International Journal of Internet, Broadcasting and Communication Vol.11 No.3 64-70 (2019)
http://dx.doi.org/10.7236/IJIBC.2019.11.3.64

IJIBC 19-3-9

Design and Implementation of Damaged Video File Recovery Tool using Container Format Structure

Yun-Seok Choi, Wan Yeon Lee

Dept. of Computer Science, Dongduk Women's University, Korea
cooling@dongduk.ac.kr, wanlee@dongduk.ac.kr

Abstract

Video files of video devices such as black box and CCTV may be damaged due to repetitive file read / write and physical environment factors. Even though there are available parts of video information, it may happen that playback can't be performed due to damage of some information. To playback the remaining video information normally, it is necessary to recover damaged areas of the files. For this, it is necessary to accurately check the damage range of the files. In this paper, we propose the design and implementation of a tool which detects damaged areas of a video file and recovers the usable area of the file to playback. The proposed tool can analyze and recover without additional information by analyzing common information of video container format and can check detailed damaged ranges with chunks. It is possible to perform recovery just only with the target file and reference file without any other information such as codec specification.

Keywords: *Video Container Format, Video File Recovery, MP4*

1. Introduction

The use of video is essential in security and safety critical situations. So, various devices such as black box and CCTV which capture sequential videos are used [1-3]. Repeated file reading and writing, and various physical environment factors can cause damages to video files [4, 5]. Repeated overwriting an existing area of files, a CRC(Cyclic Redundancy Check) error in file transmissions, a damaged file system, and a virus infection can be made damaged video files. Sometimes, a video file can't be playback despite of small areas of the file are damaged. Therefore, in case of video files storing information of important points, to playback the remaining video information normally, it is necessary to recover damaged areas of the files. For this, it is necessary to accurately check the damage range of the files. To accurate recover for damaged areas, a frame-based method which recovers damaged video files was proposed [6]. The method recovers damaged video file frames using codec information which was used to encoding the file. Therefore, recovery may be difficult without codec information of the target file. In this paper, we propose the design and

Manuscript Received: June. 16, 2019 / Revised: June. 24, 2019 / Accepted: June. 30, 2019
Corresponding Author: cooling@dongduk.ac.kr
Tel: +82-2-940-4683, Fax: +82-2-940-4170
Dept. of Computer Science, Dongduk Women's University, Korea

implementation of the tool which scans and recovers damages video file with the video container format structure of the file. The tool recovers MP4 video file using container format information of the target file and a reference file created by the target file create device. The rest of this paper organized as follows, In Section 2, we review the backgrounds of our work, Design and implementation of the video file recovery tool is shown in Section 3. The implementation and discussion about the tool are shown in Section 4. Section 4 concludes the paper.

2. Backgrounds

2.1 Key box of MP4 container format

MP4 is major multimedia container format which is used widely like AVI. It is one of the object-oriented forms of multimedia container format standard [7]. MP4 container format is composed with a set of 'box' also known as 'atom' and a BOX can be a single box or a set of boxes [8, 9]. The key boxes of MP4 are 'ftyp', 'mdat', and 'moov' which are top level box. 'ftyp' is the first box of MP4 validates file compatibility. 'moov' represents meta data of the media information and encoded media data are stored in 'moov'. 'moov' and 'mdat' have no fixed sequence. In general, 'moov' is located after 'mdat' because media information can be created only after the media data is stored. To use progress downloads or video streaming on HTTP, 'moov' should be placed in front of 'mdat'. Generally, 'moov' is located after 'mdat' in MP4 files created by black box or cctv. 'moov' may also be placed in front of 'mdat' when the file is edited by editors. Since the video file created by one device uses the same container format, common information of the video file structure can be used for damaged file scanning and recovery.

2.2 Damaged type of MP4 files

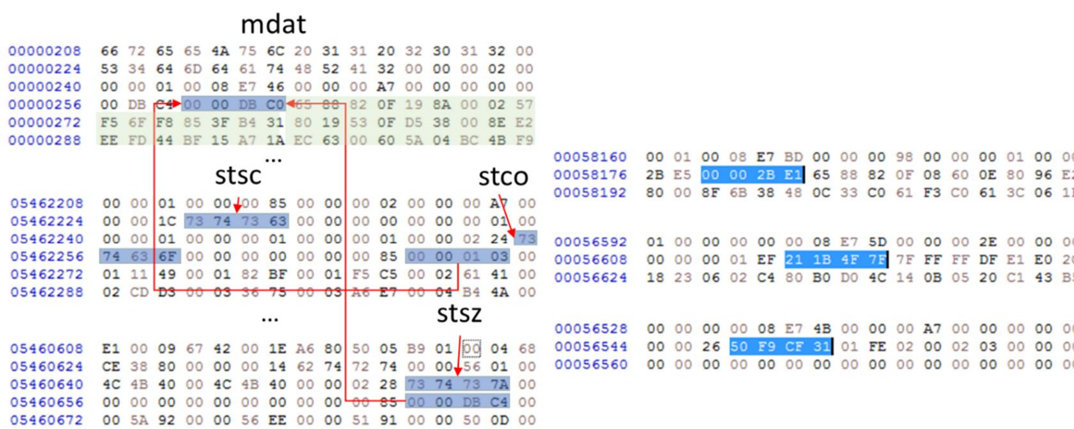
The degree of damage to 'ftyp', 'mdat', and 'moov' box affect a MP4 file play. If 'ftyp' is damaged, the file can't be played because it has no type data. In 'mdat', media data stored by chunks, so it can be used to play except damaged chunks. 'moov' has important meta data of MP4 file therefore a little damage of information can make no playable file. Damages of sub boxes in 'moov' except 'iods', 'mdhd', 'tkhd', and 'vmhd' can make non-playable files. Especially, 'stco', 'stsz', and 'stss' related with media chunks are created by unique information of a file. So, it can't be play and it is impossible to check the area of damage when the boxes are damaged. In this paper, mp4 files that 'moov' is located after 'mdat' and can be checked information of 'stco', 'stsz', and 'stss' are recovery target.

3. Design and Implementation of Recovery Tool

In this section, we will research information for video recovery and propose the architecture of video recovery tool. To perform recovery of damages areas, we analyzed the structure of video files based on the study of video file structure and searching damage locations in [10, 11], and then recover damaged areas. In this study, restoration of video track is focused on.

3.1 Key information for video recovery

In order to recover video files, the role of each box constituting the container format of the video file should be confirmed. 'ftyp' is the key box to identify video file type. Video files created by the same device have same 'ftyp' box contents. So, if 'ftyp' box is damaged on the target file, 'ftyp' of another file which created by the same device can be replaced to recover. The 'mdat' contains the media data of video, sound, and text. each of data consists of chunks and a chunk can have one or more samples. The format of contents in a chunk can have various type by codecs. Information about each chunk are stored in ‘trak’ which is a sub box of ‘moov’. ‘trak’ exists by the type of media such as video, sound, and text. ‘stco’, ‘stsz’, and ‘stsc’ are key box of ‘trak’ which used for analysis of chunk and sample information. Total number of chunks and offset of chunks for ‘trak’ can be found in ‘stco’. ‘stsc’ shows the number of samples per chunk and ‘stsz’ shows total number of samples and each sample size. In the video ‘trak’, the sample size value of ‘stsz’ can be used as a signature to identify the sample. Figure 1 shows the relationship between key boxes and chunks and examples of chunk signature.



(a) The relationship between key boxes (b) examples of chunk signature

Figure 1. The relationship between key boxes and chunks.

A chunk which offset is recorded in ‘stco’ and the first part of the chunk has a 4 byte signature. It is possible to check whether the chunk is damaged by checking whether the signature exists normally at the offset of each chunk. Therefore, the number of damaged chunk in total chunks can be calculated by check the chunk signature.

3.2 System architecture of recovery tool

The proposed tool uses the container format information of the target file and the reference file to scan damaged areas on the target file and recover ‘ftyp’ and chunk signatures of ‘mdat’. Figure 2 show the architecture of the recovery tool.

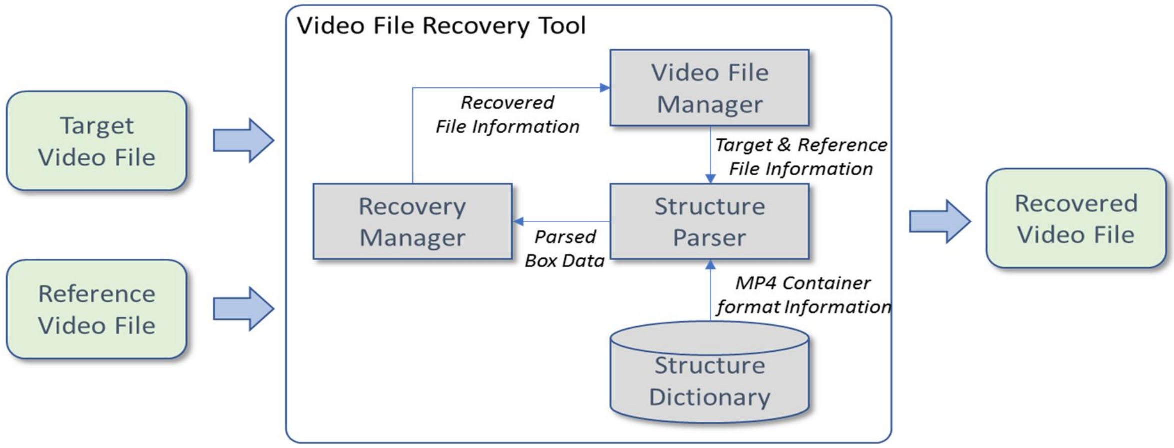


Figure 2. System architecture.

The video file manager reads the target file and the reference file and acquires file path, name, size and basic information of the each files. The acquired information passed to the structure parser. The structure parser scans the target file to find damaged areas of 'ftyp' and 'mdat' and parses 'moov'. Parsing of 'moov' using the information of the structure dictionary. The structure dictionary provides detailed information of each box such as kind of attributes, byte size and data type of attributes. 'ftyp' contents and signature patterns of chunks are obtained through the reference file analysis of the structure parser. Table 1 shows the acquired information for recovery from the target and reference file.

Table 1. Acquired information

Information	Source	Purpose of use
ftyp contents	Reference file	Recovery of start contents
Offset of mdat	Target & reference file	Recovery of mdat
Size of mdat	Target & reference file	Recovery of mdat
Offset of chunks	Target file	Scanning damaged chunks
Size of chunks	Target file	Recovery signature of chunks
Signature of chunks	Target & reference file	Scanning damaged chunks & recovery signature

The recovery manager verifies the location and size of each chunk in 'mdat'. It verifies whether each chunk is damaged by checking the chunk signature. The signature of each chunk is determined by the size of the chunk in the case of the video, the designated value of the codec in the case of the sound and the text and may vary depending on the codec. After obtaining the necessary information for recovery, the recovery manager creates a copy of the target file through the file manager and restores the video by applying recovery information.

4. Implementation and discussion

We implemented the video recovery tool based on the proposed system architecture with java. The target file information such as file path, file name, size, and check results for damage are checked and showed. The contents of 'ftyp', the type of 'trak', and signature pattern is identified from the reference file. Figure 3 shows the execution of the recovery tool.

After loading the target file and the reference file, analysis of ‘moov’ of target file can be used to display the degree of damage of the chunk according to ‘trak’ box representing each track. Damaged chunks of the whole chunk are displayed visually and information for checking by numerical value are showed. After the analysis is completed, a recovery file reflecting the recovery information can be created. To verify the usefulness of the implemented video recovery tool, file recovery was performed by dividing the video data collected from 5 types of black boxes into 'ftyp' damage, 'mdat' information damage and media information damage. Damaged target files were made by overwriting arbitrary byte values to the damage area. Figure 4 shows an example of the damaged video file and the recovered file.

As a result of recovering each file, we could check the damage areas of chunk by ‘trak’. It is possible to play back a non-playable movie normally by restoring 'ftyp' contents and chunk signatures in 'mdat'. It is possible to perform recovery just only with the target file and reference file without any other information such as codec information.

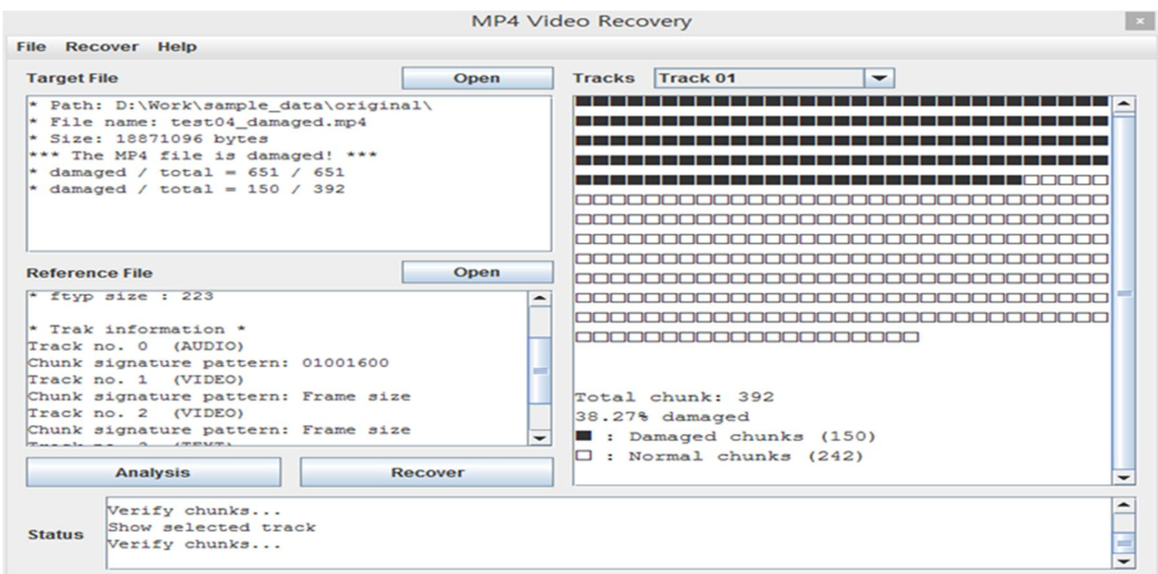


Figure 3. The recovery tool.

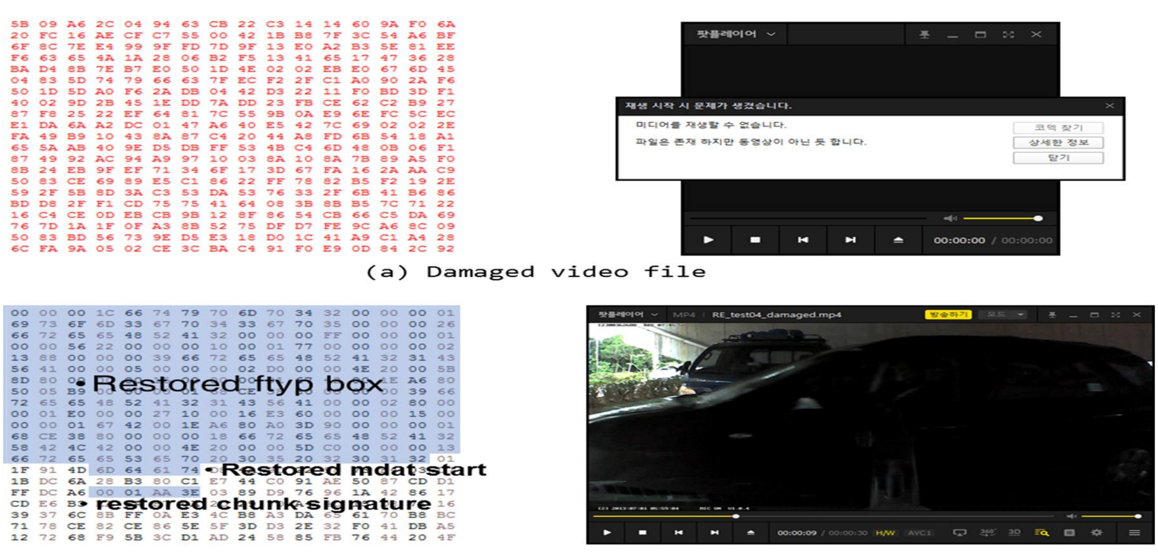


Figure 4. Example of a damaged video file and a recovered file.

5. Conclusion

Files of video devices that perform continuous shooting may cause damages of files due to various factors. Even if there are video information available, it may not be possible to play back the video by some information damage. Therefore, a recovery tool that can recover the damaged information and utilize the available information is needed. We designed and implemented a tool to recover damaged files using only damaged files and reference video files created by the same device. The proposed tool identifies the damaged area by the chunk unit and recovers video information by using the information of container format structure. The proposed tool recovers video files without any additional information other than video files, and it is possible to identify the damage area of files in chunks.

It is necessary to study the method of reconstructing the video file when the damage of ‘moov’ is occurred by extending the proposed tool as a future work. In addition, it is necessary to study the restoration of the video media information itself which contains the damaged chunk.

Acknowledgement

This research was supported by the Dongduk Women’s University Grant, 2018.

References

[1] J.Y. Choi and N.S. Chang, “Integrity Verification in Vehicle Black Box Video Files with Hashing Method,” The Journal of Korean Institute of Communications and Information Sciences(KICS), Vol. 42, No. 1, pp. 241-249, Jan. 2017.
DOI: 10.7840/kics.2017.42.1.241

[2] T. Gloe, A. Fischer, and M. Kirchner, “Forensic Analysis of Video File Formats,” Digital Investigation, Vol. 11, pp. 68-76, May 2014.
DOI: https://doi.org/10.1016/j.diin.2014.03.009

[3] C.G. Lee, J.H. Lee, Y.B. Pyo, and H.J. Lee, “Broken Integrity Detection of Video Files in Video Event Data Recorders,” KSII Transactions on Internet and Information Systems, Vol. 10, No. 8, pp. 3943-3957, Aug. 2016
DOI: 10.3837/tiis.2016.08.028

[4] R. Poisel and S. Tjoa, “Forensics Investigations of Multimedia Data: A Review of the State-of-the-Art,” in Proc. of 6th IEEE International Conference on IT Security Incident Management and IT Forensics (IMF), pp. 48-61, 2011
DOI: 10.1109/IMF.2011.14

[5] H.N. Lee, and D.H. Kim, “Video Streaming Receiver with Video Information File to correct Wrong Token Bucket Parameters by Packet Loss,” The Journal of Digital Contents Society, Vol. 17, No. 3, pp. 181-188, Jun. 2016.
DOI: http://dx.doi.org/10.9728/dcs.2016.17.3.181

[6] 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, Feb. 2014
DOI: 10.1109/TIP.2013.2285625

[7] Information Technology –Coding of Audio-Visual Object– Part 14: MP4 file format, International Standard ISO/IEC 14496-14:2003(E), *https://www.iso.org/standard/38538.html*

[8] Y.S. Choi, “A Reference Model for Representation of Video File Container Structure,” Dongduk Journal of Natural Science and Computer Science, Vol. 2, pp. 71-78, Dec. 2017

[9] Apple Developer, Quick Time File Format, *https://developer.apple.com/standards/qtff-2001.pdf*

[10] Y.S. Choi, “Design and Implementation of Video File Structure Analysis Tool for Detecting Manipulated Video Contents,” International Journal of Internet, Broadcasting and Communication(IJIBC), Vol. 10, No. 3, pp128-135, Aug. 2018
DOI: http://dx.doi.org/10.7236/IJIBC.2018.10.3.128

[11] Y.S. Choi, “A Study on Detecting Technique for Broken Parts of Damaged Video File,” Dongduk Journal of Natural Science and Computer Science, Vol. 3, pp. 63-70, Dec. 2018