


Digital Forensics
Dr. Joseph Vella
Dept. of Computer Info. Systems, FICT, UOM

FILES: HIDING & RECONSTRUCTION

What if our data file *undelete* or *restoration* attempts have not succeeded?

What if data files have not been deleted but rather obfuscated?



J Vella – Digital Forensics

Files: Carving & Slack Space 2

Techniques:

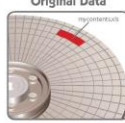
File Carving & Slack Space Use

J Vella – Digital Forensics

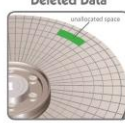
Files: Carving & Slack Space 3

How are Deleted Files and Data Recovered?

Computers Don't Immediately Remove Data that is Deleted

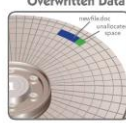


Original Data



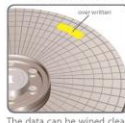
Deleted Data

The original data is still present, but marked as unallocated space.



Partially Overwritten Data

Over time, some or all of the data can be overwritten. The remaining data can still be "carved" and retrieved.



Data Wiped Clean or Shredded

The data can be wiped clean or shredded using privacy software.

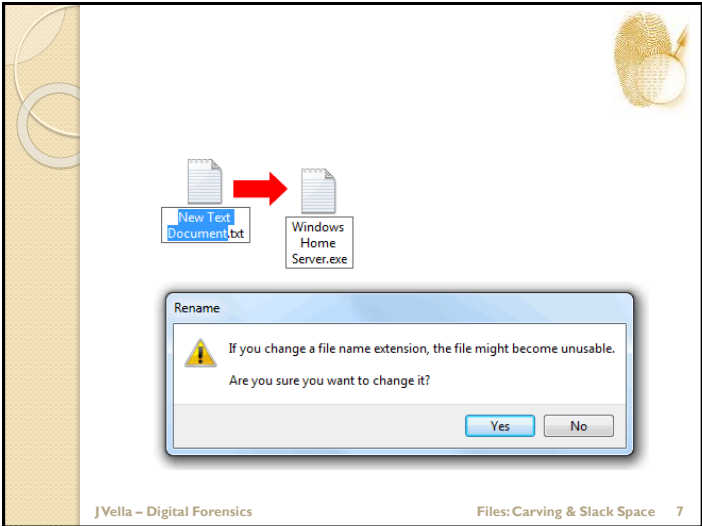
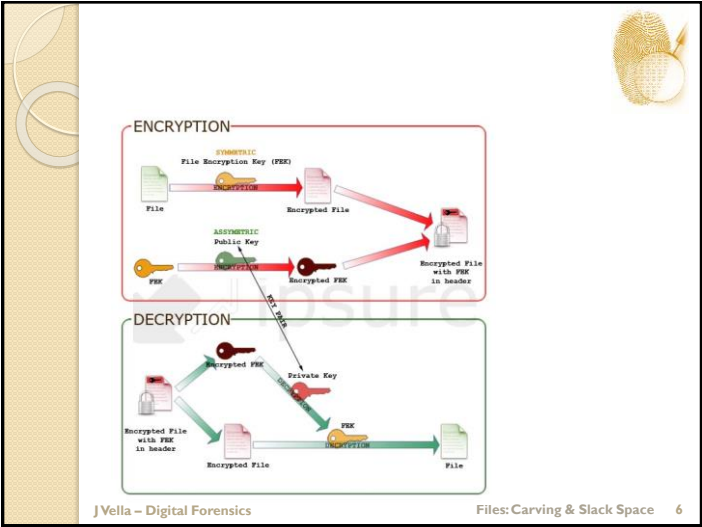
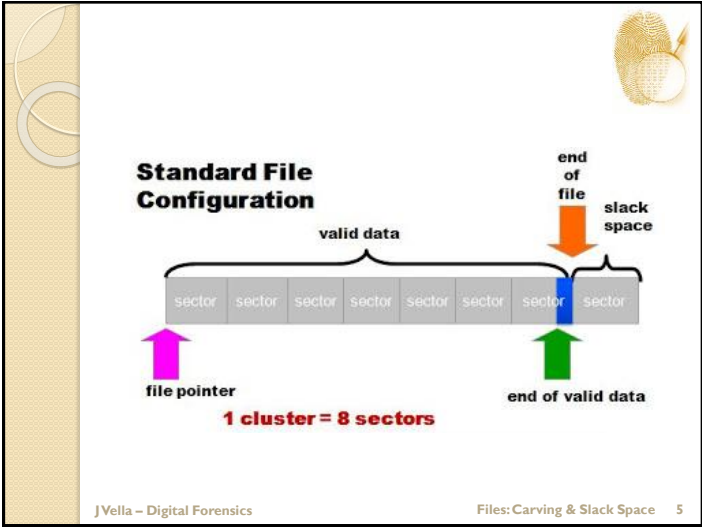
What is unallocated space?

Unallocated Space is available disk space that is not allocated to any volumes. The type of volume that you can create on unallocated space depends on the disk type. On basic disks, you can use unallocated space to create primary or extended partitions. On dynamic disks, you can use unallocated space to create dynamic volumes.

PINPOINT
LABORATORIES
©2008 Pinpoint Guidance
www.pinpointlabs.com

J Vella – Digital Forensics

Files: Carving & Slack Space 4



Also possible is a combination of

- Rename and compress;
- Slack space filled with compressed data;

J Vella – Digital Forensics Files: Carving & Slack Space 8

FILE CARVING IN MS WINDOWS

J Vella – Digital Forensics

Files: Carving & Slack Space 9


File Carving

- **Carving** is a technique for extracting files from a file system.
 - It is important to add that the process *does not depend* on the file system data and its meta data;
 - The *files of interest* are those:
 - that have been previously purged (found in the unallocated page list); or
 - Specific file corruption; or
 - not reachable from the file systems (e.g. corrupted file system).
- The **aim** of carving is:
 - Provide efficient and accurate file carving tools to max recovery;
 - Whist minimising invalid file output.
- Sometimes it's acceptable to extract a partial part of the original file:
 - E.g. mbox, image, video clip.

J Vella – Digital Forensics

Files: Carving & Slack Space 10

Example of corrupted file



J Vella – Digital Forensics

Files: Carving & Slack Space 11

Carving Methods

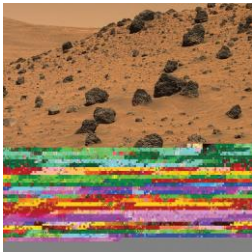
- There are a number of **file carving methods**:
 - Basic ones are based on a brute force traversal of unallocated space and match and string up disk blocks to form the original files.
 - A basic technique is **matching file type specific keys** e.g. header and footers bit patterns.
 - Or header plus a file size (length).
 - **File-structure** based where intricate details of file structure and actual file are extracted and this structure leads the process.
 - Parts might neither be sequential or all present
 - These are sometime called **deep carvers**.
 - **Data content** leads the selection process to string one block to a previous chain:
 - data content could be interspersed in MBOX, XML, etc
 - Statistical attributes of data;
 - Known language and subject matter.

J Vella – Digital Forensics

Files: Carving & Slack Space 12

Carving Validation

- It is *not* certain that an output of a carve is a valid file!
 - Therefore a **confirmation process** that automatically checks the generated file is most required.
 - It's easy to check visually an image but subtle differences might not be easily discernible.
 - How about the content of a huge database!?



File Carving vs File Recovery

- In file recovery the techniques are based on:
 - File system meta data and almost intact preservation of original file data blocks;
 - File retrieval from data backups.
- But file recovery does not work when:
 - Original data files space has been actually used in the mean time – data blocks from the original file have been overwritten;
 - File system data is not available or correct.
- File carving reads the media in **raw mode** – i.e. assumes little of filling system structure, meta-data, etc.
 - Nonetheless carving success is aided by:
 - Underlying file system details and workings;
 - Data of file structure details (i.e. internal file structure) of the file being currently carved.

Magic Numbers

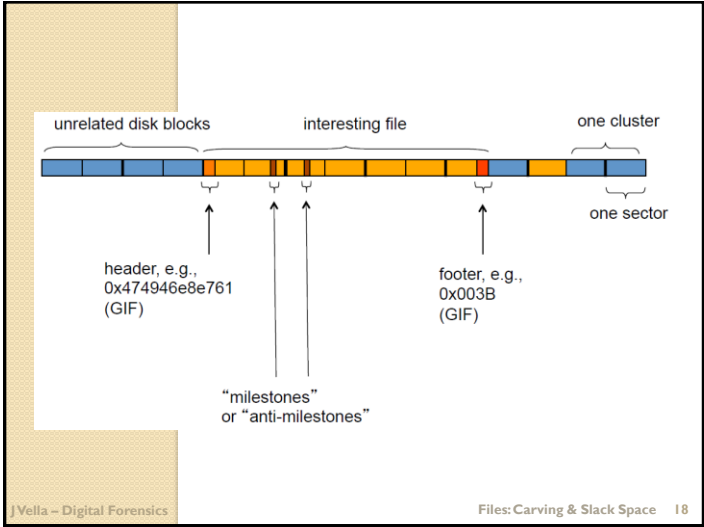
- **Magic number** is a constant used to identify a file format.
- Examples:
 - **PDF file type**
 - Start: %PDF
 - End: %EOF
 - **JPEG file type**
 - Start: 0xFFD8
 - End: 0xFFD9
- On Linux use the **file** command to determine magic number (and more) – database is usually stored locally:

```
$ file file.c
file.c: C program text
root# more /usr/share/file/magic
...
# pdf: file(l) magic for Portable Document Format
#
0  string  %PDF-   PDF document
>5  byte   x       \b,version %c
>7  byte   x       \b.%c
...
```

File carving terminology based on Pal et al.

Term	Definition
Block	The size of the smallest data unit that can be written to storage media. It refers to either the sector or the cluster size.
Header	Header blocks contain the starting point of a file.
Footer	Footers contain the ending point of a file.
Fragment	One block or a sequence of blocks that belong to one file. One file can be built from different fragments which are not sequentially connected to each other. The distance between different fragments of one file is unknown, further it is possible that fragments do not exist anymore because they have been overwritten.
Base-fragment	The first fragment of a file. It contains the header (if available for the filetype investigated).
Fragmentation point	The last block of a file before fragmentation occurs. As a file can consist of multiple fragments it is possible that there exist multiple fragmentation points.
Fragmentation area	Consecutive blocks which are grouped into a set and which contain the fragmentation point.

HEADER-FOOTER CARVING



JPEG file internal structure

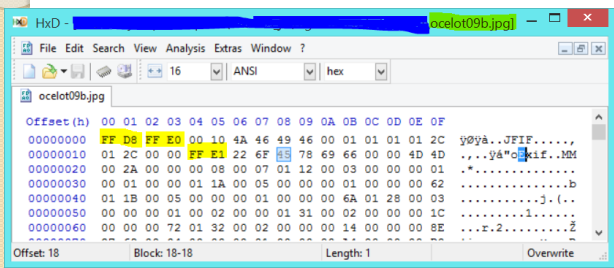
- Every file type has a header (i.e. hex sequence) – magic number.
 - This **header** is also used as start of image (SOI) – 0x FF D8.
 - The **footer**, called the end of image (EOI), is – 0x FF D9.
 - After the SOI is a sequence of file information blocks – field specifiers are hex 0x FF C0 to 0x FF DD.
 - Each specifier is qualified by two bytes to represent size.
 - After the size there is the data.
- The actual image starts at the signature 0x FF DA (SOS).
 - Some data files might have a **thumbnail** – again it's sandwiched between another SOI & EOI.

Short Name	Bytes	Payload	Name
SOI	0x FF D8	none	Start of Image
SOF0	0x FF C0	variable size	Start of Frame (Baseline DCT)
SOF2	0x FF C2	variable size	Start of Frame (Progressive DCT)
DHT	0x FF C4	variable size	Define Huffman Table(s)
DQT	0x FF DB	variable size	Define Quantization Table(s)
DRI	0x FF DD	2 bytes	Define Restart Interval
SOS	0x FF DA	variable size	Start of Stream
RSTn	0x FF D0...0x FF D7	none	Restart
Appn	0x FF En	variable size	Application-Specific
COM	0x FF FE	variable size	Comment (text)
EOI	0x FF D9	none	End of Image

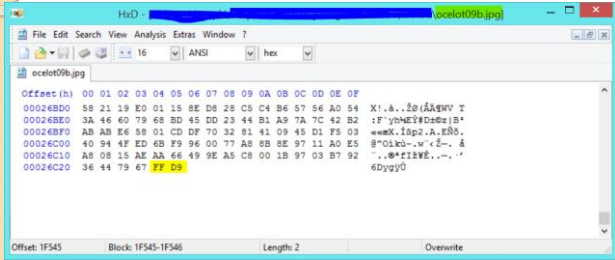
Ocelot wild cats from S America



Open any JPEG file, e.g. ocelot09b.jpg provided, with hex editor and search for hex code &FF D8 (i.e. SOI)

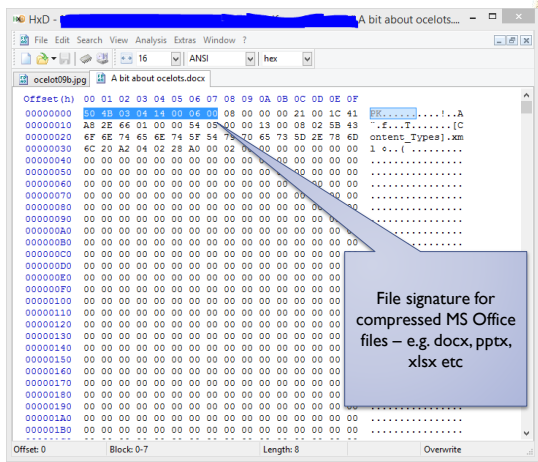


Ensure that end of file is marked with a JPEG EOI marker (i.e. hex & FF D9)



Dealing with Embedded Files

How about extracting a JPEG from a MS word doc (e.g. provided “A bit about ocelots.docx”)?



Files: Carving & Slack Space 25

Files: Carving & Slack Space 20

Files: Carving & Slack Space 27

Files: Carving & Slack Space 2

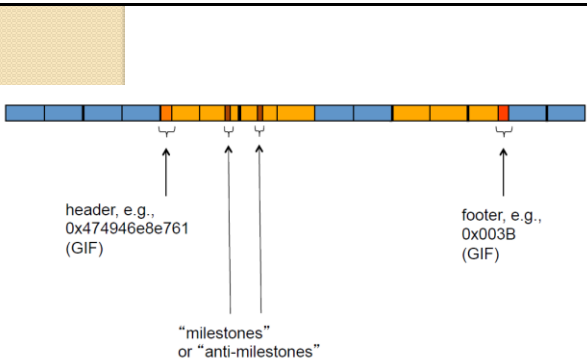
Header-Footer Carving: Summary

- As we have seen this type of carving is really simple and straightforward.
- But:
 - The files to carve *are not fragmented* into a sequence of clusters;
 - The first block (and beginning of file) *is available*; and
 - The signature being searched *does not match* with data (i.e. false positives).

Short Name	Bytes	Payload	Name
SOI	0x FF D8	none	Start of Image
SOF0	0x FF C0	variable size	Start of Frame (Baseline DCT)
SOF2	0x FF C2	variable size	Start of Frame (Progressive DCT)
DHT	0x FF C4	variable size	Define Huffman Table(s)
DQT	0x FF DB	variable size	Define Quantization Table(s)
DRI	0x FF DD	2 bytes	Define Restart Interval
SOS	0x FF DA	variable size	Start of Stream
RSTn	0x FF DO...0x FF D7	none	Restart
Appn	0x FF En	variable size	Application-Specific
COM	0x FF FE	variable size	Comment (text)
EOI	0x FF D9	none	End of Image

File Fragmentation

- Files whose data clusters are not contiguous are said to be fragmented.
- Why are files fragmented:
 - Data is appended to a file and eventually use the last cluster. If a contiguous cluster is unavailable (e.g. allocated to another file) then fragmentation occurs,
 - A new file requires a certain number of clusters but these no contiguous space is available to house these – but a number of smaller clusters are available,
 - The filling system allocation and growth mechanism management might introduce fragmentation,
 - The devices, e.g. SSDs, incorporate wear-leveling file allocation and placement.



Simon Garfinkel research on fragmented files

- Study covers 350 disks with:
 - NTFS, FAT & UFS.
- Common application files are highly fragmented:
 - Email box (@58%)
 - JPEGs (@16%)
 - MS Word (@17%)
 - MS Excel
- Many fragmented files are found in two parts (ie fragments) – 47% actually.
- Digital forensically uninteresting files show little fragmentation:
 - E.g. ini, help.

Carving fragmented files

- When no file type signature or no beginning block is known then more elaborate carving techniques are required:
 - The next technique to cover is based on file type structure:
 - E.g. in case of JPEG files we use the signatures (and their syntax) that unfurl with the file type. A field token together with its size is also used.



J Vella – Digital Forensics

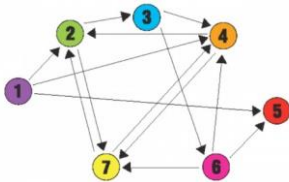
Files: Carving & Slack Space 33

GRAPH THEORETIC CARVERS

J Vella – Digital Forensics

Files: Carving & Slack Space 34

Reassembly as a Hamiltonian Path Problem



- The Hamiltonian Path Problem example:
Start at node 1, visit all other nodes once and end at 5.
- Shanmugasundaram *et al.* were some of the first to tackle recovery of fragmented files.
 - They formulated the problem of recovery as a Hamiltonian path problem and provided the alpha-beta heuristic from game theory to solve this problem.

J Vella – Digital Forensics

Files: Carving & Slack Space 35

Problem Definition

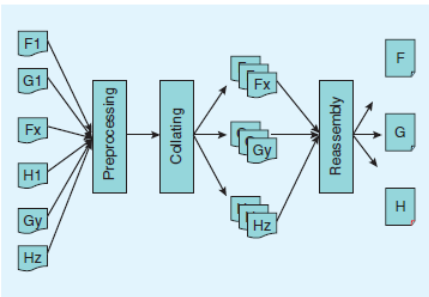
- Given a set of unallocated clusters $b_0, b_1 \dots b_n$ belonging to a document A , one would like to compute a permutation P of the set that represents the original structure of the document.
 - To determine the correct cluster ordering one needs to identify fragment pairs that are adjacent in the original document. This is achieved by assigning candidate weights (W_{b_i, b_j}) between two clusters b_i and b_j that represents the likelihood that cluster b_i follows b_j .
 - The proper sequence P is a path in this graph that traverses all the vertices and maximizes the sum of candidate weights along that path. Finding this path is equivalent to finding a maximum weight Hamiltonian path in a complete graph.

J Vella – Digital Forensics

Files: Carving & Slack Space 36

Carving Architecture

- Design of an application for carving randomly allocated fragment.
 - The application has a set of default computed weights that mimic common fragmentation tactics in filing systems.
 - There are three distinct phases:
 - Pre-processing;
 - Collating;
 - Reassembly.

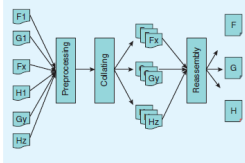


J Vella – Digital Forensics

Files: Carving & Slack Space 37

Carving Architecture - Preprocessing

- Possibly collate sources.
- Apply decryption to fragments if files are encrypted.
- Apply decompression to fragments if these have been compressed.
- At this phase all known (eg allocated) clusters are removed.
 - This is possible if some file system meta data is available / accessible.
 - This saves a good amount of processing time required by the carving system.

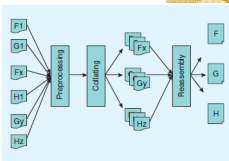


J Vella – Digital Forensics

Files: Carving & Slack Space 38

Carving Architecture - Collation

- An important process is to assign each cluster to one (or more) file-type(s) – called *cluster classification*.
 - It is important even for performance as it reduces the search space.
- Techniques used include:
 - **Keyword / Pattern Matching**
 - Starts from simple magic number sequence to more involved grammar like HTML.
 - There is always a risk this indication is wrong!
 - **ASCII**
 - ASCII characters frequency in a cluster is very high – indicating a document rather than a media file. Not very reliable!
 - **Entropy**
 - Each cluster is counted and entropies are compared. Again not terribly sound.
 - **Fingerprints**
 - Byte Frequency Distribution (BFD), Byte Frequency Cross-Correlation (BFC) for heavily repeated symbols, Header & Footer identification.
 - These distributions are arrived by sampling a good number of similar file types!
 - Accuracy is quoted at 30% and 45%
 - Distributions and Header/Footer patterns increase accuracy to 95%



J Vella – Digital Forensics

Files: Carving & Slack Space 39

Aside: progress in fingerprinting

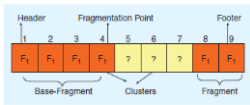
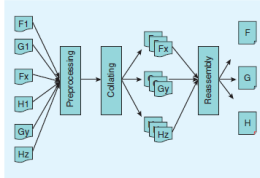
- Fingerprinting is also used to monitor anomalous data in network transmissions (Wang & Stolfo, and Wang et al):
 - Specifically using the BFDs and standard deviation of standard traffic to and then comparing both.
 - The same technique was improved by using BFA on set of models, rather than one average to capture all possible representations in a file type.
 - Research quotes very high success rates –e.g. 100%. For JPEGs it was above 75%.
 - But success rate very much linked to size of file; bigger the higher success.

J Vella – Digital Forensics

Files: Carving & Slack Space 40

Carving Architecture - Reassembly

- Basically it involves finding the fragmentation point of each unrecovered file and then finding the next fragment's starting point.
 - Earlier we stated that Garfinkel's work indicated that most files, if fragmented, are in two pieces.
- Therefore this process needs to find the starting cluster and then uses a technique to identify the ending point of the fragment.
 - If a file has three fragments then there are two fragmentation points.
- An early method studied the structure and sequence of bytes that straddled the boundary of the two clusters.



Carving Architecture – Reassembly (continued)

- Keyword/dictionary
 - If data content is textual then a language dictionary can help stitch two clusters together as a word is most likely formed merging the end and start of two clusters.
 - Similarly with mark-up languages keywords.
 - If two consecutive clusters do not share a keyword then a fragmentation point has been identified.
- File Structure Merging
 - Some files, e.g. JPEG and PNG, have keywords that have their payload quantified (e.g. length).
 - Therefore the length determines where the end of a field should be. If not then there must be a fragmentation point prior to this.
 - Also verification is aided if other than field length a CRC sum is attached to the end of the data.

File structure merging to use Sequential Hypothesis Testing

- The primary idea being that for every cluster added to the path, there is a high likelihood that the subsequent clusters belong to the path as well.
- Pal *et al.* use sequential hypothesis to determine if consecutive clusters should be merged together, not only based on the file type structure but also on the content of individual files.

Reassembly example

Let there be k files to recover, then k file headers ($bh1, bh2, \dots, bhk$) are stored as the starting clusters in the reconstruction paths P_i for each of the k files.

A set $S = (bs1, bs2, \dots, bsk)$ of current clusters is maintained for processing, where bsi is the current cluster for the i th file.

Initially, all the k starting header clusters are stored as the current clusters for each file (i.e. $bsi = bhi$).

The best greedy match for each of the k starting clusters is then found and stored in the set $T = (bt1, bt2, \dots, btk)$ where bti represents the best match for bsi .

From the set T of best matches the cluster with the overall best matching metric is chosen.

Assuming that this best cluster is bti , the following steps are undertaken:

- 1) Add bti to reconstruction path of i th file, (i.e. $P_i = P_i || bti$).
- 2) Replace current cluster in set S for i th file (i.e. $bsi = bti$).
- 3) Sequentially analyze the clusters immediately after bti until fragmentation point bfi is detected or file is built.
- 4) Replace current cluster in set S for i th file (i.e. $bsi = bfi$).
- 5) Evaluate new set T of best matches for S .
- 6) Again find best cluster bti in T .
- 7) Repeat one until all files are built.

The enhancements to PUP are in Step 3 of the above algorithm.

Step 3 will now be described in greater detail.

Reassembly example

In Pal et al.'s sequential fragmentation point detection method, the number of observation weights $W1, W2, W3, \dots$, associated with a set of clusters, are not fixed in advance.

Instead, they are evaluated sequentially and the test is ended in favour of a decision only when the resulting decision statistic is significantly low or high.

Otherwise, if the statistic is in between these two bounds, the test is continued.

Starting with the first data cluster of the base-fragment $b0$, identified during collation, subsequent data clusters $b1, b2, \dots, bn$ are appended to $b0$ and a weight conforming to, $W1, W2, \dots, Wn$ is obtained in sequence with each addition of a cluster.

Accordingly, we define the hypotheses **H0** and **H1** as the following:

H0: clusters $b1, b2, \dots, bn$ -- belong in sequence to the fragment.

H1: clusters $b1, b2, \dots, bn$ -- do not belong in sequence to the fragment.

If the evaluated data clusters $b1, b2, \dots, bx$ do not yield to a conclusive decision, the test continues with the inclusion of cluster $bx+1$ until one of the hypotheses is confirmed.

When hypothesis **H0** is true, the evaluated clusters are merged to the base-fragment and a new test is started. Each time the test starts with a new data cluster in sequence, the weight is computed with respect to the recovered part of the base-fragment.

The test procedure finalizes after one of the following conditions occur:

- 1) **H1** is achieved. (cluster does not belong to the fragment).
- 2) The file is completely recovered.
- 3) An error occurs because no data-cluster

J Vella – Digital ForensicsFiles: Carving & Slack Space45

Reassembly example

Evaluation

- Ultimately, the success of the sequential fragment point detection method depends on two factors.
 - The first factor is the choice of the weight whose design has to take into consideration different file types and to capture semantic or syntactic characteristics of the file.
 - The second factor is the accurate determination of the conditional probability mass functions under the two hypotheses.

J Vella – Digital ForensicsFiles: Carving & Slack Space46

File Craving Tools

- FTK
- EnCase
- Scalpel
- PhotoRec
- Foremost

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

J Vella – Digital ForensicsFiles: Carving & Slack Space47