FRAME BASED RECOVERY OF CORRUPTED VIDEO FILES

D.Suresh[1], D.V.Ramana[2], D.Arun Kumar[3]

*[1]Assistant Professor, Department of ECE, GMRIT, RAJAM, AP, INDIA
[2] Assistant Professor, Department of ECE, GMRIT, RAJAM, AP, INDIA
[3] Assistant Professor, Department of ECE, GMRIT, RAJAM, AP, INDIA

ABSTRACT

In digital forensics, recovery of a damaged or altered video file plays a crucial role in searching for evidences to resolve a criminal case. This paper presents a frame-based recovery technique of a corrupted video file using the specifications of a codec used to encode the video data. A video frame is the minimum meaningful unit of video data. Many existing approaches attempt to recover a video file using file structure rather than frame structure. In case a target video file is severely fragmented or even has a portion of video overwritten by other video content, however, video file recovery of existing approaches may fail. The proposed approach addresses how to extract video frames from a portion of video to be restored as well as how to connect extracted video frames together according to the codec specifications. Experiment results show that the proposed technique successfully restores fragmented video files regardless of the amount of fragmentations. For a corrupted video file containing overwritten segments, the proposed technique can recover most of the video content in non-overwritten segments of the video file.

KEY WORDS

Digital forensics, Video, Frame-based, Codec, Fragmentation

1. INTRODUCTION

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. Recovery of corrupted or damaged video files has played a crucial role in role in digital forensics [1]–[3]. 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 [4]–[6].

2. PREVIOUS WORK

Recovery of damaged or corrupted video files obtained from a crime scene or a disaster site has provided a key evidence to resolve the cause. Conventional techniques for video file restoration use the meta-information of the file system to recover a video file stored in a storage medium such as a hard drive or a memory card [7]. 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. Carrier [7] proposes a file restoration tool based on the file system, which was implemented in a software toolkit, The Sleuth Kit [8].This program is based on the information from the file and directory structure of a storage file system. Video file restoration may not be possible with such techniques, however, when the file system meta-information is not available.

DOI:10.5121/Jmicro.2016.1102

The signature-based video restoration technique proposes File Carver [9] to address this problem. This method creates a database of the file header (beginning mark of file) and footer (the end mark of file), and define a set of rules for a specific file type. This method is limited to the cases when the files are not fragmented. This method does not recover partially overwritten video files. Garfinkel [10] 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. A video file can be restored using Bifragment Gap Carving [10]. 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.

Smart Carving technique was proposed to restore a file without being restricted by the number of fragments [11]. This technique, if it identifies the occurrence of fragmentation, combines the permutations of the fragment components and searches for the order of the fragments.

Most of previous technique bases its file restoration on a file unit, however, so only when a whole file is restored can the video be obtained. In general, the signature-based file carving techniques mentioned above consist of the following three steps [2].

1) **Identification Phase:** To identify a video fragment in a storage medium and to connect it to the previous fragment.

2) **Validation Phase:** To validate if all connected video fragments successfully form a playable video file.

3) **Validate by Human Expert:** To sort out false positive video segments by human expert.

The validation step checks if a restored video file is a playable video file. Conventional file-based video restoration techniques may fail to validate a restored video when a part of video is overwritten [12]. On the other hand, the proposed frame-based method carry out video restoration frame by frame, and is therefore applicable to restoration of partially overwritten video file.

3. PRESENT WORK

VIDEO FILE RESTORATION USING VIDEO CODEC SPECIFICATIONS

Video frame of a stored video file depends on the video codec used to encode the video file. And the video file that is encoded by codec also stored the decoding header information in start or end of video file. So that, the proposed method restores the video file using combination of frame data and decoding header information. The proposed technique applies to MPEC-4 Visual [13], popular video coding standards widely used in CCTVs, mobile devices, and automobile black boxes. For recover damaged or corrupted video, the proposed technique consists of two phases, extraction and connection as shown Figure 3.1

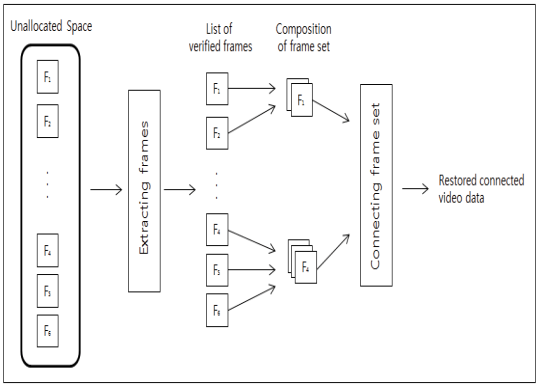


Fig. 3.1. Processing steps of the proposed frame-based video file restoration technique.

• **Extraction Phase:** The data are extracted based on video frame from the unallocated space, as extracted from the storage medium for restoration. The start code signature of video frame is searched for without considering the file system and the file composition. The frames are extracted based on the start code signature, the extracted frame data are verified through the decoder, and it is determined if the data are frames.

• **Connection Phase:** The codec and file specifications are used to connect the frames verified in previous phases. Based on the extracted frame sets, the length information of each frame recorded in the files is used to connect frame sets that are restored into a connected picture. Figure 3.1 shows an overall process of the proposed file restoration technique. In extraction phase, we extract frame data, F1, F2, F4, F5, and F6, which have a start code signature of frame from the unallocated space, the region of a video file to recover, containing the deleted video files and verify if the decoded frame is a normal frame data. Verified frames form a frame set, which will be connected as far as it can go in the stage of connecting frame set. When the video file is fragmented, we restore a video file by connecting fragmented pieces of data. In case of a partially overwritten file, not overwritten parts are connected to create a connected video. In this manner, the proposed method finds meaningful data in the video file using the codec and convert into file structure after connecting them.

Extraction of Video Frames:

A video file consists of a sequence of video frames, and each video frame is encoded into a binary data using a codec for data compression purpose. A codec inserts identifiers into each video frame to identify. The proposed method verifies if the data is a frame using the identifier characterized by a codec used in video encoding. Figure 3.2 shows the steps for extracting the verified frame data from a storage medium. Step 1 is to extract an unallocated space using file system meta-information. Because deleted file data could be stored in unallocated space. It is possible to reduce the amount of data which frame has to be analysed [14]. In practice, the popular forensic tools such as Encase [15] and WinHex [16] are used to extract unallocated space from storage medium.

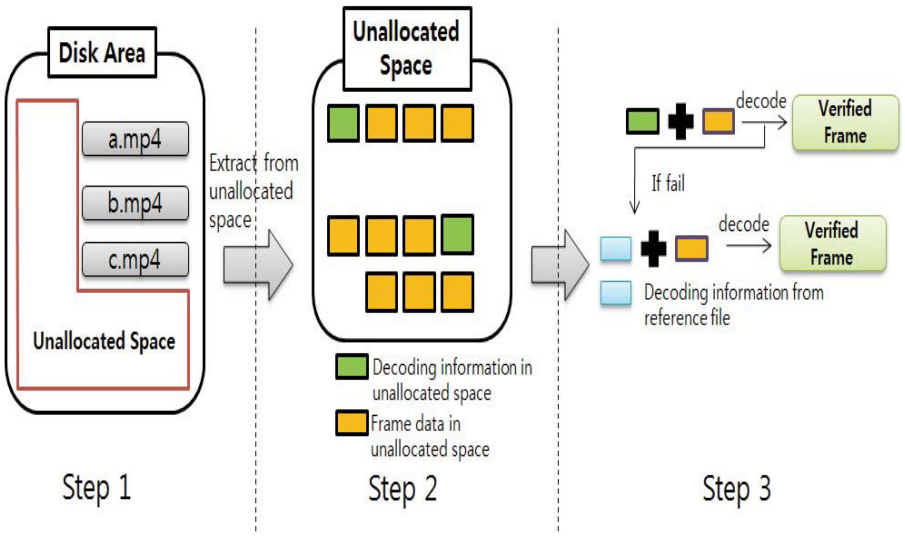


Fig. 3.2. The procedure of video file restoration using video file format specifications.

In Step 2, we extract the signature of the frame data from unallocated space extracted in Step 1. In figure 3.2, yellow rectangles indicate the frame data with signature. If the frame data is found in the unallocated space, we verify them by decoder. For verifying frame data, the decoding header is attached in front of frame data. So the proposed method also extract the signature of decoding header information from unallocated space. We search for decoding header marked in green in figure 3.2. In general, decoding header is usually recorded in the playback information and can be overwritten. In this case, if the decoding header encoded in the same manner as the file to be restored is found in an unallocated space, we can restore the video file. Even though the decoding header is not found, we can restore the video file using the decoding header of the reference file.2 Reference video file refers to a video file encoded in the same codec as the video to be restored.

In Step 3, we verify the frame data extracted by combining frame data and decoding header using the signature of each codec. The frame data, which cannot be decoded, are combined with the decoding header information of the reference video file to re-verify the decoding.

1) **Mpeg-4 Visual Frame Extraction:** MPEG-4 Visual is specified as a part of the MPEG-4 ISO/IEC standards 14496-2 [13]. The MPEG-4 Visual code begins with a start code signature (0x000001), and the next 1-byte indicates the type of the data that follows. For example, a code 0xB6 denotes the video frame. Then the start code of MPEG-4 Visual frame becomes 0x000001B6. In order to decode a data into a video frame, the decoding header information is needed. The decoding header involves the start code, followed by the video_object_layer_start_code 0x20-0x2F. And the portion that starts with 0x00000120-0x0000012F indicates the decoding header information. The frame data can be verified by decoding the frame data attaching the decoding header to front of them. Figure 3.3 shows the MPEG-4 Visual decoding information and frame data contained in the actual unallocated space.

13	00	00	01	B2	43	6F	72	65	6C	6F	67	69	63	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	01	00	00	00	01	20	00	C4	88
81	F4	51	68	43	C1	46	3F	00	00	00	18	66	72	65	65
48	52	41	32	58	42	4C	42	00	00	75	30	00	00	8C	A0
00	00	00	13	66	72	65	65	4E	6F	76	20	32	39	20	32
30	31	30	00	4D	D4	A1	6D	64	61	74	48	52	41	32	00
00	00	02	00	00	00	01	2E	DC	6C	D3	00	00	00	49	00
00	00	00	00	01	1B	20	00	00	01	B6	10	03	02	2C	33
6B	E9	F5	5F	55	E3	28	B0	3F	9C	00	6D	D5	7D	D3	34
FE	FB	C0	72	C4	D2	14	8A	13	AA	FA	0B	6A	D3	DE	DA
7B	3D	28	74	CD	57	D7	3F	C4	5E	FF	EB	EA	9E	D1	AA
BF	85	BD	F5	9E	90	7C	FF	75	AF	FA	7D	55	7D	16	BE
69	1C	2D	E7	DB	AF	A7	BB	A3	EC	FD	53	35	4C	CD	1A
82	DF	68	AF	9F	F0	F7	47	0B	59	F6	CF	FD	E3	F8	FE

Fig. 3.3. Example video data encoded using MPEG-4 Visual

For example, the red and blue boxes denote respectively decoding header information and the signature of video frame, MPEG-4 Visual video frames after 0x000001B6. The decoding header information signature (0x00000120-0x0000012F) is extracted from the unallocated space. After that, we search for the frame data information signature (0x000001B6). These two pieces of information confirms that the video was encoded by an MPEG-4 Visual codec. If frame data is verified by the MPEG-4 Visual decoder, the decoder returns the size of the frame. The returned size will be used to connect frame.

2) Connection of the Extracted Frames

The proposed technique, based on the verified frame data, forms frame set with physical locations of frame being continuous. The frame set compose verified frame in order before and after the relevant frame. The size information of each frame recorded in meta-information of the files with the stored video data are used to connect the frame sets. By connecting frame sets, the fragmented video frames can also be connected and restored.

Figure3. 4 illustrates an example of the composition of frame set using the verified frame.

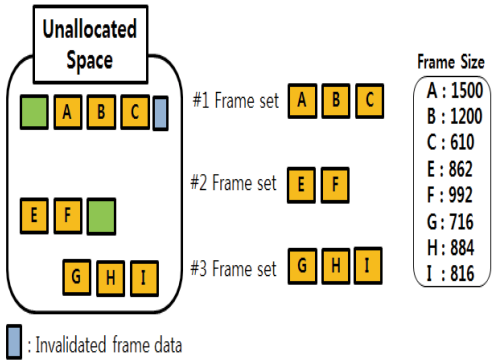


Fig. 3.4. Composition of Frame Set.

The yellow rectangle denotes verified frame through decoder, and blue rectangle denotes a frame data with a start signature but not verified by the decoder. It can be consider that the data is not frame but contain start signature of frame data, fortunately. If this data is frame, it can be considered as either fragmented or partially overwritten data. When invalidated frame data occurs in the first frame set, or the physical offset between frame data is long like between second frame and third frame, a frame set determined.

In figure 4, first frame set consists of the frame A, B, C and second frame set is frame E, F. The other frames are included third frame set. The frame size in right side results in decoding when verifying the frame. If the frame data is verified, the size of frame returns completely by decoder. And the verified frames are formed the frame set and it can be connected comparing size of frame data contained in frame set and the size information of each frame (STSZ box) which is contained in video files. The proposed technique connects frame sets using the one of the meta-information of video file. The connection phase uses the file meta-information based on the restored frame sets and restores the data into connected video. The video file meta-information includes the offset location, size, and other information for each frame. This paper only uses the size information of frame in a video file. For a MPEG-4 file, the size information of each frame is recorded in Sample-to- Size (STSZ) box. STSZ box is also found in an unallocated space in Step 1. The first four bytes of the signature starting with stsz denote the size of the STSZ box.

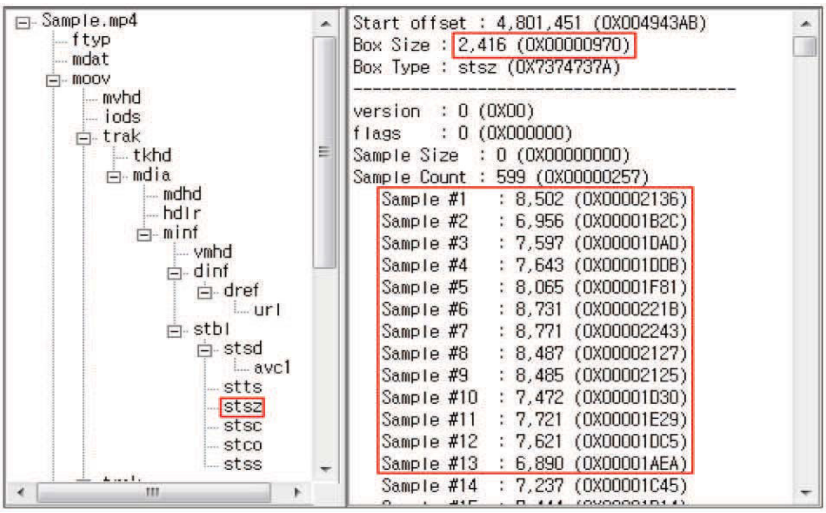
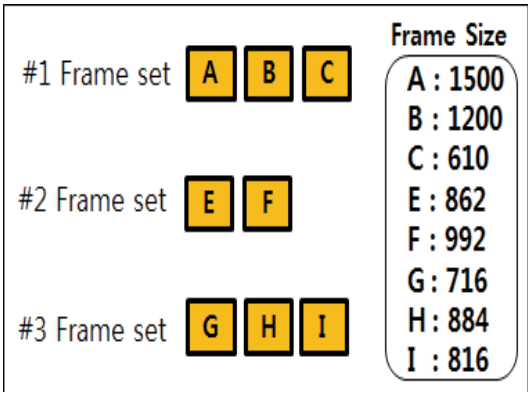


Fig. 3.5. STSZ box data in an MPEG-4 file.

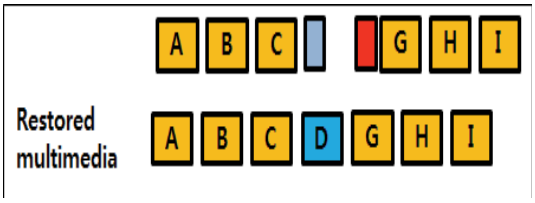
The most important thing is that this size-information (STSZ box) volume is not big, so it is less likely than frame data to be fragmented or overwritten by other data. However, STSZ box often comes at the end of an MPEG-4 file. In this case, STSZ box can also be fragmented regardless of the volume size, which makes it difficult to find all the STSZ boxes. Therefore, the proposed method connects the video frames without STSZ box. Figure3. 6 illustrate the example of connecting frame sets when the *STSZ* box is found in the unallocated space. Figure 3.6(a) is the formed frame set, and we compare the frame sizes generated by decoder and *STSZ* box from unallocated space described in figure 3.6(b). As mentioned above, *STSZ* box has the each frame size in file to recover, we can recover the video file by connecting frame set each other like figure 3.6(c).



(a)

```
Box Size : 48 (0X00000030)
Box Type : stsz (0X7374737A)
  Sample Count : 7 (0X00000007)
     Sample #1   :  1,500 (0X000005DC)
     Sample #2   :  1,200 (0X00000480)
     Sample #3   :  610 (0X00000262)
     Sample #4   :  1,615 (0X0000064F)
     Sample #5   :  716 (0X000002CC)
     Sample #6   :  884 (0X00000374)
     Sample #7   :  816 (0X00000330)
```

(b)



(c)

Fig. 3.6. Example restoring multimedia using *STSZ* box. (a) The Formed Frame set. (b) Size Index (*STSZ* Box) in Unallocated Space. (c) Restored multimedia.

The frame sizes of the frame A, B, C match with the sizes of sample #1, #2, #3 in figure 3. 6(b) and the frame size of the frame G, H, I match with the size of sample #5, #6, #7. So that we can assume that the frames can connect from sample #1to sample #7 except sample #4. In this situation, we illustrate the connected frame like top layer in figure 3.6(c). This does not include sample #4. However, we can infer a sample #4 from the rear part of the first frame set (blue rectangle in figure 3.6(c)) and the front reaming area in cluster of the frame G (red rectangle in figure 3.6(c)). It is because that the decoder does not verify the frame data in blue rectangle in figure 3.4 through its data start with start code of frame. So that, we attempt to connect the first frame set and the third frame set by combination of blue rectangle combines and the red rectangle. Generally, file systems allow the specification of a fixed record length called cluster or

block which is used for all write. From motivated it, blue rectangle and red rectangle are expended as long as exact size sample #4 from *STSZ* box. If the result of combination is verified by the decoder, the all the frames in *STSZ* box are connected perfectly like the bottom of figure 3.6(c). If not, we guess that the cluster located sample#4 is overwritten. So that we can restore two connected video as the first frame set and the third frame set.

We propose the method to connect frame set in case *STSZ* box is not found in the unallocated space as in figure 3.4.

We found the connected frame set that every frame set is 1:1 matching after that verified by decoder. In figure 3.7(a), the first frame set and the second frame set match each other according to cluster size.

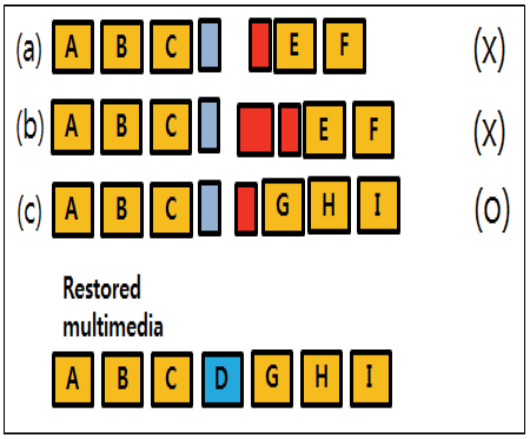


Fig. 3.7. Connection of the Verified Frame without a Size Index (*STSZ* Box).

The blue rectangle and the red rectangle have same meaning as the blue rectangle and the red rectangle in figure 3.6. If the combination of blue rectangle and red rectangle is verified by decoder, the matching process stops. If not, we extend the blue rectangle or red rectangle as long as cluster like in figure 3.7(b), and verified it. By setting a threshold, unlimited cluster expansion can be prevented. After extension, the verification by the decoder does not pass, the first frameset combine with next frame set. And the combination repeats this process until the verification success like in figure 3.7(c). We restored the connected multimedia without *STSZ* box. However, this process takes time since all frame set is being matched one by one. Despite time complexity, this method focuses to restore connected multimedia which is possible to record crime scene.

4. EXPERIMENT RESULTS

To evaluate the performance of the proposed technique, the restoration ratio was evaluated by following equation:

$$\text{Ratio}(\%)=100* \text{ No. of Restored Video Frames /No. of Total Video Frames} \text{-----(1)}$$

The number of restored frames is the number of frames extracted from the storage medium using the proposed technique, and the number of the original video frames is the number of the original video frames that were used in the experiment. If all the frames of the original video were restored, the restoration ratio would be 100%; and if none was restored, the restoration ratio would be 0%.

Results:

Size of original video file taken to recover: 73.6 MB.
Number of frames in the video: 8368
Duration of the video: 349 sec.

The following table shows the amount of overwriting and restoration ratio by keeping the number of fragmentations as constant. When the overwriting percentage is 10 then the restoration ratio was almost 88 percent and when the overwriting amount is 90 percent then the recovery ratio is nearly 9 percent.

Table1: Results of The Experiment

S.NO.	Number of fragments	Percentage of overwritten	Number of extracted frames	Restoration ratio
1	11	10.28	7405	88.49
2	11	22.33	6403	76.52
3	11	31.28	5649	67.51
4	11	43.52	4423	52.88
5	11	52.98	3243	38.75
6	11	59.78	2587	30.95
7	11	69.29	2095	25.03
8	11	79.97	1469	17.55
9	11	90.39	802	8.22

The below graph shown in the Fig4.1 is plotted between restoration ratio and amount of overwriting by keeping the number of fragments as constant. The decrease in the recovery ratio is observed as with increase in the overwriting.

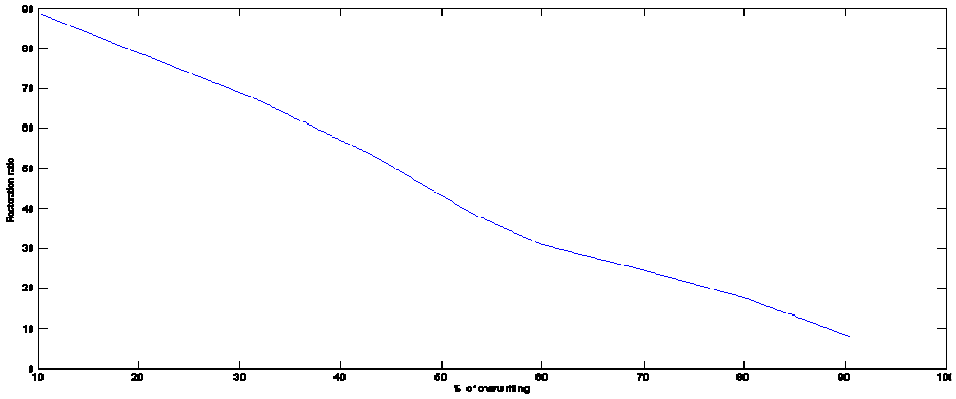


Fig 4.1. Graph between Percentage of Overwriting and Restoration Ratio

5. CONCLUSIONS

This paper presents a video restoration technique for fragmented and partially overwritten video files. The proposed technique guarantees the integrity of the restored frames because video files have the minimum number of frames to offer evidence. Large-size video files are often fragmented and overwritten. Many existing file-based techniques could not restore partially overwritten video files. Unlike most existing methods that use file format or file system meta-information, the proposed technique restores the data according to the minimum meaningful frame unit. Therefore, the proposed method restores almost frames in damaged or corrupted video files without being affected by the number of fragmentations. Especially, the proposed technique can restore the frames of the non-overwritten portions in partially overwritten files. Experiment results show that most of the frames were restored. The proposed frame-based file recovery technique increases restoration ratio.

REFERENCES

[1] K.MEDARIS AND R. MISLAN. (2008). EXPERT: DIGITAL EVIDENCE JUST AS IMPORTANT AS DNA IN SOLVING CRIMES [ONLINE] AVAILABLE: HTTP://NEWS.UNS.PURDUE.EDU/X/2008A/080425T-MISLANPHONES.HTML

[2] R.POISEL AND S. TJOA, "FORENSICS INVESTIGATIONS OF MULTIMEDIA DATA: A REVIEW OF THE STATE-OF-THE-ART," IN PROC. 6TH INT. CONF. IT SECURITY INCIDENT MANAG. IT FORENSICS, MAY 2011, PP. 48–61.

[3] H. T. SENCAR AND N. MEMON, "OVERVIEW OF STATE-OF-THE-ART IN DIG- ITAL IMAGE FORENSICS," ALGORITHMS, ARCHIT. INF. SYST. SECURITY, VOL. 3, PP. 325–348, NOV. 2008.

[4] L.HUSTON, R. SUKTHANKAR, J. CAMPBELL, AND P. PILLAI, "FORENSIC VIDEO RECONSTRUCTION," IN PROC. ACM 2ND INT. WORKSHOP VIDEO SURVEILL. SENSOR NETW., 2004, PP. 20–28.

[5] A. B. LEWIS, "RECONSTRUCTING COMPRESSED PHOTO AND VIDEO DATA," COMPUT. LAB., UNIV. CAMBRIDGE, CAMBRIDGE, U.K., TECH. REP. 813, 2012.

[6] R. POISEL AND S. TJOA, "ROADMAP TO APPROACHES FOR CARVING OF FRAG- MENTED MULTIMEDIA fiLES," IN PROC. 6TH INT. CONF. ARES, AUG. 2011, PP. 752–757.

[7] B.CARRIER, FILE SYSTEM FORENSIC ANALYSIS, VOL. 3. BOSTON, MA, USA: ADDISON-WESLEY, 2005.

[8] B. CARRIER. (2005). THE SLEUTH KIT [ONLINE]. AVAILABLE: HTTP:// WWW.SLEUTHKIT.ORG/SLEUTHKIT/

[9] G. G. RICHARD AND V. ROUSSEV, "SCALPEL: A FRUGAL, HIGH PERFORMANCE fiLE CARVER," IN PROC. DFRWS, 2005, PP. 1–10.

[10] S. L. GARfiNKEL, "CARVING CONTIGUOUS AND FRAGMENTED fiLES WITH FAST OBJECT VALIDATION," DIGIT. INVEST., VOL. 4, PP. 2–12, SEP. 2007.

[11] A. PAL AND N. MEMON, "THE EVOLUTION OF fiLE CARVING," IEEE SIGNAL PROCESS. MAG., VOL. 26, NO. 2, PP. 59–71, MAR. 2009.

[12] T. LAURENSON, "PERFORMANCE ANALYSIS OF fiLE CARVING TOOLS," IN SECURITY AND PRIVACY PROTECTION IN INFORMATION PROCESSING SYSTEMS. NEW YORK,NY, USA: SPRINGER-VERLAG, 2013, PP. 419–433.

[13] INFORMATION TECHNOLOGY—CODING OF AUDIO-VISUAL OBJECTS—PART 2: VISUAL, ISO/IEC STANDARD 14496-2:2004, 2004.

[14] 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.

[15] ENCASE [ONLINE]. AVAILABLE: HTTP://WWW.GUIDANCE-SOFTWARE.COM

[16] (2004). WINHEX [ONLINE]. AVAILABLE: HTTP://WWW.X-WAYS.NET/WINHEX/ INDEX-M.HTML