 조각난 파일을 복원하지 못하는 한계가 존재한다.

[6]에서는 파일의 signature를 기반으로 파일을 복원하는 기존 기법에서 파일의 헤더 부분에 기록되어 있는 메타 정보를 활용하였다. 일부 파일들이 파일의 헤더 부분에 파일의 길이를 기록해놓기 때문에 파일의 끝부분에 signature가 특별하게 없는 경우에도 파일의 길이를 활용하여 데이터를 추출할 수 있다. 또한 [6]에서는 “Bifragment Gap Carving”을 제안하여 파일이 두 개의 조각으로 저장되어 있을 때 복원할 수 있도록 하였다. 이 기법에서는 파일의 시작과 끝 지점을 구분한 뒤에 두 지점에서부터 확장하여 파일이 정상인지 validation하여 파일을 복원한다. 하지만 파일 단위로 복원하기 때문에 온전한 파일 형태로 복원이 되어야 영상이 가능하여 일부 덮어쓰진 동영상 파일에 대해서는 복원이 불가능하다.

2.2 프레임 단위 동영상 복원 기법

[9]에서는 파일 단위로 동영상을 복원하는 기법의 문제점인 온전한 파일 형태로 복원되어야 영상을 확인할 수 있는 문제점을 해결하여 위한 기법을 제안하였다. 제안한 기법은 동영상 파일이 영상의 의미있는 최소 단위인 프레임들이 군이며, 프레임은 코덱에 따

라 인코딩된다는 점에 착안하였다.

제안된 기법은 Fig. 1과 같이 크게 두 단계로 나뉜다.

- 프레임 복원 단계 : 프레임 복원단계에서는 저장 매체에서 복원해야할 데이터가 남아있는 있는 Unallocated Space를 추출 후 동영상 프레임을 디코딩할 수 있는 디코딩 헤더 정보와 영상 프레임 데이터를 추출한다. 프레임 데이터는 동영상 데이터를 인코딩한 코덱에서 프레임의 시작점을 알려주는 시그니처를 이용하여 찾을 수 있다. 디코딩 헤더 정보와 프레임 데이터를 결합 후 디코더를 통해 프레임 데이터가 실제 영상 프레임인지 검증하여 영상 프레임을 복원한다.

- 복원 프레임 연결단계 : 전 단계에서 검증된 영상 프레임을 물리적인 위치를 기반으로 프레임 군으로 묶은 후 파일 메타데이터(STSZ)가 발견될 경우 프레임 군의 프레임 크기를 기반으로 발견된 STSZ에 해당하는 프레임들을 연결한다. 반면, 파일 메타데이터(STSZ)가 발견되지 않을 경우 매칭되지 않은 모든 프레임 군들은 1:1 매칭을 통해 연결한다.

[9]에서 제안한 기법은 파일 메타데이터가 있는 경우에는 시간 복잡도가 낮지만, 파일 메타데이터가 발견되지 않으며 $O(n^2)$ 의 시간 복잡도를 가지게 된다. 일반적인 MP4 파일의 형태를 가지고 있는 동영상 파일은 파일 메타데이터를 가지고 있지만, MP4 파일 형태를 가지지 않는 경우에는 많은 시간이 소요되는 문제점이 있다.

3. 제안한 방법

3.1 제안한 기법의 개요

본 논문에서는 CCTV, 블랙박스에 저장된 동영상 파일이 비정상적으로 저장되어 재생되지 않거나, 삭제 후 일부가 덮어써져 영상을 확인할 수 없는 경우 동영상 파일의 의미있는 최소 단위인 프레임 단위로 복원 후 영상 프레임 데이터 앞 부분에 기록된 시간 정보를 추출하여 연결된 동영상으로 복원하는 기법을 제안한다. Fig. 2는 논문에서 제안하는 기법에 대한 전체적인 흐름을 나타낸다.

Fig. 2의 프레임 추출 단계는 [9]에서 제시한 방법으로 영상 프레임을 추출한다. [9]에서 제시한 방법인 Fig. 3과 같이 증거물에서 삭제된 영역인 비할당 영역을 추출한다. 비할당 영역에서는 Fig. 3의 2단계와 같이 디코딩에 필요한 코덱 헤더 정보들과 프레임 시그니처를 통해 프레임 데이터를 추출하고 디코더를 이용하여 디코딩 정보와 추출된 프레임 데이터를 결합하여 영상 프레임을 복원한다. 만일 디코딩이 성공하면 복원된 프레임으로 확인하고, 실패시 복원하고자 하는 영상의 참조 파일에서 디코딩 정보를 가져와서 다시 디코딩을 실행한다. 다시 실행한 디코딩에서도 실패시에는 프레임 데이터에서 제외하고 복원된 프레임으로 프레임 군을 형성한다. Fig. 3에서 복원된 영상 프레임은 F_1, F_2, \dots, F_n 과 같고, 물리적으로 연속적인 프레임들을 하나의 프레임 군으로 형성한다. Fig. 3의 예에서는 F_1, F_2, F_3 가 첫 번째 프레임 군, F_j, F_{j+1}, F_{j+2} 가 두 번째 프레임 군, F_{n-2}, F_{n-1}, F_n 이 세 번째 프레임 군으로 형성하게 된다 .

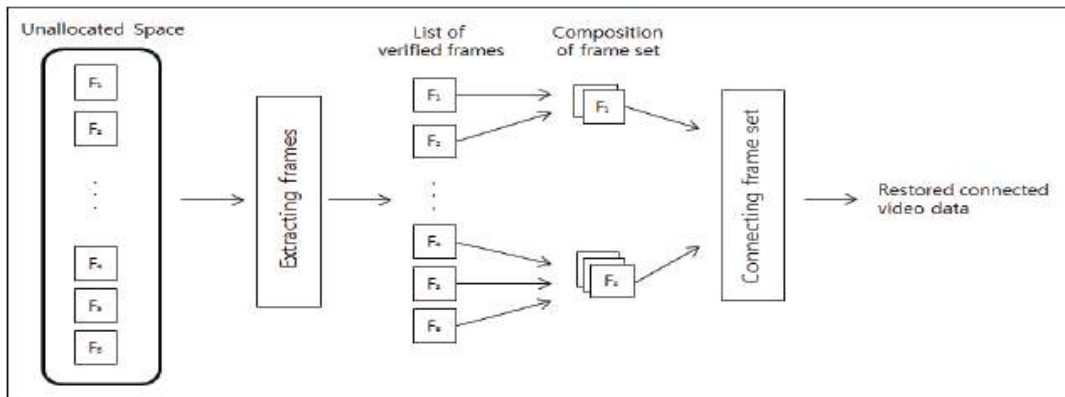


Fig. 1. Processing steps of the frame-based video file restoration technique[9].

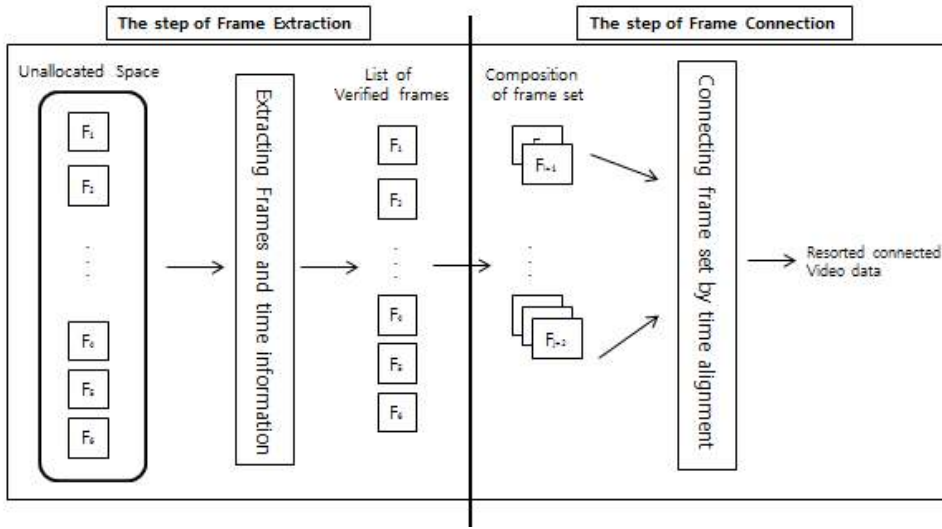


Fig. 2. Processing steps of proposed method.

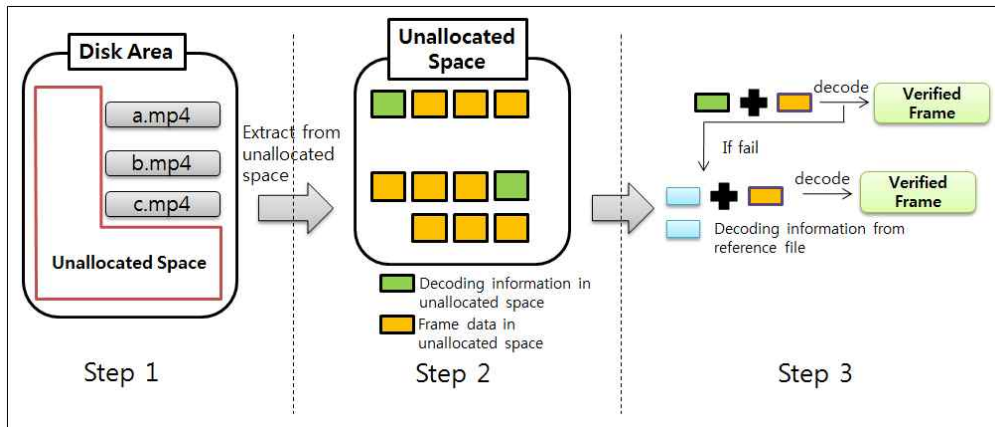


Fig. 3. Processing steps of frame recovery[9].

본 논문에서는 Fig. 3의 2단계에서 영상 프레임을 복원할 때 각 영상 프레임에 해당되는 시간 정보를 추출한다. 형성된 프레임 군의 첫 번째 영상 프레임의 시간 정보로 프레임 군을 정렬 후 프레임 군을 연결하여 연결된 하나의 동영상을 만든다.

[9]에서는 MP4 파일의 메타정보인 STSZ를 기반으로 하였기 때문에 MP4 파일 컨테이너 구조가 아닌 영상 파일과 STSZ가 떨어져 찾을 수 없는 경우에 영상 프레임을 연결할 때 brute-force 방식인 1:1 매칭으로 프레임 군을 연결하기 때문에 시간 복잡도가 $O(n^2)$ 였다. 하지만 본 논문에서 제안하는 기법으로 시간 정보를 기반으로 정렬 후 영상 프레임을 연결할

때는 정렬 알고리즘의 시간 복잡도인 $O(n \log n)$ 으로 효율이 향상된다.

3.2 시간 정보 추출 방법

본 논문에서 제안하는 기법으로 동영상 복원시 효율을 향상하기 위해서 시간 정보를 추출하는데, 영상 프레임 앞 부분에 시간 정보를 기록하고 있는 블랙박스, CCTV의 경우 약 90%의 모델들이 Table 1과 같이 시간 정보를 기록하고 있다. 나머지 모델은 제조사의 자체 시간 함수를 두어 시간 정보를 기록한다. Table 1은 2015년 3월 25일 10시 14분 42초를 타임별로 변환한 데이터를 명시하였다.

Table 1. The type of time notations

Type	Time Format	Description
1	0 × 928A1255	4Byte Hexadecimal by Little Endian
2	0 × 55128A92	4Byte Hexadecimal by Big Endian
3	0 × 014C506D4A50	6Byte Hexadecimal by Big Endian
4	0 × 0F03190A0E2A	Each byte is shown year, month, day, hour, minute, and second as a hexadecimal in the order.
5	0 × 150325101442	Each byte in order displays year, month, day, hour, minute, and second as a decimal.

첫 번째 타입과 두 번째 타입은 빅 엔디안(Big Endian)과 리틀 엔디안(Little Endian)의 차이이며, Endian 변환으로 통해 4 바이트 16진수를 10진수로 변환한다. 변환된 10진수는 1970년 1월 1일부터 흐른 초 시간으로 이를 실제 시간으로 변환하여 시간을 확인할 수 있다. 세 번째 타입은 6바이트 16진수를 10진수로 변환하면 1970년 1월 1일부터 흐른 밀리세컨드 시간이며, 네 번째 타입은 각 바이트가 연월일시분초를 16진수로 표기한 것으로 각 바이트를 10진수로 변환하면 시간 정보를 확인할 수 있다. 다섯 번째 타입은 각 바이트가 연월일시분초를 그대로 표기한 것으로 바로 변환하여 시간 정보를 확인할 수 있다.

Fig. 4은 Table. 1의 첫 번째 타입인 4 바이트 16진수로 영상 프레임 앞 부분에 시간 정보가 기록되어 있는 실제 예이다. Fig. 4에서 0×0000000165는 영상 코덱인 H.264의 프레임 시그니처이고, 시그니처 앞 48 바이트부터 4 바이트는 해당 프레임 데이터가 저장된 시간을 기록하고 있다. 0×92235051은 Little Endian 식 표기로 10진수로 변환하면, 1,364,206,482가 되고 이를 1970년 01월 01일부터 흐른 초 시간으로 계산하면, 2015년 03월 25일 10시 14분 42초인

것을 확인할 수 있다.

제안하는 기법은 영상 프레임 추출할 때, 앞 부분에 기록되어 있는 시간 정보를 함께 추출한 후 프레임 군을 연결할 때, 1:1 매칭이 아닌 프레임 군의 첫 번째 프레임 시간 순으로 정렬하여 연결할 수 있다. 또한 Fig. 3에서 프레임 시그니처의 48 바이트 앞부터 4바이트가 시간 정보를 나타내는 것과 같이 블랙박스, CCTV 기종마다 동일한 위치에 시간 정보가 있기 때문에 시간 정보를 추출하는데 소요되는 시간은 매우 작아 속도 측면에서 매우 효율적이다.

3.3 프레임 군 연결 방법

[9]에서는 동영상 파일이 삭제된 후 조각이 나서 연결이 되지 않거나, 일부가 덮어써졌을 경우에도 연결된 동영상으로 복원하기 위해 MP4의 메타정보(STSZ)가 없는 경우 모든 프레임 군을 1:1 매칭을 통해 조각난 프레임 데이터를 연결하여 하나의 동영상으로 복원하였다. 하지만 모든 프레임 군들을 1:1 매칭하게 되면 시간 복잡도가 $O(n^2)$ 로 매우 비효율적이다.

Fig. 5는 본 논문에서 제안하는 기법으로 하나의 동영상 파일이 조각이 발생하였을 경우에 프레임 군

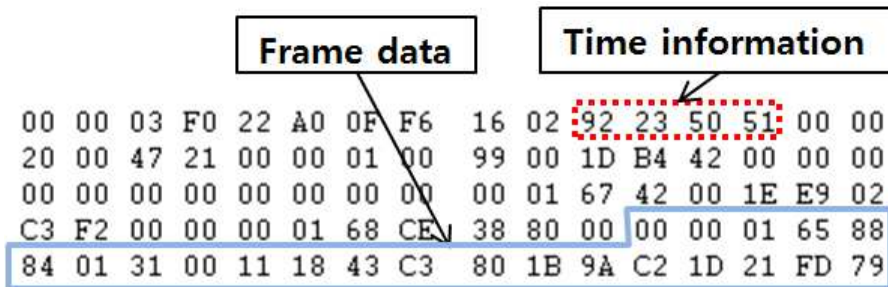


Fig. 4. The Example of Video Recording Time.

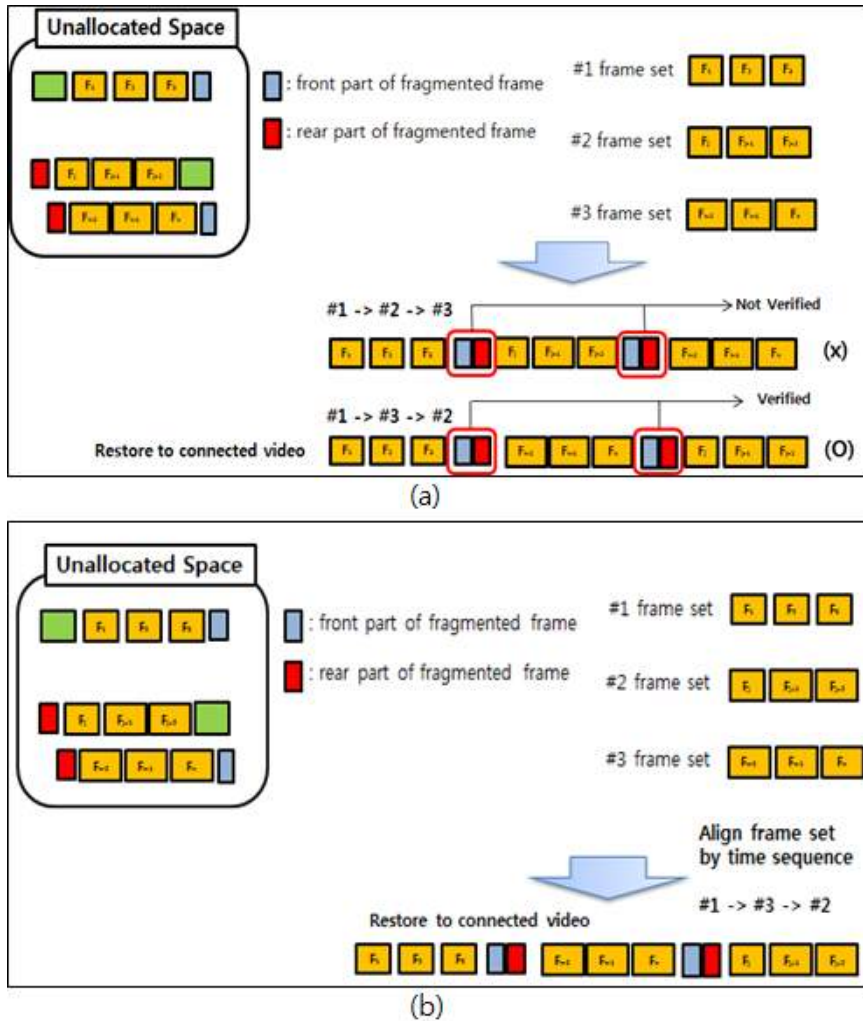


Fig. 5. Connecting frame set without STSZ, (a) brute-force method[9], (b)connecting using time sequence.

을 연결하는 방법을 도식화 한 예이다. Fig. 5에서는 비활동 영역에서 복원된 프레임을 기반으로 #1 프레임 군과 #2 프레임 군, #3 프레임 군이 형성되었다.

Fig. 5의 (a)에서는 기존 기법에서 MP4의 메타정보(STSZ)가 없을 경우 프레임 군을 연결하는 기법이다[9]. 비활동 영역에서 복원된 프레임을 기반으로 #1, #2, #3의 프레임 군이 형성되었을 때, 연결된 동영상으로 복원하기 위해 형성된 모든 프레임 군들을 연결해본다. 먼저 #1 프레임 군, #2 프레임 군을 연결해서 연결 부분에 조각난 프레임을 디코딩하여 영상 프레임인지 판별한다. 영상 프레임이 아닐 경우에는 #1 프레임 군과 #3의 프레임 군을 연결하여 조각난 부분의 프레임이 영상 프레임인지 판별하여 영상 프

레이프로 판별되면 #3 프레임 군과 #2 프레임 군을 연결하여 복원된 동영상 파일로 복원한다.

Fig. 5의 (b)는 본 논문에서 제안한 기법으로 MP4의 메타정보(STSZ)가 없더라도 시간 정보를

이용하여 프레임 군을 연결하는 기법이다. 먼저, 각 프레임 군의 첫 프레임인 F₁, F₂, F_{n-2}의 시간 정보를 확인한다. 세 프레임의 시간 정보를 정렬한 결과, 프레임 군의 순서는 #1, #3, #2인 것을 확인하면 해당 순서에 따라 프레임 군을 #1, #3, #2 순서대로 연결한다. 프레임 군들을 연결하게 되면, 조각난 프레임의 앞부분과 뒷부분도 연결되기 때문에 하나의 온전한 동영상 프레임으로 복원이 가능하다.

만일 동영상이 삭제 후 덮어써지더라도 시간 순

로 정렬하게 되면, 연속적인 시간 부분을 연결하기 때문에 삭제되지 않은 부분들은 모두 복원되어 제안하는 기법으로 연결할 수 있다.

4. 실험 결과 및 고찰

본 논문에서 제안하는 기법을 평가하기 위해 먼저 프레임 군을 연결하는데 소요되는 시간을 비교하였다. 제안된 기법은 [9]에서 MP4 파일의 메타정보(STSZ)가 없을 경우 1:1 매칭을 통해 프레임 군을 연결하는 단점을 보완하고자 하기 때문에 프레임 군을 연결하는 부분을 제외한 프레임 추출과 검증 부분은 동일하기 알고리즘을 사용하여서 성능 비교에서는 고려하지 않고 프레임 군을 연결하는 부분에 대하여서만 성능을 평가하였다.

수학적으로 분석하면 [9]에서 언급된 방법은 MP4 파일의 메타정보(STSZ)가 없을 경우에 프레임 군을 1:1로 매칭하기 때문에 프레임 군의 개수가 n개라고 하였을 때, 시간 복잡도는 $O(n^2)$ 이 된다. 동영상 파일들이 증거물에서 조각의 개수가 많을 경우에는 n이 커지고, 조각의 개수가 작을 경우에는 n이 작아진다. 조각의 개수가 많으면 많아질수록 시간 복잡도가 기하급수적으로 증가하여 [9]에서 제안한 기법은 효율이 떨어진다.

제안하는 기법은 프레임 군을 연결할 때 정렬 알고리즘을 사용하여 프레임 군을 시간 순으로 정렬 후 연결한다. 시간 순으로 정렬하여 연결을 하게 되면 정렬 알고리즘 시간 복잡도인 $O(n \log n)$ 이 제안하는 기법의 시간 복잡도가 된다. 하지만 제안하는 기법에서는 기존 기법[9]에 추가로 시간 정보를 추출하는 시간이 소요되지만 영상 프레임을 추출할 때 동시에 하기 때문에 프레임 군 연결하는 것에 비해 매우 짧은 시간이므로 시간 복잡도 계산 시 고려하지 않았다.

Table 2는 CPU Intel Core(TM) i5(2.40 GHz 듀얼 코어), OS는 Windows 7 Enterprise K Service Pack 1, 메모리는 8 GB인 컴퓨터에서 프레임 군을 연결하여 연결된 영상으로 복원하는데 소요되는 시간을 측정한 결과이며, 실험을 통해 확인한 결과 복원율에는 차이가 발생하지 않았다. 실험에 사용한 동영상 파일은 MPEG-4 Visual 코덱으로 인코딩된 약 6MB의 동영상 파일 20개에 대해 각 프레임 군의 개수가 10

Table 2. The performance for proposed method and conventional method (unit : min)

the number of frame set	Proposed method(A)	The method of [9](B)	Rate(%) (B/A*100)
10	18	24	133.3
20	24	66	275.0
30	36	126	350.0
40	48	216	450.0
50	60	312	520.0
60	90	408	453.3
70	102	600	588.2
80	126	834	661.9
90	132	972	736.4
100	156	1,176	753.8
200	360	4,434	1231.7
400	768	16,962	2208.6

개, 20개, 30개, 40개, 50개, 60개, 70개, 80개, 90개, 100개, 200개, 400개로 생성하여 프레임 군 연결하는데 소요되는 시간을 측정하였다.

Table 2와 같이 프레임 군의 개수가 10개일 때는 제안하는 기법으로 연결하였을 때는 약 18분 소요되고, [9] 기법으로는 약 24분 소요되어 본 논문에서 제안한 기법을 통해 성능이 개선됨을 확인할 수 있었다. 그리고 프레임 군이 개수가 커지면 커질수록 소요되는 시간이 매우 커지고 성능의 차이도 프레임 군의 개수가 400개일 때는 768분과 16,962분으로 매우 큰 차이를 보였다. 수학적으로 해석한 n, 즉 프레임 군의 개수가 커질수록 [9]의 기법은 연결하는데 소요되는 시간이 매우 커짐을 확인할 수 있었으며, 제안하는 기법에서는 소요되는 시간의 상승이 상대적으로 작았다. 이는 제안하는 기법에서 프레임 군을 연결하기 위하여 모든 프레임을 1:1 매칭을 하기 때문에 소요되는 시간이 매우 컸기 때문이다. 반면 제안하는 기법에서는 정렬 후 순서대로 프레임 군을 연결을 하였기 때문에 성능이 크게 개선된 것을 확인할 수 있다.

또한, 논문에서 제안한 기법의 연결된 동영상 복원 성능을 실험하기 위하여 C++언어를 활용하여 저장 매체에 MPEG-4 Visual 코덱으로 인코딩되어 MP4 컨테이너로 구성된 동영상 파일 20개를 임의의 크기로 4개로 조각내었으며, 0~90%로 덮어쓰기 하였다.

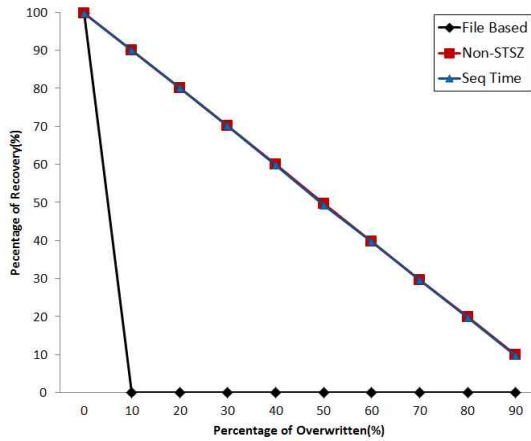


Fig. 6. Recover Performance of Frame Connection.

Fig. 6은 덮어쓰기 정도에 따른 프레임 연결에 어떤 영향을 미치는지 실험한 결과이다. 동영상 조각이 4개인 상황에서 실험 결과는 덮어쓰기의 정도에 클수록 연결된 동영상으로 복원율이 낮아지지만 덮어써지지 않는 부분은 대부분의 프레임들이 연결되어 복원된다는 것을 알 수 있다. 반면 기존 파일 단위로 복원하는 기법에서는 MP4 컨테이너로 구성된 동영상 파일은 온전한 파일이 존재하여야 연결된 동영상으로 복원되기 때문에 동영상의 일부가 덮어써지면 복원이 전혀 되지 않는다는 것을 알 수 있다.

Non-STSZ는 [9]에서 제안한 MP4의 메타정보(STSZ)가 없을 경우에 복원율을 표기한 것이다. Non-STSZ의 경우에도 디코더를 통해 조각난 부분이 있다고 검증되는 부분을 모든 프레임 군과 매칭시켜 디코더를 통해 검증하여 찾지 못하면 일부가 덮어써지더라도 나머지 프레임들이 연결되어 복원할 수 있다는 것을 알 수 있다.

Fig. 6의 Seq. Time으로 명시된 부분은 본 논문에서 제안한 시간 정보를 이용하여 프레임 군을 연결하는 기법으로 복원율은 기존 기법에서 제안한 Non-STSZ와 유사한 성능을 보였다. 형성된 프레임 군을 시간 순으로 정렬하여 조각난 부분을 연결하기 때문에 덮어써지지 않은 프레임 군들이 모두 시간 순으로 정렬과 동시에 연결되었기 때문에 덮어써지지 않은 프레임들이 복원된다는 것을 알 수 있다.

5. 결론

본 논문에서는 동영상 파일에서 의미있는 최소 단

위인 프레임 단위로 복원하는 기법에서 파일의 메타 정보(STSZ)가 없는 경우에 복원된 프레임들의 집합인 프레임 군을 연결하는 효과적인 방법을 제시하고 있다. 기존 기법에서 1:1 매칭으로 프레임 군을 연결하였는데, 제안하는 기법은 영상 프레임 앞에 기록되어 있는 부가정보인 영상이 녹화된 시간 정보를 이용하여 프레임 군을 정렬 후 연결하였다. 시간 정보가 있는 특성을 이용함으로써 소요되는 시간은 시간 복잡도가 $O(n^2)$ 에서 $O(n \log n)$ 으로 향상되었고, 실제 실험을 통해 측정된 시간에서도 성능이 프레임 군의 개수가 많을수록 성능이 매우 향상된 것을 확인할 수 있다.

실제 동영상 복구시에는 최근 메모리와 CCTV에 장착된 하드디스크의 용량이 계속 증가하기 때문에 실험에서 평가한 400개의 프레임 개수보다 훨씬 많을 것으로 예상되기 때문에 본 논문에서 제안한 시간 정보를 이용하여 연결하여야만 실제로 복구에 사용할 수 있을 것으로 예상된다.

이후에는 모든 동영상 파일들이 시간 정보가 영상 프레임 앞에 기록되어 있는 것이 아니기 때문에 영상에 자막형태로 기록되어 있는 시간 정보를 정확하게 추출하여 제안한 기법이 적용 가능한 영역을 확장할 수 있는 연구를 진행할 예정이다.

REFERENCE

- [1] R. Poisel and S. Tjoa, "Roadmap to Approaches for Carving of Fragmented Multimedia Files," *Proceedings of the 2011 6th International Conference on Availability, Reliability and Security*, pp. 752-757, 2011.
- [2] A.B. Lewis, *Reconstructing Compressed Photo and Video Data*, Technical Report in University of Cambridge, No. 813, 2012.
- [3] R. Poisel, S. Tjoa, and P. Tavolato, "Advanced File Carving Approaches for Multimedia Files," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 2, No. 4, pp. 42-58, 2011.
- [4] J.M. Kim, S.H. Kim, "Real Time Enhancement of Image Degraded by Bad Weather," *Journal of Korea Multimedia Society*, Vol. 17, No. 2,

pp. 143-151, 2014.

- [5] A. Pal and N. Memon, "The Evolution of File Carving," *IEEE Signal Processing Magazine*, Vol. 26. No. 2, pp. 59-71, 2009.
- [6] S.L. Garfinkel, "Carving Contiguous and Fragmented Files with Fast Object Validation," *Proceedings of the Digital Forensics Research Workshop*, pp. 4S:2-12, 2007.
- [7] G.G. Richard III and V. Roussev, "Scalpel: A frugal, High-performance File Carver," *Proceedings of the 2005 Digital Forensics Research Workshop*, pp. 1-9, 2005.
- [8] N. Memon, "Automated Reassembly of File Fragmented Images using Greedy Algorithms," *IEEE Transactions on Image Processing*, Vol. 15, No. 2, pp. 385-393, 2006.
- [9] G.H. Na, K.S. Shim, K.W. Moon, S.G. Kong, E.S. Kim, and J. Lee, "Frame-Based Recovery of Corrupted Video Files using Video Codec Specifications," *IEEE Transactions on Image Processing*, Vol. 23, No. 2, pp. 517-526, 2014.
- [10] B. Carrier, *File System Forensic Analysis*, Addison-Wesley Professional, Indiana, USA, 2005.
- [11] The Sleuth Kit(2005), <http://www.sleuthkit.org/sleuthkit/> (accessed Aug., 17, 2015).



나 기 현

1998년 2월 경북대학교 전자공학과 공학사
 2000년 2월 경북대학교 대학원 전자공학과 공학석사
 2001년 11월~현재 국립과학수사연구원 디지털분석과

2010년~현재 광운대학교 대학원 전자공학과 박사과정
 관심분야: 파일복원, 비디오 재구성, 디지털포렌식, 3D 디스플레이



심 규 선

2009년 2월 고려대학교 컴퓨터학과 학사
 2011년 2월 고려대학교 대학원 컴퓨터·전파통신공학과 공학석사
 2011년 11월~현재 국립과학수사연구원 디지털분석과

관심분야: 파일복원, 비디오 재구성, 디지털포렌식, 모바일포렌식



변 준 석

2003년 8월 경남대학교 컴퓨터공학과 공학사
 2010년 2월 한양대학교 대학원 정보시스템공학 공학석사
 2010년 11월~현재 국립과학수사연구원 디지털분석과

관심분야: 파일복원, 비디오 재구성, 디지털포렌식, 정보보안



김 은 수

1978년 2월 연세대학교 전자공학과 공학사
 1980년 2월 연세대학교 대학원 전자공학과 공학석사
 1984년 2월 연세대학교 대학원 전자공학과 공학박사

1987년 2월~1988년 8월 California Instituted Technology 전자공학과 객원교수

1992년 3월~현재 광운대학교 전자공학과 교수

관심분야: 3D 영상, 3D 디스플레이, 3DTV, 홀로그래피



이 중

1994년 2월 광운대학교 화학공학과 공학사
 1999년 2월 광운대학교 대학원 화학공학과 공학석사
 2004년 2월 광운대학교 대학원 화학공학과 공학박사
 1995년 4월~2012년 6월 국립과학수사연구원 디지털분석과

2012년 6월~현재 국립과학수사연구원 디지털분석과장

2008년 3월~2013년 12월 광운대학교 강사 및 겸임교수
 관심분야: 디지털포렌식, 화질개선, 영상처리, 문서보안