STUDY ON VIDEO FILE RECOVERY AND CARVING IN VIDEO CODEC SPECIFICATION

N.Valarmathi,
M.Phil Research Scholar,
Department of Computer Science,
Sengunthar Arts and Science College,
Tiruchengode,Tamilnadu, India.

R.Bharathi,
Assistant Professor,
Department of Computer Science,
Sengunthar Arts and Science College,
Tiruchengode,Tamilnadu, India.

Abstract: In digital forensics, to recover the damaged or altered video file plays a crucial role in probing for evidences to determine a criminal case. File recovery techniques make use of the file system information that remains after deletion of a file. By using this information, many files can be recovered. For this technique to work, the file system information needs to be correct. If not, the files can't be recovered. If a system is formatted, the file recovery techniques will not work either. Carving deals with the raw data on the media and doesn't use the file system structure during its process. Although carving doesn't care about which file system is used to store the files, it could be very helpful to understand how a specific file system works. This paper presents a study of recovery and carving technique of a corrupted video file using the specifications of a codec used to encode the video data.

Keywords: File recovery, carving, frame, video encode and decode

I. INTRODUCTION

Recovery of corrupted or damaged video files has played a crucial role in role in digital forensics. Recently, a large amount of video contents have been produced in line with wide spread of surveillance cameras and mobile devices with built-in cameras, digital video recorders, and automobile black boxes. In criminal investigations, video data recorded on storage media often provide an important evidence of a case. As an effort to search for video data recorded about criminal, video data restoration and video file carving has been actively studied [1].

Data recovery is the process to restore damaged, failed or corrupted data from digital storage devices such as disks, tapes, and Compact Disks (CDs) [2]. In context of digital forensics, data recovery techniques are used to restore deleted, damaged, or hidden data in a controlled environment. A forensically controlled environment is the one where examination is conducted by trained examiners and all actions are taken with their permission [3]. Traditional techniques to recover data depend on file system of the underlying operating system. A file system is a hierarchical structure where information about each file and its associated data is present. [4].

Data carving techniques frequently occur during a digital investigation when the unallocated file system space is analyzed to extract files. The files are "carved" from the unallocated space using file type-specific header and footer values. File system structures are not used during the process. File carving is a powerful technique for recovering files and fragments of files when directory entries are corrupt or missing. The block of data is searched block by block for residual data matching the file type-specific header and footer values. Carving is also especially useful in criminal cases where the use of carving techniques can recover evidence.

II. LITERATURE REVIEW

Gi-Hyun Na, Kyu-Sun Shim, Ki Woong Moon, Seong G. Kong, Eun-SooKim, and Joong Lee [5] are researched,the frame based recovery of corrupted video files using video codec specification is given which uses a frame which is a meaningful measure of video files. Recovery of corrupted video files plays a crucial role in digital forensic. Many efforts have been taken to recover the video using a conventional video restoration of technique. This paper proposes a technique to restore the video data on a frame-by frame basis from its corrupted versions where the video data has been significantly fragmented or partly overwritten in the storage media. A video data consists of a sequence of video frames as the minimum meaningful unit of video file. The proposed method identifies, collects, and connects isolated video frames using the video codec specifications from non-overwritten portions of the video data to restore a corrupted video file.

A. Pal and N. Memon [6] discussed file carving process and various steps involved in the file carving during reconstruction of video files are mentioned. Data recovery is a key component of the disaster recovery, forensics, and ediscovery markets. Digital data recovery can consist of both software and hardware techniques. Hardware techniques are often used to extract data from corrupted or physically damaged disks. Once the data has been extracted, software recovery techniques are often required to order and make sense of the data. The various methods of data recovery are traditional data recovery, file carving, file systems and fragmentation, FAT32. File carving was born due to the problems inherent with recovery from file system meta-data alone.

R Poisel and S,Tjoa [7] are File carving is a recovery technique that recovers files based on information about their structure and content without matching file system information. As files can be recovered from their content and/or file structure this technique is indispensable during digital forensics investigations. So far many approaches for the recovery of digital images have been proposed.

G. G. Richard and V. Roussev [8] discussed about, video file can be restored using Bi-fragment Gap Carving. This method find a combination of the region containing the header and the footer to test if a video sample is valid. This computes the difference between the two data regions and check if the difference passes the predefined validation procedure. This procedure repeats until the gap passes the validation test. However, this method can only be applied to a video file with two fragments and this technique has limitation when the gap between the two file fragments is large. The file system meta-information contains the information such as the address and the link of a video file that can be used for file restoration. Utilizes additional information stored in the file to extend the idea to signature-based restoration techniques. For some files, file header may contain the information of file size or length. When the file footer does not exist, they can use this information to extract a file. Signature-based file restoration techniques search for the *start marker* (header) and the *end marker* (footer) to find valid connection of the regions containing the header and the footer. To increase the accuracy of the connection of the header and the footer regions, they used other information such as maximum size, embedded length recorded in the header. The analysis of the signature may offer a low success rate in video file restoration, when there are many file fragments and when some of them are overwritten. Especially, in the case a portion of a video file is overwritten, restoration of the video data using the file unit can be almost impossible because validation of restored file is failed by partially overwritten of restored file.

L.Aronson and J.Van Den Bos [9] are discussed File carving is the process of recovering files without the help of (file system) storage metadata. A host of techniques exist to perform file carving, often used in several tools in varying combinations and implementations. This makes it difficult to determine what tool to use in specific investigations or when recovering files in a specific file format. They define recoverability as the set of software requirements for a file carver to recover files in a specified file format. This set can then be used to evaluate what tool to use or which technique to implement, based on external factors such as file format to recover, available time, and engineering capacity and data set characteristics.

L.Laurenson [10] presented ―Performance analysis of file carving tool'. File carving is the process of recovering files based on the contents of a file in scenarios where file system metadata is unavailable. A new file carving data set was also authored and testing determined that the wider variety of file types and structures proved challenging for most tools to efficiently recover a high percentage of files. Results also highlighted the ongoing issue with complete recovery and reassembly of fragmented files. Future research is required to provide digital forensic investigators & data recovery practitioners with efficient and accurate file carving tools to maximize file recovery and minimize invalid file output.

III. FILE CARVING

File carving is a technique that utilizes information of internal file structure and contents of a deleted file for recovery [11]. File carving does not depend on the file system and can recover files out of the raw data set. Traditional recovery techniques are relatively fast as we can see the only processing involved behind this is reading the file system. Carving is used mostly for files that are in unallocated space. This is the area that doesn't have any metadata information referring it in the file system [11] [12].

An important concept in file carving is the handling of partial files also called fragmented files. There are different techniques implemented in operating systems to efficiently allocate blocks while creating a new file or adding data to an existing file. The operating system first searches for consecutive blocks but if not enough consecutive blocks are available then the file is stored on two or more locations. A file stored on multiple locations is called a fragmented file [13]. Common causes of fragmentation are low disk space and continuously appending more data to a file [11]. Traditional recovery techniques cannot recover a fragmented file if its metadata entry is not present that contains a link to blocks allocated to the file. With file carving it is possible to recover a file even if it is fragmented and stored on multiple locations in parts. This is because file carving techniques analyze a block or a set of blocks against characteristics of a specific file format and/or its contents [14].

a).Frame Size

Unlike other multimedia formats MPEG-1 frame headers do not contain any information that can be used to calculate size of the frame. Moreover frame sizes are variable and frames do not have an end code or a footer value that marks end of the frame. However the scenario is slightly different in our case as we are not interested to parse the whole sequence of frames but only individual I frames which are the first frames in a group of pictures. But still it is important in order to avoid losing valid frame data or addition of irrelevant data at the end of the frame. So the first requirement is to carefully locate end point of a frame[15].

b).Differences in Frame Structure

An MPEG- 1 file may consist of only a video elementary stream or a system stream. In both cases file structure is highly different and therefore I frames look differently. The two different structures are shown in figures 1 and 2 respectively. Figure 1 shows the first scenario where I frame is part of a video only MPEG-1 file. This is the simplest form of an I frame. The frame starts with a frame header and is followed by arbitrary number of slices each having its own header. After the frame header and before the first slice there can be arbitrary number of bytes of optional user data. I frame ends where the next frame header starts which are a P frame as shown in figure 1. Carving I frame in this case requires to verify the frame header, presence of any optional data, and then verifying slices. A slice can be decoded independently from other slices in the frame. To decode a frame successfully there should be at least one slice present. So we have to identify each slice by verifying its header.

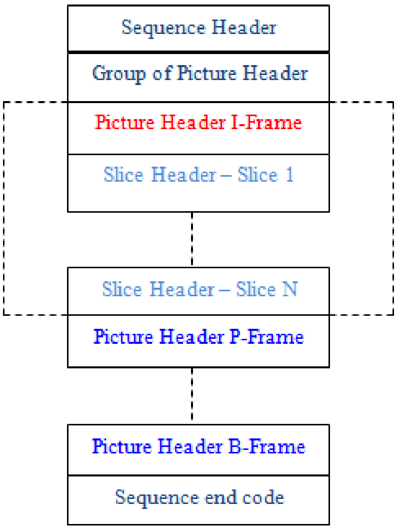


Figure 1: I frame structure in video elementary stream

Figure 2 shows the second scenario where I frame is part of a system stream. It shows how audio and video data is encapsulated into packets and then packs. The I frame in this case is divided among several video packets as highlighted in the figure. The first video packet contains Sequence header, GOP header and I frame header. Slice data is divided into these three video packets. Some of the slice data comes immediately after the frame header which is in the first packet. The second packet also contains some slice data. The last packet contains the remaining slice data of the I frame, header of the next P frame and some slice data belonging to the P frame.

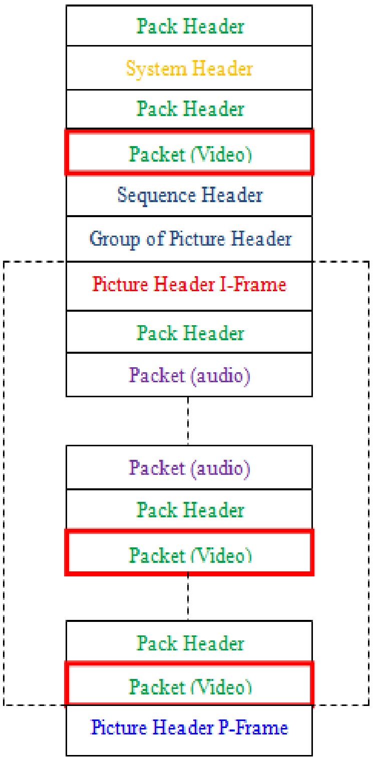


Figure 2: I frame structure in system stream

Like frames, a single slice may also span in more than one video packet. This is because a slice is of arbitrary size as already discussed in previous chapter. Additionally as during packetization process lower layers are not considered so we might also have slice headers at boundary of two PES packets. Figure 1 also shows that audio packets may come in between the video packets of a frame which further breaks continuity of a video frame. To carve I frames in this scenario first requires extracting all slices belonging to the frame spanned across different packets and then concatenating the frame header and slices together. This will result in an I frame having structure similar to the one shown in figure 3. This extraction is necessary to decode the frame. There are actually three decoders that work together to decode an MPEG-1 system stream. At first the system stream is passed through a decoder that handles the system layer and extracts the video and audio elementary streams. The video and audio elementary streams are then passed to video and audio decoders respectively. So we have to separate system layer information and audio data and extract only the video elementary stream part of the frame to pass it to the video decoder.

We have to consider both types of I frame structures to develop carving strategy. There can be multiple MPEG-1 files present in the raw dataset and the files may belong to either of the structures. So the requirement is to carve I frames from multiple MPEG-1 files present in the raw data set that includes both types of MPEG-1 file i.e. video only MPEG-1 and multiplexed MPEG-1 files.

c).Frame Resolution

In order to visually present a frame we have to decode it and then convert it to an image format such as JPEG or Bitmap. To decode a frame correctly we need to read the frame resolution information and pass it to the decoder. The frame resolution information is present in the sequence header in two fields i.e. width and height as already described in previous chapter under sequence header structure. Now there is no information present in a frame header or somewhere else in the MPEG-1 structure that tells which sequence a frame is part of so that we may identify the relevant sequence header and read the frame resolution information.

d).Handling Frame Resolution

As explained above, the carved frame is sent to the decoder along with frame resolution information in order to decode the frame successfully. We also mentioned that there is no information present in the frame header or anywhere else in the MPEG-1 structure that links a frame to its sequence. Nevertheless, in some cases we can use information of the internal MPEG-1 file structure to identify the sequence header of some I frames. This depends on two factors. The first factor is the position of the I frame in a GOP and the second factor is the number of GOPs in a sequence.

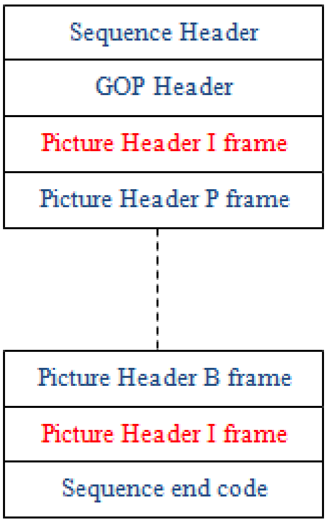


Figure 3: GOP structure with two I frames

Lets us consider the first factor. Normally a GOP contains only one I frame, but it is also possible to have more than one. Figure 3 shows a sequence with only one GOP. The GOP contains two I frames. Now it is possible to link the first I frame with its sequence header using some information of the MPEG-1 file structure. For each frame found in the raw data set we search the data set backwards within a certain threshold value to find the sequence header. If the sequence header is found in that range, we assume that the sequence header and the frame belong to each other. Then, we can read the resolution information from the sequence header and pass it to the decoder along with the frame.

IV.CONCEPTS OF VIDEO CODEC

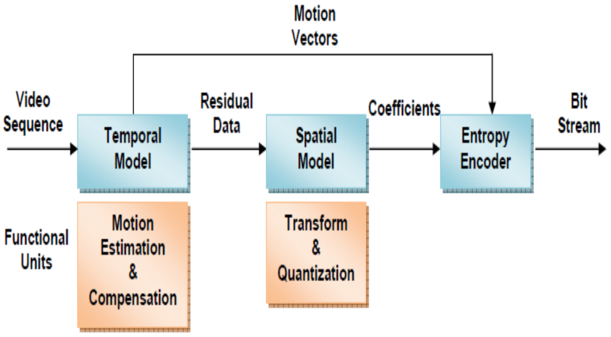
In this section, a general introduction of the video CODEC system is given, followed by a detailed discussion of a few particular functional units (e.g. motion estimation and compensation, transformation, quantization).

a). Redundancy

The source video is compressed by removing redundancy, which mainly has three types: statistical redundancy, temporal redundancy and spatial redundancy. In lossless compression systems, statistical redundancy is removed and the reconstructed signal is identical to the original one without any loss. However, this method can only achieve a modest amount of compression of video signals. In practice, a hybrid CODEC system based on lossy compression is commonly used. Besides removing statistical redundancy, it also reduces redundancy in temporal and spatial domains, taking advantages of limitations of the human visual system (HVS). For example, consider a video sequence captured by a digital camera at high frame rate. Neighboring frames may have little differences and smooth content of a frame has small variations in pixel values. Differences or variations that are unnoticeable by human eyes make up what is called "redundancy"

b). Encoder

Figure 4: Video Encoder



A video encoder (video encoder) includes three basic models: a temporal model, a spatial model and an entropy encoder. Each of them contains key functional units (Only those mentioned in this thesis are shown in Figure 4.

Temporal Model

Uncompressed video is fed to the temporal model, whose function is to search for similarities between adjacent video frames to reduce temporal redundancy. A prediction of the current frame is generated from previous or future frames (or from the combination of both previous and future frames). The prediction is often improved by compensating motion differences between relevant frames (by motion estimation and compensation). In the end, by subtracting the prediction form the current frame, a residual frame is generated and transferred to the spatial model, together with a set of motion vectors describing the motion.

Spatial Model

The spatial model is applied to find similarities between neighboring pixels within a residual frame, reducing spatial redundancy. This is achieved by transforming the residual frame into another domain, in which the residual data is represented by coefficients that are more independent with each other. Then insignificant coefficients values are removed through quantization, leaving only a relevant small number of significant coefficients to be sent to the entropy encoder.

Entropy Encoder

Quantized coefficients and motion vectors are compressed by the entropy encoder. It removes statistical redundancy by representing commonly-occurring vectors and coefficients by short binary codes (e.g. CABAC, UVLC, VLC). Finally a compressed bit stream is produced, whose format is standardized by video compression standards such as H.264. That includes header information, coded motion vector parameters and coded residual coefficients.

c). Decoder

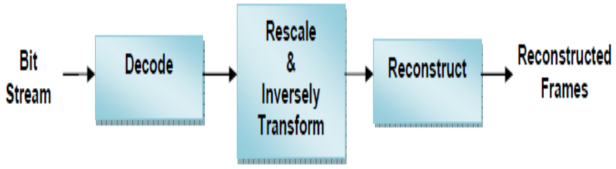


Figure 5: Video Decoder

A decoder (Figure 5) works similar to the encoder, but in a reverse way. When a standard bit stream comes into the decoder, the motion vectors and quantized transform coefficients are firstly decoded. Then the coefficients are rescaled (similar but not the same as the original ones since quantization is a lossy process) and inversely transformed back to the original domain, restoring residual data. Eventually, residual data will be added to a previously reconstructed reference frame, according to the decoded motion vectors, to reconstruct the current frame. The currently reconstructed frame is stored and might be used as reference frame for decoding future frames.

CONCLUSION

In this paper discussed about file carving and video codec specifications, File carving is a difficult task and does not guarantee 100 % recovery of deleted files even if deleted data is not overwritten. We introduce the concept of stream carving and discuss how streaming multimedia formats are suitable for carving due to the nature of their file structure. After recovering a video file from seized storage device it has to be analyzed manually in order to see if it contains any illegal contents. Since video files may consist of hundreds of frames depending on their length, manual screening is time consuming and intensive and significantly prolongs the duration of the forensic analysis. To overcome this limitation, we propose a carving approach for streaming video formats that can be used to automate analysis of recovered video files.

REFERENCES

[1]. Data recovery, [Online]. Available: http://en.wikipedia.org/wiki/Data_recovery

[2]. Data recovery,[Online]. Available: http://www.pcmag.com/encyclopedia_term/0,2542,t=data+recovery&i=40834,00.asp

[3]. Data recovery, [Online]. Available: http://www.afred.net/index_files/Page1109.htm

[4]. B. Carrier, *File System Forensic Analysis*. Boston, MA: Pearson Education, Addison-Wesley Professional, 2005.

[5]. Frame-Based Recovery of Corrupted Video Files Using Video Codec Specifications IEEE TRANSACTIONSON IMAGE PROCESSING,VOL. 23, NO. 2, FEBRUARY 2014 Author:Gi-Hyun Na, Kyu-Sun Shim, Ki Woong Moon,Seong G. Kong, *Senior Member, IEEE*, Eun-SooKim, and Joong Lee.

[6]. A. Pal and N. Memon, "The evolution of file carving," IEEE SignalProcess.Mag., vol. 26, no. 2, pp. 59–71, Mar.2009.

[7]. R. Poisel, S. Tjoa, and P. Tavolato, "Advanced file carving approaches for multimedia files," *J. Wireless Mobile Netw. Ubiquitous Comput., Dependable Appl.*, vol. 2, no. 4, pp. 42–58, 2011.

[8]. G. G. Richard and V. Roussev, "Scalpel: A frugal, high performance file carver," in *Proc DFRWS*, 2005, pp.1–10.

[9]. L. Aronson and J. Van Den Bos, ―Towards an engineering approach to file carver construction, in *Proc. IEEE 35th Annu. OMPSACW*, Jul. 2011, pp. 368–373.

[10]. T. Laurenson, ―Performance analysis of file carving tools, in Securityand Privacy Protection in Information Processing Systems. New York,NY, USA: Springer-Verlag, 2013, pp. 419–433.

[11]. Pal, N.Memon,. The evolution of file carving. In Signal Processing Magazine, vol. 26, no. 2, pp. 59—71. IEEE, 2009.

[12]. Christiaan Beek, "Introduction to File Carving", http://securitybananas.com/wpcontent/uploads/2010/05/Introduction-to-file-carving.pdf

[13]. Data carving, [Online]. Available: http://www.dfrws.org/2006/challenge/index.shtml

[14]. S.J.J. Kloet, "Measuring and Improving the Quality of File Carving Methods master thesis on File carving", Master's Thesis, Eindhoven University of Technology, 2007.