

Exhibit A

IN THE MATTER OF:
United States of America v. Keith Raniere et. al.
United States District Court, Eastern District of New York
Case No. 1:18-cr-00204-NGG-VMS

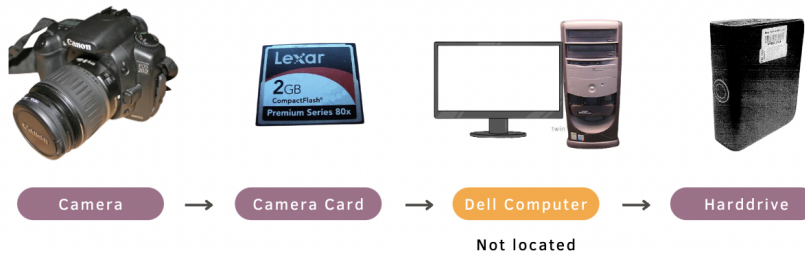
Joint Expert Declaration in Response to FBI Senior Computer Scientist David Loveall II

We, the undersigned, each declare under penalty of perjury, pursuant to 28 U.S.C. 1746, that the following is true and correct:

I. Introduction

1. Government expert David Loveall II challenged select findings of Dr. Kiper. (Loveall Decl., Doc. 1213-3) In this response, we refute his rebuttals and further prove that a camera card (1B15a), which was found in a seized Canon camera (1B15), and a hard drive (1B16) were deliberately and extensively manipulated. Notably, the government used these devices to introduce and support charges of production and possession of alleged child pornography.
2. We have determined that photos were planted and staged across the two devices, with timestamps and folder names manipulated, apparently to simulate a 2005 timeframe. This conclusion is critical because the government depended entirely on these photos' timestamps and folder names being authentic to argue that 22 of the photos on the hard drive were taken in 2005 and therefore illegal, based on the subject's age being fifteen in 2005.
3. This tampering, which involved the manipulation of hundreds of files and timestamps, created a seemingly cohesive chronology and linkage between the devices that perfectly aligns with the government's narrative that the defendant took 22 illegal photos in 2005 with the seized Canon camera (1B15), transferred them to his alleged, missing Dell computer, and later backed them up to the hard drive (1B16).

Diagram 1: The Government's Narrative



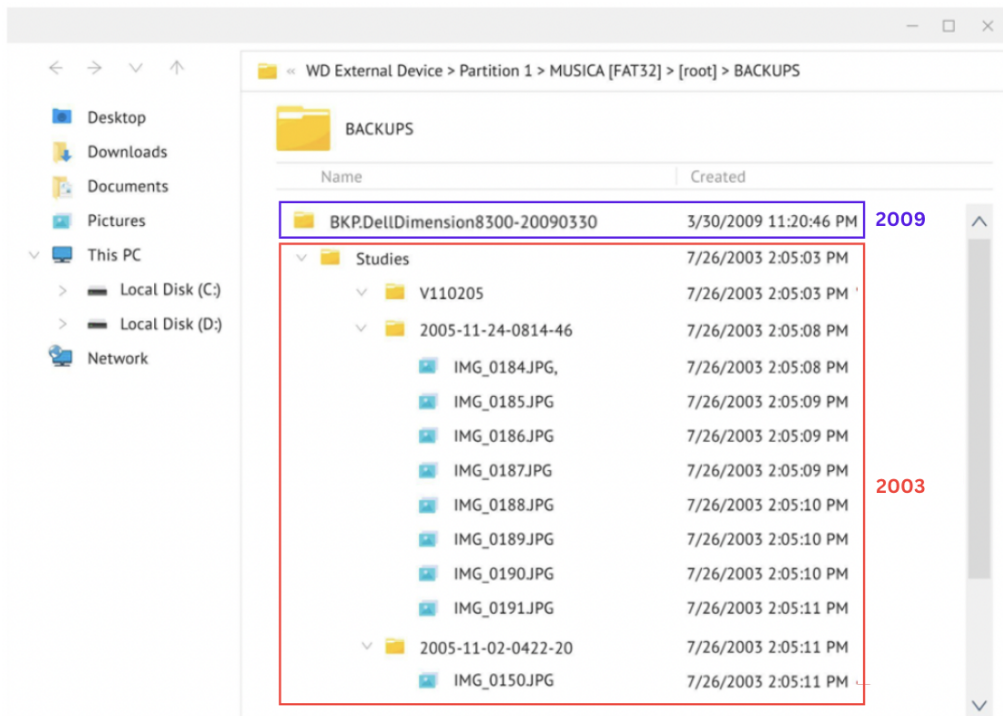
II. Refutation of Loveall's Rebuttals

4. In his report, Loveall made simple but critical errors and deviated from basic forensic standards, which calls into question the reliability of his entire analysis. For instance:
 - a. He asserted that “DellDimension8300-20090330” is the model of a computer. (Loveall Decl., Doc. 1213-3 at ¶ 17). It is not. It is part of a folder name.
 - b. He asserted that a camera (1B15) was “identical” to its camera card (1B15a), but in fact, they are two completely different devices (Id. at ¶ 9).
 - c. He failed to provide any substantiated information from the original evidence, such as MD5 hash values for the forensic copies of the camera card, despite having direct access to those items.
 - d. His rebuttals are based on speculation rather than evidence. In digital forensics, merely providing hypotheses or unsubstantiated assertions is scientifically invalid.
 - e. He selectively responded to the evidence of tampering presented, failing to address key aspects integral to the findings, for example, Findings 2 and 7.
 - f. He failed to address the second part of Dr. Kiper’s report, detailing serious FBI protocol breaches regarding the camera card and camera, and the additional findings from the other undersigned experts.
5. Loveall concurred with one of the seven technical findings, acknowledging that the camera card was altered in FBI custody (Finding 3).
6. Loveall also did not dispute that folder names on the hard drive, some of which were used to corroborate the age of the subject as fifteen, were manipulated (Finding 6).
7. Loveall’s rebuttals were limited to Findings 1, 2, 4, 5, and 7, which we address below, organized by device.

Finding 7: Planting of the Alleged Contraband on the Hard Drive

8. Loveall theorized that all the files in an apparent backup of a Dell computer, including the alleged contraband, have creation dates of 2003 due to a malfunctioning computer battery that reset the computer's clock to 2003. (Loveall Decl. at ¶ 18) But this doesn't explain why the files are dated to 2003, but the backup folder in which they are located has a creation date of March 30, 2009, consistent with the string "20090330" in its name, referring to the same date. If the computer's clock was, in fact, reset to 2003, then when the backup process started, the backup folder created would also have been given a 2003 date by the same clock that assigned a 2003 date to the files being backed up, and not a 2009 date. Loveall's response to Finding 7 was the only instance where he engaged in any type of testing or analysis, but ultimately his theory is scientifically incorrect. We conclude that 168 photos, including the alleged contraband, were planted on the hard drive and disguised as a computer backup. This invalidates the government's theory and evidence of possession — that the photos were authentically on the hard drive.

Diagram 2: Anomalous 2003 File Creation Dates in Alleged 2009 Computer Backup



Finding 4: Timestamp Manipulation on Photos 127-137 on the Hard Drive

9. Loveall theorized that the two-hour shifts in timestamps for Photos 127-137 could have occurred due to “a 2006 Microsoft Windows update that introduced dynamic DST time zones” (Loveall Decl. at ¶ 11). This is incorrect for several reasons: the update was not available in 2005 when the photos were supposedly saved; such an update would only change the computer clock by one hour, not two; and it would not alter the timestamps of files already existing on the computer. Thus, Loveall's explanation is demonstrably false. Our conclusion stands that someone manually changed the photo timestamps, resulting in these two-hour discrepancies. This most plausibly happened because they were attempting to mimic an automatic Daylight Savings Time change from 2005 but made errors in the process.

Finding 5: Timestamp Manipulation on Photo 175 on the Hard Drive

10. Loveall's theory is that someone used the timestamp-editing feature in Photoshop Elements on Photo 175 to change its timestamp. Yet they supposedly ran the tool without selecting a new timestamp, leaving the original timestamp unchanged. (Loveall Decl. at ¶ 15) He offers this to explain why Photo 175's timestamp indicates it was unaltered despite its metadata indicating Photoshop use. This theory is not plausible. It does not make sense to use a tool designed to change timestamps but not change the timestamp. Contrary to Loveall, the far more plausible explanation is that someone used the tool to modify the timestamps of this photo, but did not erase the metadata left behind by using the tool. Furthermore, using such a tool calls into question the authenticity of any timestamp, as we can no longer be sure if it was altered or left as is. Loveall's claim inadvertently demonstrates how easily timestamps can be changed. This contradicts the prosecution's case at trial, based on their expert Booth's false testimony that such timestamps are reliable and hard to change. (Trial T. at 4820:12-20; 4830:3-8)

Finding 2: Additional Photo Files Only Appearing on the Booth Report

11. Loveall only addressed that Booth's report of the camera card contained 37 more files, which appeared to be recovered photos taken by the seized camera. He overlooked the multiple anomalies regarding those files, which support the likelihood that some of these

additional files, which have no metadata or visual content, were staged to mimic originals of photos on the hard drive. (Kiper Report at Bates 005-006) Therefore, Loveall neglected a bulk of the finding, rendering his rebuttal incomplete and misleading, as it fails to address the substantive concerns about the authenticity of the files in question.

12. Furthermore, he inaccurately claimed that the camera and the camera card were “identical.” (Loveall Decl. at ¶ 9) Perhaps he meant to refer to the two copies of the camera card. Regardless, he did not provide the hash values¹, the digital fingerprints required to prove such claims scientifically. (Trial T. at 4784:2-22) Using hash values to prove that content has not been altered is such a staple in digital forensics, and is trivial to do, that it raises questions about why Loveall, a purported forensic examiner, would choose to leave them out of his response. Setting aside his confusion regarding the camera and camera card, we are left to speculate whether Loveall really did verify the two camera card copies were identical, and if so, what verification process he used. To claim two pieces of digital information are identical without providing hash values is unheard of in our field. If the hash values are different, that would indicate additional tampering occurred in FBI custody.
13. Finally, New Finding 8 strongly indicates that some of 37 files were planted and staged with edited timestamps, including Photos 90-98, which on the hard drive depict a government witness identified as ‘Daniela.’ (“The Daniela Range”) She testified to being photographed with a Canon camera as an adult by the defendant in 2005, the same year of these photos’ timestamps. (Trial T. at 2422:7 - 2424:4) Therefore, the “Daniela Range” appeared to provide a critical link from the seized camera and the photos to the defendant. However, as analyzed in Appendix A, this range was planted and staged.

Finding 1: Manipulation of Purported Originals of ‘Daniela’ on the Camera Card

14. Loveall claimed that portions of Photos 180-183 being incorrectly associated with Photos 93, 94, 96, and 97 (part of the ‘Daniela’ range) in the FBI forensic report could be due to an error from a forensic tool rather than manipulation. (Loveall Decl. at ¶ 6, 7) His theory is speculative and incorrect. He didn’t provide any evidence that the tool used,

¹ The problem of Loveall's missing hash values is compounded by the fact that we cannot independently verify his claim, as the government has not granted us access to these copies.


ForensicToolkit – which is rigorously tested by the FBI – could make such an error (Id.) His hypothesis relies on a method of file overwriting that the camera doesn't use. (See Appendix B) He also suggested this error is more likely when the card is full, but it appears this card was at most 6% fully, using at most 120 MB of this 2 GB camera card, based on the furthest located files being Photos 21-41, which have not been overwritten. (Id.; Government Exhibit 521 A - Replacement) Regardless, new Finding 8 strongly indicates that the purported originals of 'Daniela' range on the camera were planted, manipulated, and staged.

III. New Findings

15. In responding to Loveall, we uncovered more evidence of tampering on the camera card and concluded that:
- a. Someone used a computer to plant Photos 81-100 (which includes the 'Daniela' range), and subsequently edited their creation dates. (Finding 8)
 - b. Someone used a computer to plant Photos 224-243 on the camera card, and subsequently edited their creation dates. (Finding 9)
 - c. Someone used a computer to manually manipulate the last accessed dates of Photos 21-42. (Finding 10)
 - d. Someone used a computer to manually manipulate the last accessed dates of Photos 193-99. (Finding 11)


IV. Conclusion

16. We have refuted Loveall's rebuttals and further demonstrated the camera card and the hard drive were extensively tampered with. Hundreds of files were planted, staged, and manipulated across both devices. Given admitted government misconduct, including violating evidence protocols, providing evidence to unidentified and unauthorized personnel, and altering the original camera card, the involvement of government personnel in this evidentiary fraud is inescapable – an unprecedented finding in our combined 150+ years of forensic experience.

DocuSigned by:

9B8F4F5049A24DD...
Signature: _____
Date: 11/20/2023 _____

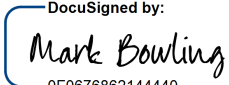
Name: Dr. James Richard Kiper, Ph.D.

Background: Former FBI Special Agent, Computer Forensic Examiner, and Unit Chief at the FBI Academy, 20 years' service to the FBI

DocuSigned by:

7E268C7A16C94A2...
Signature: _____
Date: 11/20/2023 _____

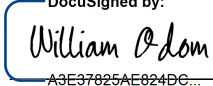
Name: Stacy Eldridge

Background: Former FBI Senior Forensic Examiner, 10 years' service to the FBI

DocuSigned by:

0F0676862144440...
Signature: _____
Date: 11/20/2023 _____

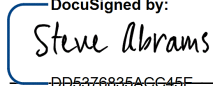
Name: Mark Daniel Bowling

Background: Former FBI Forensic Examiner, Cyber Program Manager, FBI Inspector in Place, and FBI Assistant Special Agent in Charge, 20 years' service to the FBI

DocuSigned by:

A3E37825AE024DC...
Signature: _____
Date: 11/20/2023 _____

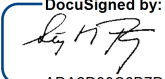
Name: William Odom

Background: Former FBI Special Agent and Forensic Examiner, Manager of FBI Forensics Lab in Houston, 5 years' service to the FBI, 25+ years' experience in the field.

DocuSigned by:

DD5376835AGG45E...
Signature: _____
Date: 11/20/2023 _____

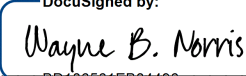
Name: Steven Abrams, J.D., M.S.

Background: 25+ years in digital forensics, worked 1,500+ cases, served 11 years as a South Carolina State Constable, for the US Secret Service.

Signature: 
Date: 11/20/2023

Name: Stephen Bunting

Background: Former Captain of the University of Delaware Police, created the University of Delaware Police's digital forensics unit; trained hundreds of examiners and authored five textbooks in the field.

Signature: 
Date: 11/20/2023

Name: Wayne Norris

Background: 60+ years of software development experience across 35 operating systems, the government's lead software development expert witness in the landmark *Microsoft vs. Commissioner of Internal Revenue* case, 36 years of computer forensic expert witness experience.

Appendix A

Appendix A: New Findings of Planting and Staging on the Card

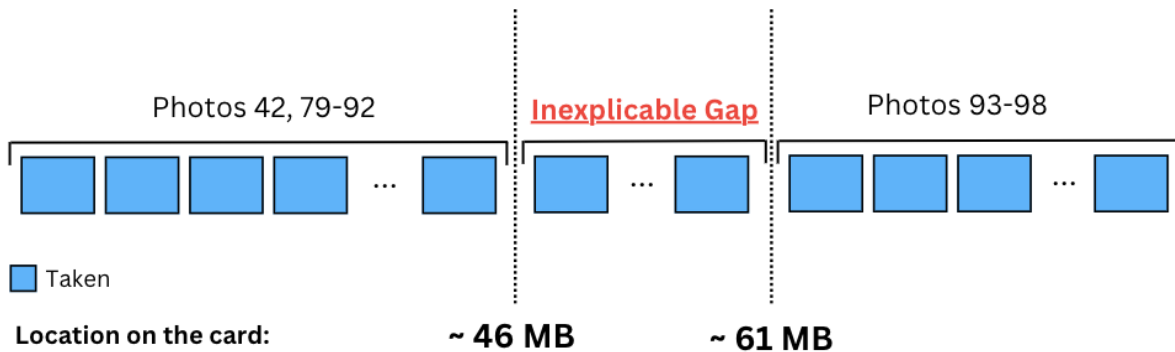
Finding 8: Planting and Staging of Photos 81-100 on the Card

1. We have determined that photos 81-100, **were staged and planted on the card**. These photos include the “Daniela range,” which as discussed in our joint statement at point 13, appeared to link the photos on the hard drive to the seized Canon camera and linked the photos and the seized camera to the defendant.
2. The calculations in our analysis are based upon tested and verified Canon EOS 20D camera behavior, as well as the file sizes and byte offset locations of carved photo data provided in the government’s forensic reports and file listings (See Appendix B)
3. Our conclusion is based on the finding that Photos 93-97 could not occupy their current locations on the card if they were authentically saved to the card by the camera, as discussed below.
4. When the camera starts taking photos, it looks for the earliest available physical space to save them on the card. Photos 90 to 98 were taken one after the other, so when Photo 90 was taken, the space on the card before it was already full. This leads us to the question we’ll examine below: *What was in the space before Photo 90, and was it voluminous enough to justify the current physical location of Photo 93?*
5. To answer this, we examine how this type of camera, a Canon EOS 20D with firmware 2.0.2, overwrites the directory entries of deleted photos. It only overwrites the filenames, more specifically directory entries, of deleted photos with new ones if they're taken **within the same set of 100**. Once a photos’ filename (directory entry) is overwritten, the previous photo’s filename is not listed on the card anymore. For example:
 - a. The photographer takes photos 500 through 550 and later deletes photos 520 – 530.
 - b. The photographer then takes photos 551 – 570.
 - c. The file directory entries for Photos 551 – 561 will overwrite the file directory entries for photos 520 – 530.
 - d. When a forensic tool is run on the card, it will not find directory entries for photos 520-530, as they no longer exist.
6. Going back to our question of what was in the space before Photo 90, this camera’s overwriting behavior tells us that the photos originally occupying the space before Photo

90 must either (a) still be listed by name in a directory entry or (b) they've been deleted and their filenames (directory entries) were overwritten by others before Photo 101 was taken. The photos in category (b) could have only been overwritten after the set of photos 90-98 were taken. Since there were only two post-98 photos left in this set of 100 – photos 99 and 100 – this means, at most, two photos were taken and saved before the physical location of Photo 90. If there were more than two, we would see more deleted directory entries listed because 99 and 100 can collectively only overwrite two filenames.

7. Checking the card, we see that Photos 21-41, 42, and 81-92 are the only pre-93 photos still listed. However, Photos 21-41 are physically located after Photo 93, so they cannot be the ones that were in the space before 93. That leaves us with Photos 42, 81-92, and at most, two unknown not-yet-overwritten photos as the possible occupants of the physical space before Photo 93. Photos 79 and 80 were deleted after Photo 98 was taken but before Photo 99 was taken. This means the two not-yet-overwritten photos were 79 and 80.
8. Here's the crux—Photo 93 is positioned ~61 megabytes (MB) into the card, but Photos 79-92 occupy only ~35 MB, leaving a 26 MB gap. (See Diagram 1 below) Given the maximum possible photo size for this camera of ~12.3 MB, it would be impossible for Photo 42 to fill that gap such that the camera would write Photos 93 through 97 to their current locations on the camera card. Therefore, Photo files 93 through 97 were not genuinely saved by the camera to their current positions on the card; instead, someone copied them onto the card from somewhere else, using a computer.

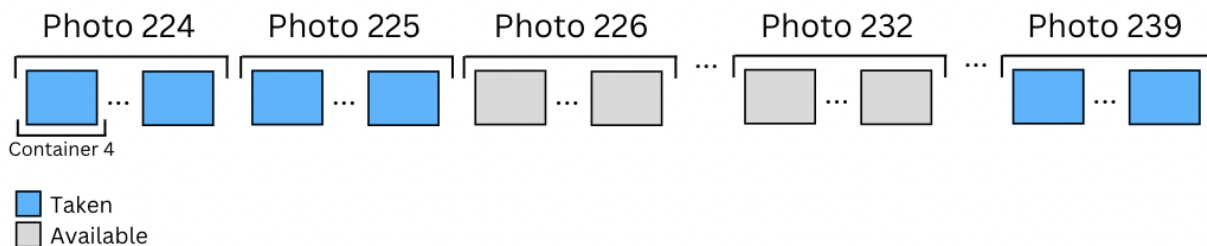
Diagram 1: Illustration of Inexplicable Gap in Photo Locations



9. Furthermore, copying the files to the camera card would update the file creation timestamps, resulting in a disparity between the file creation timestamp stored in the FAT and the creation timestamp stored in their EXIF metadata. The fact that these timestamps match for these files indicates that someone has intentionally edited them.
10. To recapitulate this finding:
 - a. Someone used a computer to add photo files 81-100 to the camera card.
 - b. They edited the timestamps of each of these twenty files, in such a way that it perfectly aligned with the timestamp sequence on the hard drive.
11. These deliberate actions constitute planting and staging and call into question the authenticity of these photos' metadata, including their timing and the origin of the photos.
12. Considering that (a) the photos on the hard drive, including 81-100 and the alleged contraband, all have metadata linked to the seized Canon camera, giving the impression they were taken by that camera, (b) that FBI Special Agent Christopher Mills testified that he found this card inside that seized camera (Trial T. at 4305:14-24), and (c) that photos were planted and staged in a manner that perfectly aligned with the hard drive, the discovery of these manipulations directly erodes the authenticity of the main evidence presented by the government.
13. This discovery demonstrates deliberate planting, manipulation, and staging of these photos on the card.

Finding 9: Planting and Staging of Photos 224-243 on the Card

14. The range of Photos 224-243, the final sequence on the card, though not the alleged contraband, seamlessly continues from the last photo, Photo 223, on the hard drive. This continuity supports the government's narrative that the photos on the hard drive originated from the camera and card. However, we determined that this final range was planted and staged.
15. This conclusion is drawn primarily from the observation that Photo 242, the only photo in this range whose location we were explicitly given, is not located where it should be on the card if authentically saved to the card there by the camera. To understand why, let's describe how this camera's firmware determines where to store photos on the card.
16. The card consists of a series of data containers, technically called clusters, which we'll call "containers" for simplicity. The earliest container that the camera starts adding photos to is container 4. In each session, meaning the time from when the camera is turned on to off, it looks for the earliest container not occupied by an active (not deleted) file. This becomes its starting point. All subsequent photos taken in the session are added forward from there. If one of those photos is deleted, creating a gap of available containers, this gap is only filled in the next session (when the camera is next powered off/on).
17. Photo 242 is part of the range 224-243. When Photo 224 was taken, there were no active (not deleted) photos remaining at the beginning of the card, and since Photos 224 - 243 were the last photos taken, they must be at the beginning of the card, starting at container 4.
18. In this range, the first session consists of Photos 224-239, but the directory entries for Photos 226 and 232 are missing, indicating they were deleted, leaving gaps that were overwritten by later photos.



19. The soonest that either Photos 226 or 232 could have be overwritten is after this session, by Photo 240. If Photo 240 occupied the gap left by 226 and/or 232, when Photo 240 was deleted, that space would be available again. Photo 240 must have been deleted before Photos 241-243 were taken in order for it to have been overwritten by one or more of Photos 241-243 as they were the last taken. Thus, when Photo 241 was taken, the earliest physical location available is where Photo 226 began. Then, Photo 242 would start from where Photo 241 ended. Depending on the sizes of Photos 226 and 232, Photo 242 would be placed somewhere in the gaps left by Photos 226 and / or 232.
20. Therefore, we calculate that the space from container 4 to where Photo 242 started was occupied by Photos 224 through 231 (sans 226) and 241, totaling 16,744,448 bytes. Given each container holds 32,768 bytes, this would take 511 containers, positioning Photo 242 at container 514, not at container 977, as is currently the case.
21. This misplacement proves that Photo 242, along with the entire series from 224-243, could not have originated from the camera. Based on the same reasoning as points 9 and 10 above,
 - a. Someone used a computer to add photos 224-243 to the camera card.
 - b. They edited the timestamps of each of these twenty files, in such a way that it perfectly extended from the timestamp sequence on the hard drive.
22. Since Photos 224-241 collectively take up 968 containers, beginning at container 4, and Photo 242 begins at cluster 977, this provides convincing evidence that whoever added these photos to the camera card did not take into account this camera's method of placing the photos.
23. This finding demonstrates that these photos were planted, manipulated, and staged on the card.

Finding 10: Tampering of Last Accessed Dates of Photos 21-42 on the Card

24. The earliest range of recovered photos on the card, 21-42, though not the alleged contraband, is significant because it seamlessly precedes the first range of photos on the hard drive (43-58), suggesting those photos came from this camera and card. However, this range contains an anomaly that reveals tampering of their last accessed dates, as discussed below.

25. The last accessed dates for 21-42, which note when each photo was last opened, are inconsistent. All but photos 29 and 42 have a last accessed date of October 16, 2005, two days after their alleged creation. However, photos 29 and 42 show a last accessed date of two days earlier, October 14, 2005. See the table below.

Table 1: Last Accessed Dates of Photos 21-42

Photo	Last Accessed Date
21	10/16/05
...	...
28	10/16/05
29	10/14/05
30	10/16/05
...	...
41	10/16/05
42	10/14/05

26. Thus, the last accessed date being changed to October 16, 2005, must have been triggered by an access event from a computer, more specifically, while the card was in a card reader or slot. This is because this particular camera model does not update the last accessed date when photos are viewed or accessed on the camera itself, nor does the last accessed date get updated when the pictures on the card are attached to a computer via USB connection directly to the camera, as the card is not mounted as a drive, but rather the camera is attached as a “portable device”. Thus, a program like Photoshop Elements cannot modify the pictures while the computer hosting it is connected to the camera; the program can only delete them after uploading them. If the user opts not to delete them from the card after they have been uploaded to a computer, then the last accessed date listed on the card is still not updated.

27. The contradiction in the data comes from the fact that this computer access event would have undoubtedly affected all of the photos in this range. They were all stored in the same folder, they were all active (i.e. not deleted) at that time, they were not deleted until after

Photo 101, purportedly taken on October 20, 2005, they are still listed on the camera card (See point #4 above), and stored in the same folder. tampering.

28. The distinct differences in the last accessed dates for photos 29 and 42, deviating from the rest, provide evidence of tampering. These discrepancies cannot be attributed to regular camera use or routine Windows activities.

Finding 11: Tampering of Last Accessed Dates of Photos 193 - 199 on the Card

29. Similar to Finding 10 the last accessed dates for Photos 193 - 199 are inconsistent. Photos 194 and 197 - 199 have a last accessed date of December 21, 2005, two days after their alleged creation. However, Photos 193, 195, and 196 still show a last accessed date of two days earlier, December 19, 2005; the alleged creation date. See the table below.

Table 2: Last Accessed Dates of Photos 193 - 199

Photo	Last Accessed Date
193	12/19/05
194	12/21/05
195	12/19/05
196	12/19/05
197	12/21/05
198	12/21/05
199	12/21/05

30. Thus, the last accessed date being changed to December 21, 2005, must have been triggered by an access event from a computer, more specifically, while the card was in a card reader or slot. This is because this particular camera model does not update the last accessed date when photos are viewed or accessed on the camera itself, nor does the last accessed date get updated when the pictures on the card are attached to a computer via USB connection directly to the camera, as the card is not mounted as a drive, but rather the camera is attached as a “portable device”. Thus, a program like Photoshop Elements cannot modify the pictures while the computer hosting it is connected to the camera; the

program can only delete them after uploading them. If the user opts not to delete them from the card after they have been uploaded to a computer, then the last accessed date listed on the card is still not updated.

31. The contradiction in the data comes from the fact that this computer access event would have undoubtedly affected all of the photos in this range. They were all stored in the same folder, they were all active (i.e. not deleted) at that time, they were not deleted until after Photo 201 was taken on or after December 26, 2005, they are still listed on the camera.
32. The distinct differences in the last accessed dates Photos 194 and 197 - 199, deviating from the rest, provide evidence of tampering. These discrepancies cannot be attributed to regular camera use or routine Windows activities.

Appendix B

FAT16 File Creation / Deletion / Overwriting in the Canon 20D Camera

By Stephen Bunting – Bunting Digital Forensic, LLC

18 November 2023

Overview

FAT16 is an organizational framework for storing files and directories (folders) on various storage devices, ranging from hard drives to floppy disks, USB drives, and media cards. It was created in 1984, and the original FAT16 file system can store a maximum of 2 gigabytes. Thus, it's rarely used today, as the size of most media today far exceeds that limit.

It derives its name from its predominant feature, the File Allocation Table (FAT), and the 16 its name from the length of each cluster entry in the FAT, which is 16 bits or 2 bytes.

The FAT16 file can be used by any operating system (OS). This paper focuses on the use of the FAT16 file system in the Canon 20D camera (Firmware Version 2.0.2). While the structure of the FAT16 file system doesn't vary between operating systems, how those operating system actually makes use of the directory entries and to which clusters they write data does vary.

Generally, how the Canon 20D operating system writes to FAT16 prioritizes speed, as burst rate (how many pictures per second the camera can record) is a highly sought-after feature by photographers. Windows, by contrast, seems to write to FAT16 in a manner that facilitates data recovery. The differences between how the Canon 20D and Windows write to FAT16 will also be reviewed.

Directory Entries

A directory entry in FAT16 lists files and directories for a parent folders. Each directory entry is 32 bytes long. Each line begins with the file or folder name followed by its file extension. For example, in IMG_0001.JPG, "IMG_0001" is the file name and "JPG" is the file extension. Among other information, the directory entry contains two additional very important pieces of information: the starting cluster and the file size in bytes. This is how the operating system knows where to find the first cluster of data and how much data is required, in bytes, for the file for purposes of creation, modification, or deletion.

File Allocation Table (FAT)

FAT16 uses two file allocation tables, FAT1 and FAT2, and they are identical, as the second copy is a data protection feature. It will be referred to as the FAT. The FAT in FAT16 contains 2-byte arrays of numbers for each cluster in the data storage area, which is organized in Clusters. If the value for a cluster is zero, the cluster is available to store data. If not zero, it will contain a number. The number is the next cluster in the chain of clusters that make up the file. When the last cluster is used for a file, a special number is used to denote the cluster as EOF or end of file. Often referred to in forensics as EOF. Should a cluster be found to be bad, another special number will be used to mark it as bad and it will not be used.

In an ideal world, all clusters for a file would be contiguous with one cluster immediately following the other for the same file. However, fragmentation does occur. For example, suppose that the cluster run for a file is 2,3,8,9,10. What has happened is that another file occupies clusters 4,5,6,7. When this condition occurs, it is called file fragmentation. Its impact on file deletion and recovery will be discussed later.

The directory entry, among other things, tells us the file's name and extension, at which cluster the data for a file starts, and how large the file is in bytes. The FAT tracks cluster space utilization (which clusters are available or not), the next cluster in cluster runs (the clusters that make up a file in the order they used by the file), and the end of the file, once the starting cluster is known.

CLUSTERS

Clusters are where a file or folder's data is actually stored. The directory entry and the FAT are pointers to this data storage area. Clusters have a fixed sized that is determined when media is formatted. In this example, and on the 2GB CF Card in the Raniere matter, the cluster size is 32,768 bytes. The first cluster available for file and folder data is cluster 2.

Figure 1, below shows a file written to a FAT16 file system. The directory entry shows a file named IMG_0001.JPG with a starting cluster of 4 and a file size of 130,233 bytes. Looking at the entry for cluster four, the starting cluster, in the FAT we see that the number in cluster four, is 5, meaning 5 is the next cluster in the run. Then in cluster five, we see the number there is 6, the next cluster. Looking at cluster six, we see a 7, the next cluster. Finally, if we go to the entry for cluster 7, we see it contains the special number for the EOF. Dividing the size of the file by the size of a cluster, it is not an even number, meaning that there is some space left at the end of the cluster between the end of the file and the end of the final cluster. We call this slack space. The purple in the cluster space depicts the actual file or picture data; note cluster 7 is not completely full.

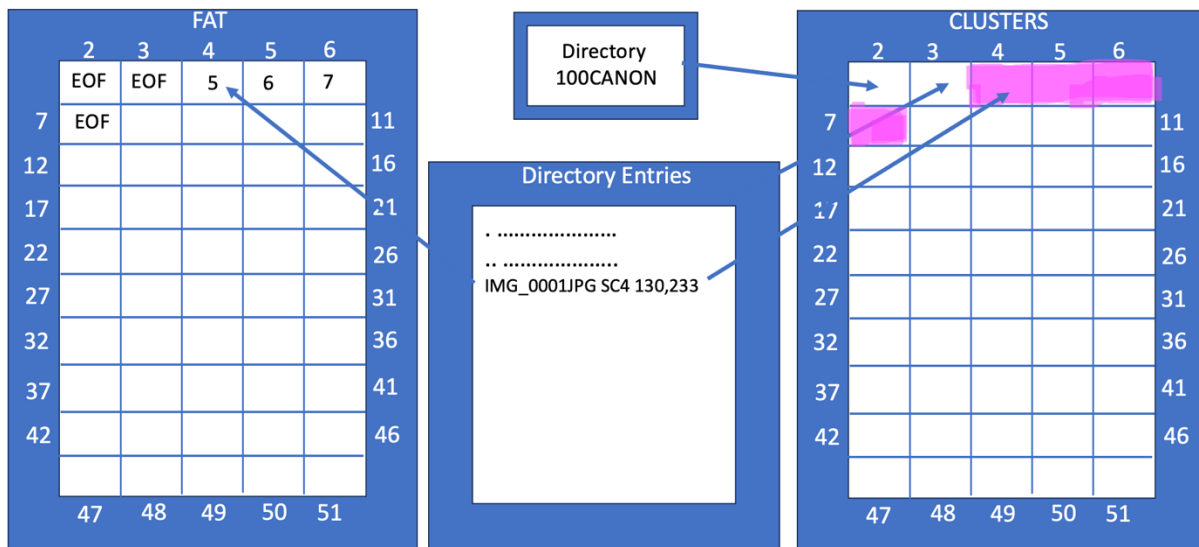


Figure 1 – Scenario one: Shows one file written to a FAT16 file system

Scenario Two: A Second Photo is Added to CF Card

Figure 2, below, depicts a second photo being taken and added to the FAT16 file system. The directory entry shows the file name is IMG_0002.JPG, that it starts at Cluster 8, and its size is 130,233. If we go to our FAT and look at the entry for the starts at cluster, 8, and the cluster run is 8,9,10,11. The green area in the cluster space depicts where the data for this file is stored. Currently, there is no fragmentation.

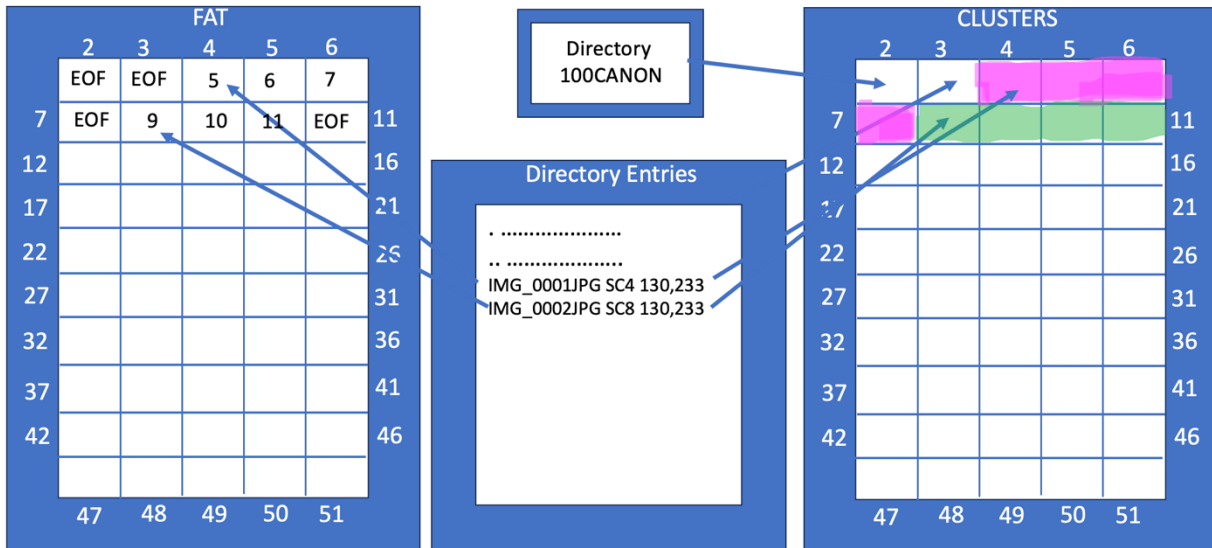


Figure 2 – Scenario Two: Shows adding a second file to our FAT16 file system

Scenario Three: Deleting a File using the Camera Trash Button during current power-cycle

Figure 3, below, shows what happens when a file is deleted using the in-camera trash button, while the camera has not been powered off yet. It's important to note that power cycling the camera dictates certain behaviors. When a file is deleted, two things occur.

1. The first character of the directory entry is changed to the hexadecimal notation of "E5" (0xE5), as seen below.
2. The FAT entries that make up the cluster run for the file are changed to zero, meaning the clusters are available for storing other data, BUT the data itself in the clusters is not touched. At this stage, the file is very much recoverable.

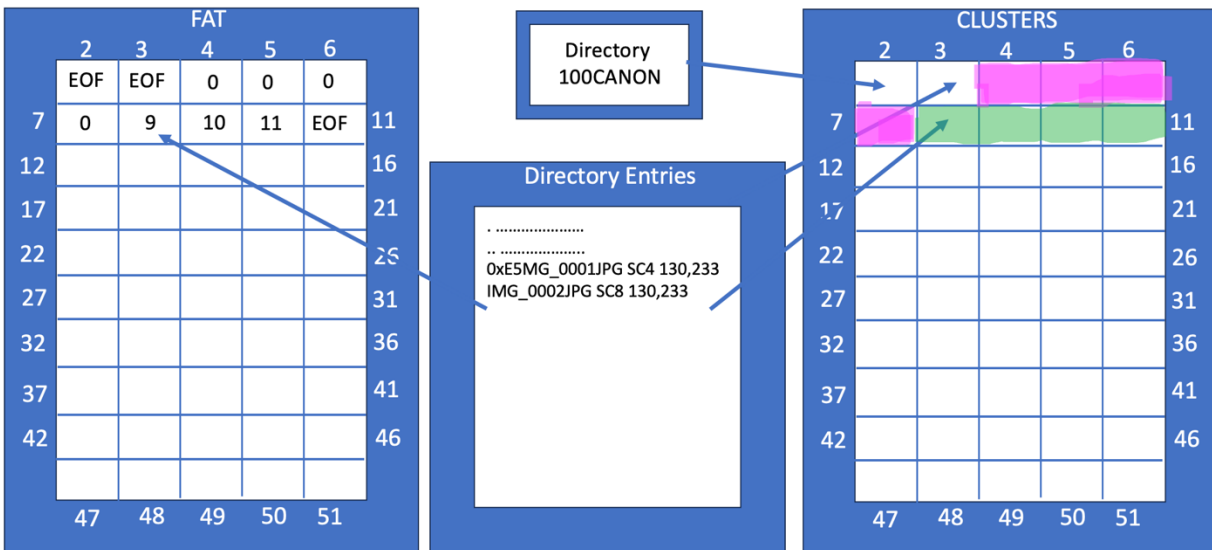


Figure 3 – Scenario Three: Shows what happens when a file is deleted

Scenario Four: Another Picture is Taken by the Camera During Same Power-Cycle

Figure 4, below, shows what happens when another picture is taken after the above file was deleted and within the same power-cycle, i.e. the camera has not been turned off and turned on again. Canon will overwrite any directory entries marked for deletion from the top down.

Canon will overwrite directory entries that are marked for deletion any time a new file is created until the active folder changes, i.e. file IMG_0101.JPG is taken, and folder 101CANON is created to hold the next set of 100 images. After the next folder in sequence is created, any file deleted in-camera by the Canon OS will be preserved in the inactive folder.

Canon writes in sequence default, i.e. IMG_0001.JPG to IMG_9999.JPG, and does not repeat a file number. It stores files in folders in increments of 100. The folders named sequentially, i.e. files 1-100 go into folder 100CANON, 101-200 go into 101CANON, 201-300 go into 102CANON, and so forth. While the Canon OS is writing new files to a folder, that is the active folder. Any preceding folders are no longer active and will not be written to again, therefore, deleted directory entries in those inactive folders are preserved.

Even though cluster 4 is the first available file (cluster marked zero), the Canon 20D, within the same power-cycle, will not write to cluster 4. Instead, the Canon 20D will look forward from the last cluster it wrote to for the next available cluster to write to, which in this case is cluster 12. The FAT shows the cluster run for file IMG_0003.JPG is clusters 12, 13, 14, 15. The area in the cluster space that depicts where IMG_0003.JPG will be written to is shown in aqua. The directory entry for IMG_0001.JPGs has been completely overwritten, but its data remains untouched and contiguous. It can be fully recovered by a technique known as carving.

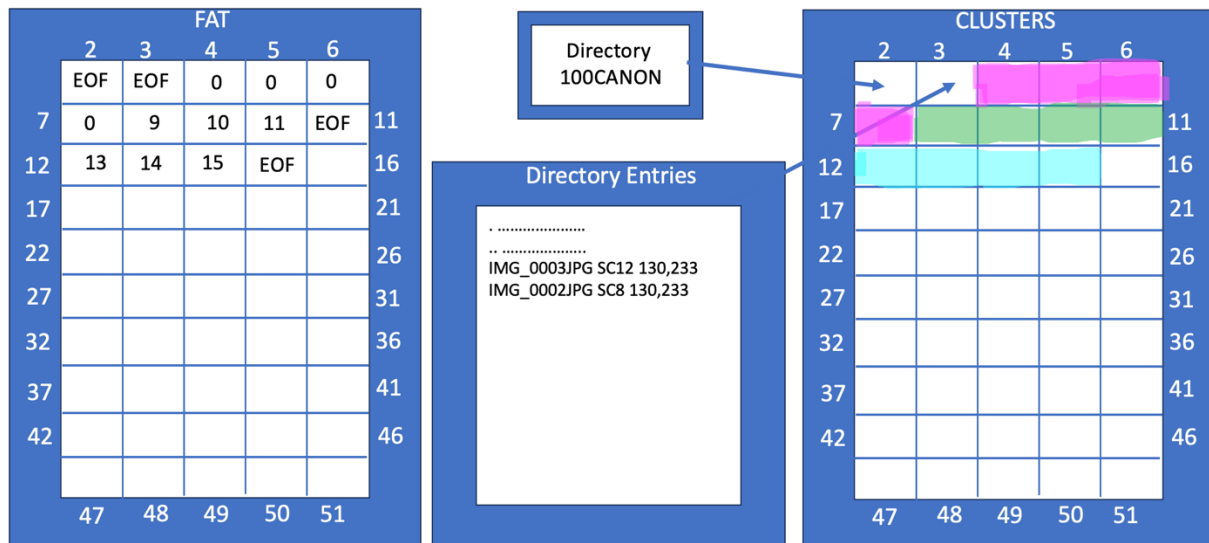


Figure 4 -Scenario Four: Shows a new file overwriting a directory entry in the same power-cycle

Scenario Five: Another Picture is Taken by the Camera after powered off and back on

Figure 5, below, shows a similar situation to the one described in Scenario Four; however, before taking IMG_0003.JPG, the camera is turned off and then powered on. When powered on, the cluster availability is refreshed as the first picture is taken, and Cluster 4 is now seen as the first available cluster. Just as before, the directory entry from the deleted file is overwritten as the folder 100CANON is still the active folder receiving new media files.

The FAT shows the cluster runs for this file start at cluster 4, and include 4,5,6,7, which differs from the previous scenario where the picture was taken before a power-cycle of the camera. This difference has caused all traces of file IMG_0001.JPG to be overwritten; there is no directory entry and no data is left in the clusters.

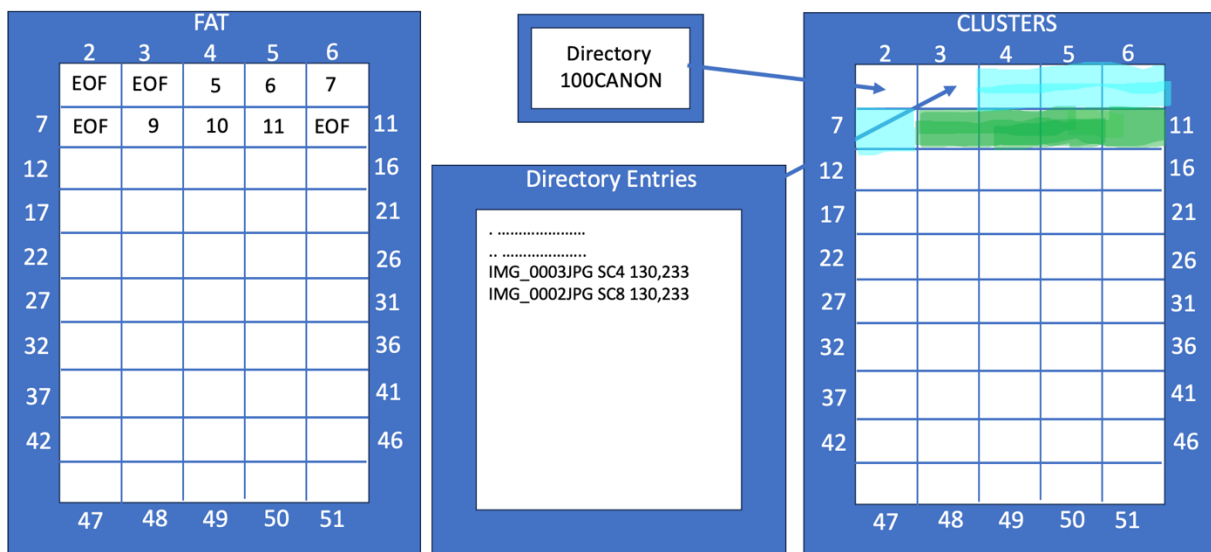


Figure 5 -Scenario Five: Shows IMG_0003.JPG taken after power-cycling the camera

Scenario Six: The Canon OS vs the Windows OS

Figure 6, below, shows the difference between the Canon OS and Windows. Instead of taking a picture to create IMG-0003.JPG, a program is used in Windows to create a file of the same exact size 130,233 bytes, and Windows is used to place the file on the CF Card. To achieve this, the CF Card is removed from the camera and placed it in a card reader attached to a Windows computer. In Windows Explorer drag and drop the Windows created file, whatever its contents, and place it in the folder 100CANON on the CF Card.

Windows will look at the clusters written on the CF Card and choose the starting cluster immediately following the last cluster in use at the time, which is Cluster 12; it does not matter that clusters 4-7 are available and able to fully contain the file without fragmentation. The cluster run for this file is 12, 13, 14, 15. Further, Windows will append the directory entry to the end of the current directory list. It will NOT overwrite any directory entries that have been marked for deletion.

The Canon OS and Windows OS behave quite differently. The Canon OS will overwrite directory entries marked for deletion from the top down and continue to do so until the current active directory becomes inactive. Once a new folder becomes active no more images can be written to the inactive folder, and deleted directory entries therein will be 'preserved'. Windows, quite by contrast, will not overwrite any directory entries.

Canon will use the first available cluster at startup, and continue writing forward from that point, even though cluster space behind that point (i.e. closer to the beginning of the card) may become available due to file deletions, *during that power cycle*. Only when Canon has reached the last cluster of the CF Card will it go back to the beginning of the CF Card to seek any available space. In the Ranieri matter, there is no indication that the card was ever filled as the highest file number written does not support having filled the 2GB CF Card in this case.

By contrast, Windows seeks the first available cluster immediately following the last cluster written to. Windows avoids writing fragmented files, whereas Canon writes to the next available cluster without regard to fragmentation.

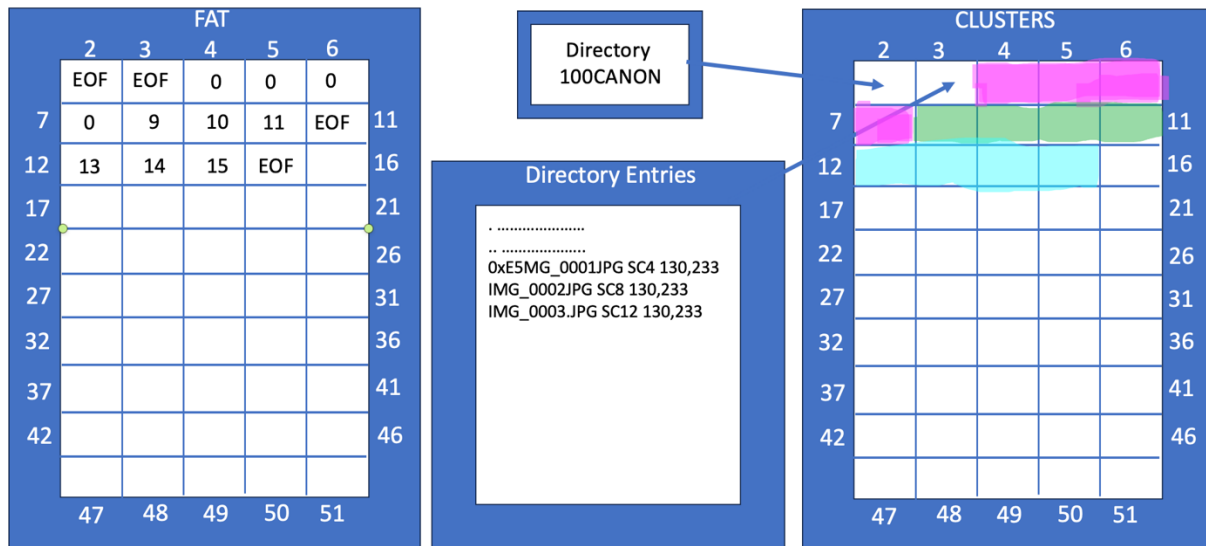


Figure 6 -Scenario Six: Showing file placed on CF Card by Windows

Scenario Seven: Writing Fragmented Files

Thus far, contiguous files of the same size have been used to keep things simple and to be able to make comparisons between different scenarios. Figure 7, below, shows a fragmented file.

Rarely are JPG images the same size, as picture content varies greatly. In Scenario Five IMG_0003.JPG is one byte larger than would fit in 4 clusters. 4 clusters can hold 131,072 bytes, and one more byte forces the file to use one more cluster, even though it's just one byte. Thus, the cluster runs for this file are now 4,5,6,7,12. They are no longer contiguous, which is known as a fragmented file. The one byte of data in cluster 12 is depicted by the thin sliver of aqua. Regardless of how much space remains, no file can use it, but there could be data from another file in this slack space.

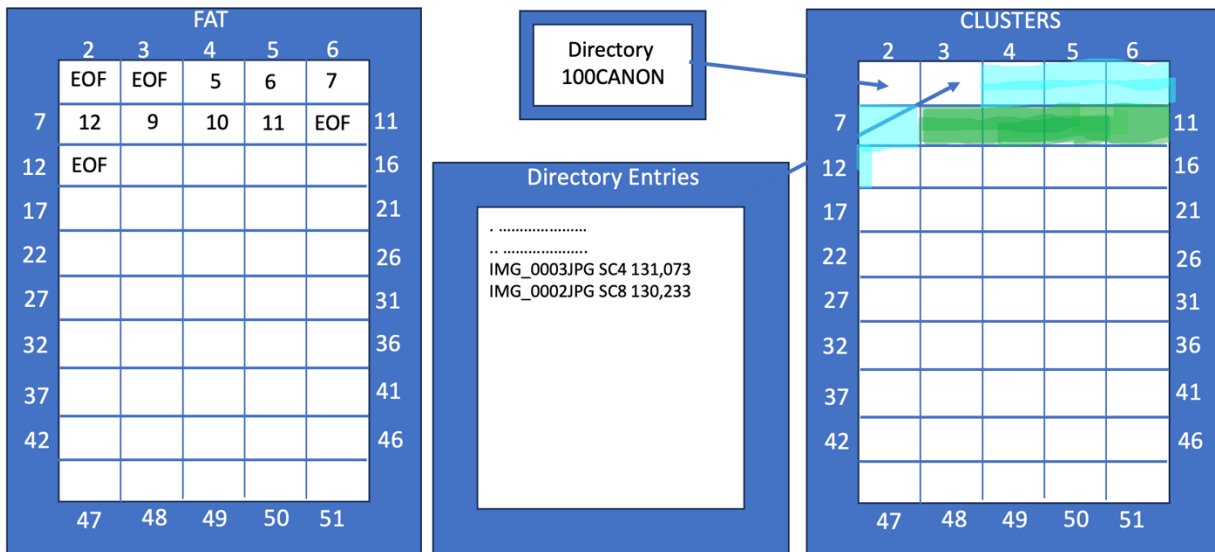


Figure 7 – Scenario Seven: Shows file fragmentation

Scenario Eight: Deleting Fragmented Files

Figure 8, below, shows what happens when fragmented files are deleted. In this case, the camera remains in the same power cycle as Scenario Seven. At this point the camera will use cluster 13 as the starting cluster as pictures have already been taken during this power cycle. Using the camera trash button, file IMG_0002.JPG is deleted. Its directory entry will be marked as deleted, and its clusters will be marked as available. Next, picture IMG_0004.JPG (130,233 bytes) will be taken, and it will occupy clusters 13, 14, 15, and 16 (shown in orange). Keep in mind the Canon OS only seeks available clusters in a forward direction, not backward, during a power cycle, continuing until it reaches the end of the cluster space. The directory entry for IMG_0004.JPG will overwrite the directory entry for IMG_0002.JPG, as it's marked for deletion. Next, still in the same power cycle, we will use the trash button to delete IMG_0003.JPG. Its directory entry will be marked as deleted and its cluster space will be marked as available.

Figure 8, below, shows the final result of the above activity. If attempting to recover data from this CF Card the fragmentation and lack of cluster run data makes file recovery inaccurate. In this case, the deleted directory entry for file 0xE5MG_0003.JPG, shows the starting cluster and file size. While this helps, the cluster runs are unknown. The assumption is to go to the starting cluster, which is Cluster 4. The file size is 131,073 bytes and requires 5 clusters.

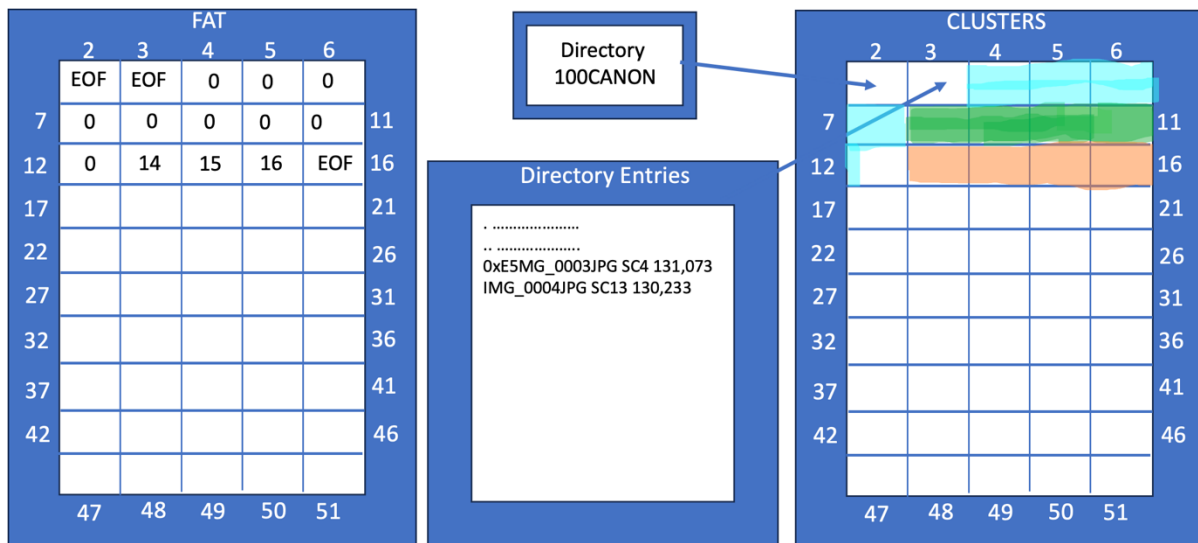


Figure 8 – Scenario Eight: Shows fragmented files when deleted

Scenario Nine: Recovering deleted, fragmented files

Figure 9, below, shows the problem when trying to recover a fragmented, deleted file. The starting cluster 4 is correct, as and that information was contained in the deleted directory entry. Starting at cluster 4, the data is recovered from the next 4 available clusters, 5-8. Recall that cluster 8 was originally the starting cluster for file IMG_0002.JPG, but its directory entry is gone, and its clusters are marked as available. Without the cluster runs there is no information indicating that the last byte of the file has to be recovered from cluster 12. Logically, the last byte of the files is recovered from cluster 8, which is incorrect. Figure 9, below, shows the sliver of aqua in cluster 8 that this recovery method would return, which is incorrect but unavoidable in these scenarios.

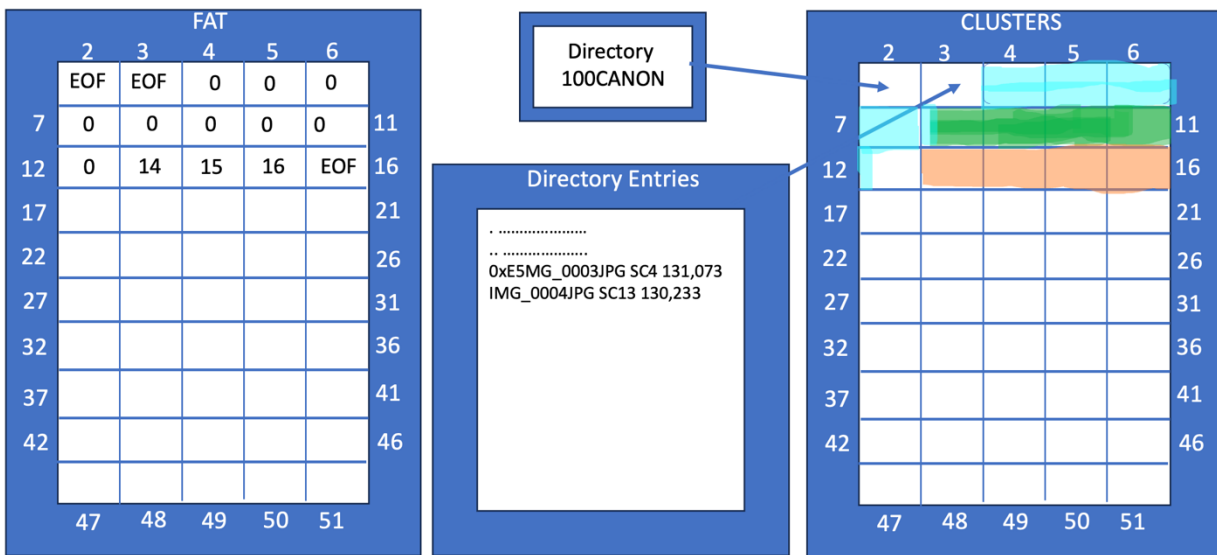
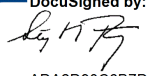


Figure 9 – Scenario Nine: Shows problems recovering deleted fragmented files

These scenarios and accompanying diagrams have been created after testing these scenarios repeatedly to test the rules and behaviors of these two operating systems (Canon 20D OS and Windows XP and 10). The results were consistent and repeatable.

The above findings and diagrams are true and correct to the best of my ability.

Stephen M. Bunting

DocuSigned by:

 ADA2D90C6B7D4FF...
 11/20/2023