Performance Analysis of File Carving Tools

Thomas Laurenson

Department of Information Science, School of Business,
University of Otago. Dunedin, New Zealand.
thomas@thomaslaurenson.com

Abstract. File carving is the process of recovering files based on the contents of a file in scenarios where file system metadata is unavailable. In this research a total of 6 file carving tools were tested and reviewed to evaluate the performance quality of each. Comparison of findings to a previous similar study was conducted and showed variable performance advances. 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 maximise file recovery and minimise invalid file output.

Keywords: File Carving, Data Recovery, Digital Forensics

1 Introduction

File carving is a particularly powerful technique because computer files can be recovered from raw data regardless of the type of file system, and file retrieval is possible even if the file system metadata has been completely destroyed [1]. The process therefore provides additional data recovery methods to augment digital investigation where existing traditional data recovery techniques are not suitable or have been unsuccessful. Scenarios where file carving is exceptionally useful is when recovering data and files that have previously been deleted, extracting files from the unallocated space of a digital data storage device, and in cases when a storage device or a file system has been damaged or corrupted.

Previous research has advanced file carving techniques and algorithms resulting in newer state-of-the-art file recovery methods. Specifically, the Digital Forensic Research Workshop (DFRWS) conference promoted file carving techniques and tools by issuing a Forensic Challenge in 2006[1] and, again, in 2007[2]. The contests greatly extended file carving knowledge resulting in the discovery of new carving techniques and the release of associated tools.

[1] http://www.dfrws.org/2006/challenge/
[2] http://www.dfrws.org/2007/challenge/

Additionally, academic research has also contributed towards the improvement of file carving techniques, such as increasing file carving speed using GPUs [2], advanced file structure carving for binary file types [3], and multimedia files [4]. Furthermore, methods have also been developed for scenarios including carving network packets; e.g. IP packets from forensic images [5] and carving file objects from memory dumps [6]. Advanced file carving techniques have also been investigated including *in-place* file carving to reduce storage space and processing time [7] and recovery and re-assembly of fragmented JPEG files [8].

1.1 Problem

Digital Forensics is a relatively new discipline which presents numerous challenges for researchers and practitioners alike. Unfortunately, current research intended for forensic applications often has little or no impact, because in many instances the researchers are poorly acquainted with the *real-world* digital forensic problems encountered and the practical constraints frequently placed on investigators [9]. The solution is to conduct research which is *investigator-centric* with the aim of providing findings of practical usefulness lessening the gap between academic research and requisite real-world investigation tools and techniques. Furthermore, the targets of investigations are increasing in size and complexity [10]. Practitioners need informed results to confidently identify the correct tool for a specific scenario in order to decrease the overall case processing time while also maintaining investigation integrity.

File carving can be a difficult and complex process which is further complicated by the variety of available tools. Many forensic investigators are unaware of the capabilities and/or the limitations of the various file carving tools. Despite targeted active research a number of problems still exist for the professional digital forensic investigator or data recovery practitioner: which file carving tools provide the best performance in regards to *1)* the percentage of files recovered; *2)* the correctness and reliability of tool output; and *3)* the processing speed of the tool.

This paper aims to provide digital forensic practitioners with practical information and recommendations to assist in reliable and thorough implementation of file carving techniques. An additional goal is to identify current weaknesses in file carving techniques and tools so that future research areas can be targeted for technology advancement.

1.2 Structure

Firstly, the basics of file carving is described in order to understand the subject matter. A tool testing methodology is then outlined including data sets, performance measurements and a thorough testing procedure. Carving tool results from the testing phase are reviewed and findings are discussed. Finally, conclusions are drawn and areas for future research are suggested.

2 File Carving

File carving seeks to recover files based on content, irrespective of supporting file system metadata being available. The following subsections include a detailed summary of the various file structures, established file carving techniques and the associated file carving tools used by investigators to recover files in digital investigations.

2.1 File Structure

The structure of data in computer based systems is controlled by the file system allowing users the facility of long-term storage and retrieval of data in a hierarchy of files and directories [11]. Fig. 1 displays the 3 major types of file structures encountered: contiguous, fragmented and partial files. Embedded files are also discussed in this section.

Contiguous Files: A file is said to be *contiguous* when the data held in the file is stored in blocks in a logical order of sequence on the storage medium. The file is therefore stored in a single fragment occupying sequential file system clusters. A contiguous file ($FileA$) is shown in Fig. 1, which occupies 3 consecutive blocks spanning from block 4 to 6.

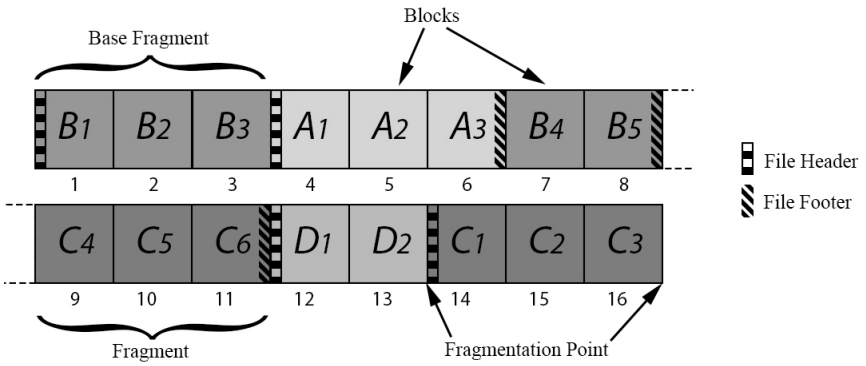

Fig. 1. A simplified diagram displaying various file structures where each square represents a single storage block. A contiguous (*File A*), linear fragmented (*File B*), non-linear fragmented (*File C*) and partial (*File D*) file structures are illustrated, where each block of a file is numbered consecutively (e.g. N_1, N_2 ... N_n). A base-fragment, file fragment and file fragmentation points are also displayed. (Source: Figure adapted from [8] and [12])

Fragmented Files: A file is *fragmented* when one or more chunks of the file are not stored in a sequential order and, thus, are comprised of two or more fragments separated from each other by an unknown number of clusters [12]. As files are added, deleted or modified the structure of a file system becomes divided and files may not be stored on consecutive clusters. Fragmentation in hard disks is therefore a result of the file system's allocation strategy, usually to optimise techniques such as fast file access and increased storage efficiency [13].

Fragmented files have a variety of different forms. However, files with 2 fragments, known as *bifragmented*, have been recognised as being the most common [14]. Fragmented files can be found in a linear and non-linear structure[3] depending on where the separate fragments are stored on the file system. Fig. 1 displays a linear fragmented file (*File B*) stored on a total of 5 blocks, separated by 3 blocks which are occupied by *File A*. The base-fragment of *File B* occupies blocks 1 to 3. For comparison, a non-linear fragmented file (*File C*) is also shown, stored on a total of 6 blocks. The base-fragment of this file occupies blocks 14 to 16 while the second fragment occupies blocks 9 to 11.

The issue of file fragmentation and the potential occurrence in actual investigations is debated among digital forensic researchers and professionals. Analysis of 324 second-hand hard drives showed that a total of 6% of all files recovered were fragmented [14]. Additionally, of all the fragmented files approximately 47% were discovered to be bifragmented. Although 6% seems a relatively small amount in general, it is highly significant that file types of forensic interest (e.g. AVI, DOC, JPG and PST) had considerably higher fragmentations than file types of little interest (e.g. BMP, HLP, INF and INI).

The availability and uptake of Solid State Drives (SSD) also has an impact on the level of fragmentation likely to be encountered. SSDs incorporate wear-levelling which results in files being moved more regularly and, although not yet proven, the probability is that SSDs would naturally be fragmented [12].

Partial files: As the term implies, *partial files* are incomplete files where some portion of the file is unavailable. The reason why partial files exist is due to a fragment of the original file being overwritten by other data. Fig. 1 displays a partial file (*File D*) occupying blocks 12 and 13 which lacks a file footer.

Embedded Files: When the contents of one file are added or stored in another file it is known as an *embedded file*. A common example is a JPEG image embedded within a Microsoft Word document or files embedded in an archive file; e.g. ZIP files. Embedded files can be contiguous, fragmented or partial depending on the scenario.

[3] Linear and non-linear fragmented files are also commonly referred to as sequential and non-sequential fragmentation files respectively.

2.2 File Carving Techniques

Previous research has identified various methods to perform file carving. An overview is provided outlining selected file carving techniques including header-based, file structure and block-based carving, as well as the role of file validation in the file carving process.

Header-Based Carving: Files have unique headers, also known as magic numbers or file signatures. These unique values can be used to help identify the beginning of a file and aid in carving files without the corresponding metadata. *Header-footer carving* is the most basic carving technique which searches data for patterns that mark a distinct header (start of file marker) and footer (end of file marker) [15]. The process is achieved by extracting all data contained within the headers and footers and copying that data into an external file.

An alternative header-based carving technique is *header-maximum size carving*. When a header is discovered (with no footer value available), the maximum carve size is used to calculate how far away from the header the end of the file might be [1]. As some file types can vary dramatically in size, this technique can have varying results and can also increase the size needed to store recovered files. However, it remains a viable approach because many file formats (e.g. JPEG, MP3) are not affected if additional data is appended to the end of a valid file [14]. Another header-based carving technique is *header-embedded length carving*. Some file formats have internal file information which specifies the length, or size, of the file and provides an identified point for the footer of the file [13].

File Structure Carving: Another file carving technique is based on the internal structure of a file, where specific knowledge of the contents can help reconstruct the original file. *File structure carving* is primarily aimed towards assembling fragmented files, where header-based carving fails to reconstruct multiple file fragments. An example is semantic carvers (also known as deep carvers) which use information about the internal file structure to control the carving process in some way [13].

Block-Based Carving: An advanced carving technique is *block-based carving* which calculates meta information of the content of a data block; for example, by implementing character counts or calculating statistical information [15]. The premise is that computer systems use fixed block sizes (sectors) for storing data (usually 512 bytes) and file carvers can examine every block for every file type definition [16].

File Validation: The method of *file validation* is an integral aspect of the file carving process. Validation provides the confirmation that the carved data actually results in a valid file output. Therefore, an automated format validator is a function that accepts a block of data and then determines whether it conforms to the defined structure of the file format before resulting in a validated file [17].

2.3 File Carving Tools

There is a wide selection of file carving tools available ranging from expensive proprietary forensic software suites (EnCase, FTK & WinHex) to open source software (Scalpel, Foremost & PhotoRec). A total of 6 file carving tools were selected for testing and are listed in Table 1. The basis for tool selection criteria included: wide file type support, advanced carving features and tool availability. Each tool listed has the associated license, tool version number and tool platform details. Additionally, the availability of tool configuration is also provided which illustrates the ability to modify the database of file signatures used by the tool.

Table 1. File Carving Tools Used During Testing

Name	License	Version	Platform	Configurable
EnCase	Proprietary	7.05	Windows	No
FTK	Proprietary	4.1	Windows	Yes
WinHex	Proprietary	16.8	Windows	Yes
PhotoRec	Open Source	6.13	Multi	No
Scalpel	Open Source	2.0	Multi	Yes
Foremost	Open Source	1.5.7	Linux	Yes

3 Tool Testing Methodology

In order to produce reliable and valid results a digital forensic tool testing methodology was used which implements function orientation testing to evaluate the ability of software tools to perform specific functions or tasks [18]. In this research the specific function to be tested is the ability of a file carving tool to recover assorted file types in various different scenarios. The following subsections outline the data sets, performance measurements and the testing procedure used during the experimental phase of this research.

3.1 Data Sets

Data sets in digital investigations and forensic research are usually comprised of a forensic image of a target device; for example, a bitwise copy of a computer's hard drive. However, in order to correctly evaluate file carving tools and produce reliable results, detailed knowledge of the data contained within the data set is essential. The use of documented data sets provide a baseline for scientific evaluation of tools and research reproducibility of useful findings to the academic community and practitioners alike [19].

Therefore, specific purpose based data sets for testing file carving tools were used. Each data set has extensive documentation including the following details: *1)* File name; *2)* File type; *3)* MD5 hash value; *4)* File location (offset);

and, if pertinent, *5)* File scenario. A total of 3 data sets were used to test tool performance:

1. Basic Data Carving Test #1 (11-carve-fat.dd)[4]
2. DFRWS2006 Forensics Challenge Data Set (dfrws-2006-challenge.img)[5]
3. Baseline Carving Data Set (bcds.raw)(see following section)

Baseline Carving Data Set: A new data set was created specifically for the second testing portion of this research. Justification for this is based on several limitations of data sets that are currently available. Firstly, the structure of files in the available data sets are not representative of, or in proportion with, data encountered in *real-world* investigations; for example, 11-carve-fat only contains contiguous files, while the DFRWS challenge data sets are predominantly fragmented (being designed to advance carving techniques, not test the performance capabilities of carving tools). Additionally, the variety of file types contained within the identified data sets are limited in scope.

The newly created data set was dubbed *Baseline Carving Data Set*[6]. The overall purpose is to represent a file structure that is indicative of what may be encountered in investigations in order to provide more viable carving performance results. It included numerous different user file types (a total of 25 different file types, and 67 files in total). Various file structures are also tested based on file sizes, fragmentation rates and gap sizes from an analysis of file systems from the wild [14]. The file types selected were classified into 4 distinct categories:

1. Documents: DOC, XLS, PPT, DOCX, XLSX, PPTX, PDF, TXT, HTML
2. Images: JPG, PNG, GIF
3. Multimedia: MP3, WAV, MPG, AVI, WMV, WMA, MOV, MP4, FLV
4. Archive: ZIP, 7ZIP, GZIP, RAR

The following file structures are to be tested: *1)* Contiguous files; *2)* Fragmented files; and *3)* Partial files. All documents and images used in the data set were sourced from the Digital Corpora, which provide an unrestricted file corpus[7] of 1 million *real* files sourced from web servers in the .gov domain and come with associated file metadata [19], while all multimedia and archive files were sourced from the public domain. This allows unrestricted distribution of the completed data set to other researchers or tool vendors.

[4] The Basic Data Carving Test #1 is authored by Nick Mikus and available from: `http://dftt.sourceforge.net/test11/index.html`

[5] The DFRWS2006 Forensics Challenge Data Set is authored by Brian Carrier, Eoghan Casy & Wietse Venema and avilable from: `http://www.dfrws.org/2006/challenge/index.shtml`

[6] The Baseline Carving Data Set is available from: `https://github.com/thomaslaurenson/`. All documentation including data set layout, hash sets, testing scenarios and file sources is also provided.

[7] Available from: `http://domex.nps.edu/corp/files/govdocs1/`

3.2 Performance Measurement

The performance of file carving tools can be measured based on the ability to recover correct files from a data set while avoiding the recovery of incorrect, corrupt or partial files. A widely known performance measurement used for Information Retrieval was applied to determine the performance of each tool. The measurements include versions of Recall, Precision and Fmeasure metrics which were modified specifically for tool testing performance[8]. The following 4 quality measurement metrics with associated symbols are defined below [20]:

$$carving\_Recall({}_cR) = \frac{all - sfn - ufn}{all} \tag{1}$$

$$supported\_Recall({}_sR) = \frac{sp - sfn}{sp} \tag{2}$$

$$carving\_Precision({}_cP) = \frac{tp}{tp + ufp + \frac{1}{2}kfp} \tag{3}$$

$$carving\_Fmeasure({}_cFm) = \frac{1}{\alpha\frac{1}{{}_cP} + (1-\alpha)\frac{1}{{}_cR}} \tag{4}$$

- All (*all*) refers to the total number of files in a data set.
- Supported files (*sp*) define the total number of file types in a data set supported by the specific carving tool.
- True positive (*tp*) is a file that is correctly carved from the data set.
- False positive is any carved file which is not a true positive. Known false positive (*kfp*) are files identified by the tool output as incorrect or corrupt, while unknown false positives (*ufp*) are false positives not identified as incorrect by the tool.
- False negative is the fraction of a file that was not correctly carved. A supported false negative (*sfn*) is the fraction of a file not carved by a tool, while an unsupported false negative (*ufn*) is a file type not supported by a tool.
- Alpha (α) is the factor used to assign weight to the relative importance of recall compared to precision. For this research $\alpha = 0.5$, meaning recall and precision each make up 50% of the importance of the Fmeasure metric.

The speed of processing a data set, measured in Megabits per second (Mb/s), will also be recorded to determine the time taken to perform file carving on the various selected data sets[9]. Furthermore, each test will be run 5 times to calculate an average processing speed.

[8] See ref. [20] Chapter 4 for additional information and reasoning behind the modified metrics to suit file carving performance measurement.

[9] To ensure viable processing speed results, all testing was conducted on the same computer system with the following specifications: Intel Core i5-3570K CPU with 8GB RAM and running either Backtrack Linux 5R3 or Mircosoft Windows 7 depending on the supported platform of each file carving tool.

Score Interpretation: The tool quality is tested and scored with a value between 0 (low) and 1 (high). Each of the 4 performance metrics and possible reason(s) of the resultant score are reviewed below [20]:

1. **carving_Recall:** Tests the ability of a tool to extract a high number of correct files from the data set. Low scores are either caused by unsupported file types, file structures or tool failure.
2. **supported_Recall:** Similar to carving recall, but determines the ability of a tool to extract a high number of supported file types only. Low scores are indicative of tool failure to extract only supported file types.
3. **carving_Precision:** Measures the correctness of the tool, where low scores are usually indicative of a large number of false positive files carved.
4. **carving_Fmeasure:** The results of the recall and precision scores are combined to provide an overall score for a tool, thus enabling indicative comparisons to be made.

3.3 Testing Procedure

A rigorous testing procedure was implemented to ensure that correct data collection and analysis was achieved in order to provide accurate results. At the outset each file carving tool was sourced and the tool documentation reviewed extensively. The selected tools were then run against the 3 specified data sets and results compared to the appropriate data set documentation. The specific testing procedure used in this research is adapted from 2 previous similar studies [20, 15] and is made up of the following phases:

1. **Determine true positives:** Calculate and compare MD5 hash values for all output from the file carving tool against the MD5 hash values from the data set documentation[10]. The remaining output files are then checked to determine if the carved file occupies the same block ranges as the file in the data set. If either of the 2 scenarios are true, files are marked as *tp* matches.
2. **Determine false negatives:** A combination of piecewise hashing [21] coupled with manual analysis was performed on tool output to determine any remaining false negatives and the fraction weight for files not already accounted for.
3. **Determine known false positives:** The log file created by each specific tool is then reviewed to identify any carved files which are marked as incorrect or corrupt. These files are marked as a *kfp* and counted accordingly.
4. **Determine unknown false positives:** The remaining output files are marked as *ufp* and counted accordingly.
5. **Calculate performance measurements:** The 4 performance metrics were then calculated using the defined formulae and the findings tabulated.

[10] The Hashdeep tool (http://md5deep.sourceforge.net/) was used to first create a list of the unique hash values of all files in the data set and then to compare the MD5 hash values to the list of known files. In digital forensic investigations this process is referred to as hash set analysis.

4 Carving Tool Review

Each selected file carving tool was run against the target data sets and the testing procedure implemented. Comparison of the results were made of the first two data sets against those of a previous similar study followed by testing results from the newly authored Baseline Carving Data Set.

4.1 Results & Comparison to Previous Research

The six file carving tools were each tested against the 11-carve-fat and the DFRWS2006 data sets. Tables 2 & 3 show the results as calculated from the defined performance measurements as well as the processing speed of the tool. The results were then compared to the previous findings collected by Kloet in 2007 [20]. An arrow is displayed to indicate either an increase or a decrease in the comparative performance score of each tool[11].

Table 2. File carving performance scores for 11-carve-fat.dd

Tool	Carving Recall	Supported Recall	Carving Precision	Carving Fmeasure	Processing Speed (MB/s)
EnCase	0.669	0.772	0.500	0.572	7.750
FTK	0.736 ⬆	0.736 ⬆	0.733 ⬇	0.735 ⬆	6.889 ⬆
WinHex	0.933	0.933	1.000	0.966	31.000
PhotoRec	0.933 ⬆	0.933 ⬆	1.000 ⬆	0.966 ⬆	20.667 ⬆
Scalpel	0.800 ⬇	0.800 ⬇	0.917 ⬆	0.854 ⬆	10.333 ⬆
Foremost	0.708 ⬇	0.708 ⬇	1.000 ⬆	0.829 ⬇	62.000 ⬆

A high overall performance was achieved by most tools on the 11-carve-fat data set, due to wide file type support and because only contiguous file structures make up the data set. WinHex and PhotoRec produced identical results and were noted for obtaining the highest performance scores, where only one false negative carving result was counted. Interestingly, all tools failed to carve a JPEG file with a corrupt header which demonstrates the importance of a complete and uncorrupted file header to allow correct file type identification from raw data.

In comparison to previous findings it was anticipated that there would be a widespread increase in tool performance. Both FTK and PhotoRec did have increased performance scores apart from the precision results from FTK which was caused by 9 false positive carved files. It was also discovered that decreases in performance were from tools with a highly editable configuration file (Scalpel & Foremost) and it is the author's opinion that a different method of tool configuration was possibly used in previous testing. This is justified by the lower

[11] For the 11-carve-fat.dd data set FTK, Scalpel, Foremost & PhotoRec have values to compare to previous results. For the dfrws-2006-challenge.raw data set comparative results are for FTK, Foremost & PhotoRec only.

recall but higher precision scores for Scalpel & Foremost. More file signatures could have been enabled during testing, but preliminary results indicated a very large number of false positives, thus, would have resulted in increased recall but decreased precision scores for each tool.

Each of the tools supported all 11 file types in this data set, apart from EnCase, therefore the carving recall and supported recall results are identical for each tool. Due to the very small size of the data set (62MB), the processing speed results are not conclusive findings of tool speed performance.

Table 3. File carving performance scores for dfrws-2006-challenge.img

Tool	Carving Recall	Supported Recall	Carving Precision	Carving Fmeasure	Processing Speed (MB/s)
EnCase	0.565	0.565	0.429	0.488	0.889
FTK	0.481 ⬇	0.513 ⬇	0.563 ⬆	0.519 ⬆	1.021 ⬆
WinHex	0.623	0.623	0.622	0.623	12.000
PhotoRec	0.813 ⬇	0.813 ⬇	0.963 ⬆	0.881 ⬆	0.980 ⬆
Scalpel	0.385	0.425	0.333	0.357	4.800
Foremost	0.546 ⬇	0.603 ⬇	0.341 ⬆	0.420 ⬆	9.600 ⬆

Although the DFRWS2006 data set contains only 6 different file types the image layout is significantly more complex than the 11-carve-fat data set. It includes 15 contiguous files and 17 fragmented files. Due to the difficulty of carving fragmented files, the scores for all tools were much lower. PhotoRec had the highest overall performance and extracted the most positive carving matches and lowest rate of false positive results. WinHex had the second highest overall Fmeasure score and the second lowest number of false positives.

Compared to previous findings all Fmeasure scores showed an increase indicating that the overall performance of file carving tools has improved for the scenarios in this data set. However, both carving recall scores were down albeit very close to previous findings. The decreases in performance may, again, be due to differences in tool configuration or operation varying between this research and the previous study. The exclusion of known bad file signatures recovering less true positive matches but also producing dramatically fewer false positive matches dictates higher precision but lower recall scores; e.g. by default Foremost has 3 file signatures for JPEG images one of which is known to produce high false positives but would have resulted in additional files recovered. Another potential reason for lower recall scores was that sector boundary scans (of 512 bytes) were specified during testing.

4.2 Baseline Carving Data Set Results

Slight changes were made to the testing procedure during testing the Baseline Carving Data Set as comparison of results to previous research was not necessary.

Firstly, carved results must validate in order to be counted as a true positive. A method known as *fast object validation* was implemented which attempts to open the file using it's native application without generating an error message, therefore, validating the carved file[14]. Secondly, the performance measurement scheme was updated to include the counting of true positives as a fraction, similar to the original method of counting supported false negatives as a fraction. The reasoning was that true positive matches are commonly carved as a fraction of a file, a notable example being a thumbnail image carved from a JPEG image which displays the original image but in a smaller file size quality. Additionally, with the updated procedure, true positives, supported false negatives and unsupported false negatives should always equal the total number of files in the data set. This can be summarised as: $tp + sfn + ufn = all$.

The results displayed in Table 4 show the performance results of each tool for each measurement metric along with the corresponding processing speed results. Additionally, optimised testing was performed for 2 of the tools identified as Scalpel Opt and Foremost Opt.

Table 4. File carving performance scores for bcds.raw

Tool	Carving Recall	Supported Recall	Carving Precision	Carving Fmeasure	Processing Speed (MB/s)
EnCase	0.390	0.413	0.093	0.150	0.029
FTK	0.445	0.508	0.098	0.160	0.714
WinHex	0.776	0.776	1.000	0.874	21.500
PhotoRec	0.825	0.825	0.938	0.878	3.822
Scalpel	0.428	0.453	0.004	0.007	0.068
Scalpel Opt	0.503	0.548	0.767	0.607	28.667
Foremost	0.421	0.452	0.004	0.008	0.065
Foremost Opt	0.539	0.587	0.694	0.607	24.571

Testing of the Baseline Carving Data Set revealed that the greater variety of file types and file structures proved difficult for most file carvers to efficiently extract a high percentage of files. Nevertheless, PhotoRec and WinHex were again the top performing file carving tools. PhotoRec had a slightly higher Fmeasure score due to obtaining a higher recall score. Both carvers also supported all 25 different file types, however, WinHex had a notably higher processing speed.

EnCase, FTK, Scalpel & Foremost all retained a high number of false positive files resulting in very low precision scores and in turn decreasing the overall Fmeasure result. The majority of the errors were caused by the MPEG file type, defined by short and very common header values which produced hundreds of false positive carved files. Both Scalpel & Foremost carved over 5,000 false positive MPEG files while FTK carved 300 false positive MPEG files. The low precision score by EnCase was due to carving numerous embedded files which were unable to be excluded using the embedded file hash set as most output files

were corrupt. FTK, Scalpel & Foremost also carved out embedded files which were able to be excluded using the embedded file hash set.

The default file signature databases (conf files) used by Scalpel and Foremost proved to greatly decrease performance scores, especially precision and processing speed, mainly due to excessive numbers of false positive carved files. Therefore, both file signature databases were optimised in an attempt to achieve better performance results. This involved adding new file signatures for Office 2007, HTML, MP4 & FLV file types and updating existing file signatures for JPEG, PNG & Office 2003 file types. Additionally, maximum file sizes were updated and the MPEG file signature was removed from the databases. As the results indicate both Scalpel Opt & Foremost Opt had a dramatic increase in Fmeasure score from 0.007 to 0.607 and 0.008 to 0.607 respectively which demonstrates the importance of tool configuration and file signature databases used.

The use of a significantly larger data set (237MB) and complex file structure give a better understanding of the processing speed of the 6 tools. The results indicate that, as expected, processing speed decreases greatly as the number of false positives increase. This is specifically caused by the time required to write false positive file matches to permanent disk storage.

4.3 Discussion of Findings

The experiment results highlight numerous insights into the current performance of file carving tools in terms of capabilities and limitations. One of the most important factors is a detailed knowledge of tool configuration and the selection of file types, or signatures, chosen by the investigator for potential recovery; for example, MPEG and ZIP files proved difficult to carve without numerous false positives due to common header values. In this scenario manual analysis, or a specialised carver developed for a specific file format, may be implemented to enhance file recovery. However, with the increasing sizes of targets being investigated, such detailed analysis may be hindered by technical or time frame limitations.

Another important configuration option is the specification of sector size. Targeting the beginning of a sector offset for file headers greatly reduced false positive results for all tools where this option was available. However, enabling sector scanning also has the limitation of potentially missing files of interest and, as this research discovered, embedded files could not be recovered separately from the original container file.

The selection of the right tool for the job at hand is essential; for example, Scalpel uses header-based carving very efficiently with high computational performance on large data sets whereas PhotoRec uses predominantly structure-based carving with potentially lower computational performance. However, PhotoRec results illustrated that a higher percentage of correct files was usually carved while also minimising false positives. Knowledge of specific file types and associated file structure also contributes to more efficient tool usage and improved carving results.

In terms of file structure the results reinforce that contiguous files are much simpler to carve compared to fragmented files. Nevertheless, most tools were found capable of extracting the base fragment of a high percentage of fragmented files from each data set. Again, implementation of manual analysis may then provide complete file recovery. The inability to reconstruct file fragments, despite advanced academic research and proof of concept software development, is potentially troublesome to practitioners. However, it was identified that the type of data separating fragments is an important factor in how well a carver could extract the fragmented file; for example, an HTML file intertwined with a TXT file separating the two fragments was difficult to recover whereas a JPEG document separated by randomly generated data or another JPEG proved simpler to recover.

There are numerous factors influencing the processing speed of a file carving tool and it can be highly dependant on the avoidance of carving and writing false positive files. Additionally, the file carving technique used will also affect processing speed. In general, the more complex the extraction method, the slower the processing speed. Another factor is the scan type selected on tools; for example, PhotoRec has a *brute force mode* scan which greatly increases scan time while Foremost & Scalpel have a *quick mode* which decreases scan time by only searching the start of a block of the input file as specified by the investigator.

5 Conclusion

In conclusion, this research investigated the performance of 6 file carving tools by conducting testing on various data sets and analysing tool output. The fact remains that there is no single best file carver. However, informed selection of the correct tool for a specific task plus knowledge of tool configuration stand out as the most important aspects to increasing both the number of files recovered and the reliability of tool output. Additionally, the findings highlight the ongoing issue and limitations of reconstructing file fragments.

A selection of available tools has also been compiled from this research to promote and advance future research. A new file carving data set, post processing file validation scripts and tool configuration files have been authored and made available for use[12].

Future advancement of techniques and tools based on academic research could greatly improve the performance of file carving tools. Implementation of advanced data abstractions to store file carving metadata, continual advancement of file validation techniques and the automated post-processing of carving output can all help to increase file carving file performance. Additional research is also needed to reverse-engineer new file types to support the carving process. File carving, however, remains a valuable technique enabling the recovery of files and the retrieval of potential evidence for digital investigations.

[12] All files are available from: `https://github.com/thomaslaurenson/`

References

1. Richard III, G.G., Roussev, V.: Scalpel: A Frugal, High Performance File Carver. In: 2005 Digital Forensic Research Workshop, New Orleans, LA (2005)
2. Marziale, L., Richard III, G.G., Roussev, V.: Massive Threading: Using GPUs to increase the performance of digital forensics tools: Digital Investigation 4(Supplement), 73-81 (2007)
3. Hand, S., Lin, Z., Gu, G., Thuraisingham, B.: Bin-Carver: Automatic recovery of binary executable files: Digital Investigation 9(Supplement), 108-117 (2012)
4. Yoo, B., Park, J., Lim, S., Bang, J. and Lee, S.: A Study on Multimedia File Carving Method: Multimedia Tools and Applications, 1-19 (2011)
5. Beverly, R., Garfinkel, S., Cardwell, G.: Forensic Carving of Network Packets and Associated Data Structures: Digital Investigation 8(Supplement), 78-89 (2011)
6. van Baar, R.B., Alink, W., van Ballegooij, A.R.: Forensic Memory Analysis: Files mapped in memory: Digital Investigation 5(Supplement), 52-57 (2008)
7. Richard III, G.G., Roussev, V. and Marziale, L.: In-Place File Carving: Advances in Digital Forensics III, 217-230 (2007)
8. Sencar, H.T., Memon, N.: Identification and Recovery of JPEG Files with Missing Fragments: Digital Investigation 6(Supplement), 88-98 (2009)
9. Walls, R.J., Levine, B.N., Liberatore, M., Shields, C.,: Effective Digital Forensics Research is Investigator-Centric. In: 6th USENIX Conference on Hot Topics in Security, pp. 1-11. USENIX Association, San Francisco, CA (2011)
10. Garfinkel, S.L.: Digital Forensics Research: The Next 10 Years: Digital Investigation 7(Supplement), 64-73 (2010)
11. Carrier, B.: File System Forensic Analysis. Addison-Wesley Professional (2005)
12. Pal, A., Memon, N.: The Evolution of File Carving: IEEE Signal Processing Magazine 26(2), 59-71 (2009)
13. Cohen, M.I.: Advanced Carving Techniques: Digital Investigation 4(3-4), 119-128 (2007)
14. Garfinkel, S.L.: Carving Contiguous and Fragmented Files with Fast Object Validation: Digital Investigation 4(Supplement), 2-12 (2007)
15. Tomar, D.S., Malviya, O., Verma, R.: Analysis Framework for Quality Measurement of Carving Techniques. In: 6th National Conference on Emerging Trends in Computing and Communication, pp. 421-426. Hamirpur, India (2008)
16. Metz, J., Mora, R.J.: Analysis of 2006 DFRWS Forensic Carving Challenge, `http://www.dfrws.org/2006/challenge/submissions/mora/dfrws2006.pdf`
17. Aronson, L., Van Den Bos, J.: Towards an Engineering Approach to File Carver Construction. In: 35th IEEE Annual Computer Software and Applications Conference Workshops, pp. 368-373. IEEE, Munich, Germany (2011)
18. Lyle, J., White, D., Ayers, R.: NIST Internal Report 7490: Digital Forensics at the National Institute of Standards and Technology. National Institute of Standards and Technology, Gaithersburg, Maryland (2008)
19. Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G.: Bringing Science to Digital Forensics with Standardized Forensic Corpora: Digital Investigation 6, 2-11 (2009)
20. Kloet, S.J.J.: Master's Thesis: Measuring and Improving the Quality of File Carving Methods. Eindhoven University of Technology (2007)
21. Kornblum, J.: Identifying Almost Identical Files using Context Triggered Piecewise Hashing: Digital Investigation 3(Supplement), 91-97 (2006)