A Method for Recovering Data From Failing Floppy Disk Drives

Dr. Frederick B. Cohen, Ph.D.
and Charles Preston

Background:

This paper is about a method for recovering data from floppy disks that are failing due to weak bits. It describes a repetitive read technique that has successfully recovered data from failing floppies in forensic cases and describes other related techniques. None of these techniques are new or particularly unique, however, they are not widely published to the best of the authors knowledge and some of the related analysis may be helpful in making more definitive determinations in some cases.

The nature of 'weak' bits and failure modes:

Floppy disks tend to degrade in various ways over time and under various environmental conditions such as temperature, humidity, and so forth. In some cases this results in the presence of so-called “weak” bits on the media. Weak bits are bits that are sufficiently degraded in their electromagnetic field so as to yield voltages between the values for a '1' and a '0' when read by the read heads on most floppy disks. This is a result of reduced flux density in the electromagnetic media. In particular, the floppy disk coding used in most current disks is identified in:

http://cma.zdnet.com/book/upgraderepair/ch14/ch14.htm

as Modified Frequency Modulation (MFM) which uses timed flux density transitions to indicate bits. In particular, it uses a “No transition, Transition” (NT) sequence to indicate a “1”, a TN to indicate a '0' preceded by a '0', and an NN to indicate a '0' preceded by a '1'. If a transition is not detected because of the loss of electromagnetic flux density, it can turn an NT into an NN or a TN into an NN, but it cannot turn an NN into either an NT or a TN. Pairs of bits always involve a transition. In particular, a '11' will produce NTNT, a '00' will produce either a TNTN (if there was another zero preceding it or a NNTN if there was a '1' preceding it, a '10' will always produce an NTNN, and a '01' will produce either a TNNT if it was preceded by a '0', or an NNNT if it was preceded by a '1'. If no transitions are detected, the controller will normally indicate an error condition and the CRC code will be irrelevant. So weak bits will either produce controller errors indicative of the inability to observe transitions at all, or weak transitions will result in the change of a T to an N. They cannot turn the lack of a transition into a transition. As a result, 7 out of 11 possible weaknesses turn into invalid codings that will be detected by the drive controller as invalid data. Of the remaining 4 errors that could produce valid data, 3 require that the previous bit be a 1 or they too will produce invalid data in the controller. All the possible changes are shown in the following table. In this table, the data values represented by *transition* and *no transition* sequences in flux density are enclosed in brackets (e.g., [11]) and required preceding bits are indicated prior to the bracketed pairs (e.g., 1[00]) where appropriate.

Data	***Originally***	***Can turn into***	***Result***
[11]	NTNT	NNNT	1[01]
	NTNT	NTNN	[10]
	NTNT	NNNN	invalid
0[00]	TNTN	NNTN	1[00]
	TNTN	TNNN	invalid
	TNTN	NNNN	invalid
1[00]	NNTN	NNNN	invalid
[10]	NTNN	NNNN	invalid
0[01]	TNNT	NNNT	1[01]
	TNNT	TNNN	invalid
1[01]	NNNT	NNNN	invalid

Note that none of these errors can produce a transition of the coded data from a '1' value to a '0' value. So no weak bit error can ever turn a '0' into a '1', it can only turn a '1' into a '0'. Additional consistency checks could potentially detect errors such as the transition of 0[00] into 1[00] because the previous '1' bit could not be the result of a weak bit (or its coding would be a '0' to '1' transition that weak bits cannot produce in its position). This then eliminates the otherwise possible error turning 0[01] into 1[01] and 0[00] into 1[00], leaving only the transition of [11] into 1[01] or [10] as results from reduced electromagnetic flux density in transitions. If the previous bit was not a 1 [NT] or the reduction in flux density reduced the T to an N, then the 1[01] error is also impossible.

This analysis is based on the assumption that a weakened field density in the locality of a bit cannot trigger a transition and that is worth discussing further. Normally, in order for a transition to be detected by a floppy disk controller, the electromagnetic field density in one region has to be positively charged while the adjacent region has to be negatively charged. Which is positive and which is negative combined with the direction of the movement of the disk in the drive dictates whether the drive head gets a positive or negative impulse, but these are not differentiated by the controller – both are considered transitions.

If a transition from the maximum field density to zero field density were to trigger a transition, floppy disks would be very unreliable because regions near tracks are commonly not used and any minor movement in the head could cause such a transition. In addition, the design of such devices is such that the positive and negative field densities are used to assure sound triggering. A half-level density change should not trigger a transition on most floppy disk drives.

For that reason, even a full field density area next to a zero field density area should not trigger a transition, and thus the weakening of electromagnetic strength on the disk should not create transitions where none existed. Of course the physical phenomena associated with weak bits are analog at this level of granularity. The size of a region of storage on a 720K floppy disk is on the order of 1/8000th of an inch circumference. Because of this relatively high density, most common physical phenomena is unlikely to reduce the field density of one region to near zero while retaining the density of the area right next to it at full strength. Perhaps a scratch could cause this to happen, but in the case of a scratch, the damage would be permanent and would likely produce the same level of transition on each use.

An electromagnetic field such as is produced by a magnet passing near the disk or a temperature condition, or even a biological phenomena is highly unlikely to produce such a dramatic edge condition. There is a strong tendency for these phenomena to produce regions with decreasing effects as a function of distance, and this produces a slow transition in field density resulting in a change in field strength with distance that will not normally produce a transition in the floppy disk controller.

As a result it is a sound assumption that no transitions will be created by reduction in electromagnetic field density associated with weak bits, and only the loss of transitions is likely to occur from these physical phenomena.

In addition to the MFM coding, floppy disks also use a cyclic redundancy check (CRC) code to encode a value at the end of each sector written. This is highly likely to be inconsistent when specific classes of errors occur to portions of the sector. Specifically, they are readily able to detect single bit flips, multiple bit flips in close proximity, and many other combinations of bit flips. According to:

http://www.ee.unb.ca/tervo/ee4253/crc.htm

> *"Any bit error term E(x) which is an exact multiple of P(x) will not be detected. This is the case, in particular, for the two bit error 10000001, where the two bad bits are 7-bits apart. Note that 10000001 =(1011)(1101)(11). The allowable separation between two bad bits is related to the choice of P(x). In general, bit errors and bursts up to N-bits long will be detected for a prime P(x) of order N. For arbitrary bit errors longer than N-bits, the odds are one in 2^N than a totally false bit pattern will nonetheless lead to a zero remainder. In essence, 100% detection is assured for all errors E(x) not an exact multiple of P(x). For a 16-bit CRC, this means:*
>
> *100% detection of single-bit errors;*
> *100% detection of all adjacent double-bit errors;*

100% detection of any errors spanning up to 16-bits;

100% detection of all two-bit errors not separated by exactly 2^{16}--1 bits (this means all two bit errors in practice!);

For arbitrary multiple errors spanning more than 16 bits, at worst 1 in 2^{16} failures, which is nonetheless over 99.995% detection rate.

Note that the impact of this coding on the available error modes from weak bits is such that the degradation mechanism would have to produce reduced flux densities exactly 32 transition distances from each other in order for the CRC code to fail to detect pairs of errors. Reductions in flux densities producing lost transitions in adjacent bits or other sequences of less than 32 transition areas (representing 16 bits of data) are 100% detected by CRC codes unless they range over large areas, in which case they would produce invalid codes in the MFM decoding mechanism in the controller. Thus, the physical phenomena identified with producing weak bits is highly unlikely to ever produce a condition in which a correct match between data from a sector and the CRC code match and no MFM coding error is produced and yet an alteration comes from the loss of a transition occurs undetected.

This implies that if weak bits are the cause of an error and a successful read of the data with matching CRC code is completed, it is highly likely that the data recovered accurately reflects the data last written to that sector. While we do not know how to produce a precise calculation of the resulting probability, it is certainly less than the probability of errors associated with either the MFM or CRC codes alone. That is, there is no synergistic effect that can cause one to correct an error produced by the other.

The recovery technique:

There are a number of options available for trying to regain the data once stored in sectors having weak bits, including using analog read techniques, disk head realignment, and attempting single and multiple bit changes in various ways to determine a set of bits consistent with the checksum stored at the end of each floppy disk sector. But these techniques are expensive and/or time consuming, require special skills and equipment, and might be challenged on the basis of their scientific validity. In some cases they may also involve damaging the original disk and certainly involve altered equipment that has to be validated in some manner prior to use.

The technique described here involves the use of multiple reads of single blocks in order to eventually get a valid combination of bits in the sector and its checksum. For specific actions performed as part of the case study effort identified here and used in an actual legal matter, a 3M brand write-locked double-sided 135 TPI light gray floppy disk was used, and is referred to heretofore as the Evidence Disk.

According to: Ritter, T. 1986. The Great CRC Mystery. Dr. Dobb's Journal of Software Tools. February. 11(2): 26-34,76-83.

> *The IBM 8-inch floppy disk specification used the CRC-CCITT polynomial for error-detection, and this CRC is now used in almost all disk controller devices. A disk controller computes a CRC as it writes a disk sector, and then it appends that CRC to the data. When the data is read back, a new CRC is computed from the recovered data and compared to the original CRC. If the CRC values differ, an error has occurred and the operation is repeated. The standard disk CRC (CRC-CCITT) is hidden in the controller, and nowadays receives little comment.*

We attempted to align floppy disk drives to different calibration settings in order to determine if we could use realignment of the disk drives to improve readability of content from the Evidence Disk. If the original writing disk drive is misaligned, there is a chance that better results can be achieved by trying different alignments. We went from one extreme of the range to the other and to several intermediate points between in order to see if improved results could be attained from different alignments but found that all alignments produced the same or additional errors. Based on this we concluded that it is likely that the reason for the errors is that the original disk has electromagnetic failures associated with “weakly” written or stored bits. Specifically, we believe that the mechanism of failure is that areas of the Evidence Disk associated with these errors have data that, when read by a typical floppy disk drive, produce voltage levels that are not clearly “one” or “zero”, but rather levels that lie between these extremes.

In order to attempt a recovery of the remaining data, we decided to try a technique that has worked in the past in similar cases based on repeated reads. In this method, each sector of the disk is read repeatedly until it produces a valid output. For the mechanism identified as the likely cause of the failure mode, repetitious reading with occasional disk head resets tends to produce occasional reads that have different values for the weak bits. When the combination of weak bit reads produces output that results in a consistent CRC code, the read for that sector succeeds, producing data that has a match between the CRC code and the data in the sector. The program then moves to the next sector and so forth until the whole disk is read with correct output data.

The specific program used in this case was executed from a bootable White Glove Linux CD which was kept with the evidence after processing to assure that the process could be precisely repeated if necessary. The shell script code executed in that environment is as follows:

```
for i in `count 0 1439`; do
        dd conv=noerror bs=512 count=1 skip=$i if=/dev/fd0 > noerr/$i.out
done
```

Within the White Glove environment, the “count” command used in this syntax counts from the first value (0) to the second value (1439) by increments of one. For each count value, the “dd” command is then executed with the “noerr” conversion option that specifies that on error, retries are to be attempted an unlimited number of times, with a block size of 512 (the normal block size for such a floppy disk), and a count of 1 block. This is done after skipping the count number of blocks from the beginning of the media, in this case the floppy disk “/dev/fd0”, with the output stored in a file named by the block number, in this case “noerr/[count].out” where [count] is the block number and noerr is the directory used to store all of the blocks. On each read attempt, a file is created, but unless the file is the result of a correct checksum, it is overwritten on the next attempt.

The reason it is beneficial to read a sector at a time is that a single error in a read produces a failure for the whole read. If a single sector takes 20 attempts on average to succeed, than on average, reading 2 sectors would take 400 attempts, and so forth. Since reading less than one whole sector does not produce different hardware execution, this approach minimizes the number of reads and reduces unnecessary wear and tear on the Evidence Disk while still reading repeatedly until a match between the CRC code and the data is attained.

In practice, weak bits tend to be fairly close to producing the voltage necessary to read them unless the sector is severely damaged, in which case the failures continue indefinitely. Because the voltage is close, occasionally the read head picks up a “one” even if most of the time the voltage is not quite high enough to get that indication. The weaker the bit (i.e., the less orientation alignment remains stored in the magnetic media), the more reads it takes, and if the bit is weak enough, the read will never succeed.

This process was applied to the Evidence Disk and produced different numbers of retry cycles on different sectors. On sectors that read without error consistently, there were no retry cycles. On the previously unreadable sectors, the number of retry cycles required ranged from one to more than 70 with many in the range of 20 to 30. Each sector was stored individually in a file of 512 bytes on a hard disk as it was read, and stored with a filename associated with the sector number as indicated above. For block number ranging from 0 to 1439, the total is 1440 blocks of 512 bytes each, or 737260 bytes of data, the entire readable contents of a 720K floppy disk.

The individual files representing the blocks of the Evidence Disk are then either independently examinable or may be assembled together into a single file representing the entire content of the original floppy disk and mounted using a loopback mounting interface or written to a fresh floppy disk which can then be used to read the data as if it were the original evidence disk. In the specific case used as an example here, the assembly was done using the following program in the same environment described earlier:

```
for i in `count 0 1439`; do dd seek=$i if=noerr/$i.out of=noerrdd.out ; done
```

In this case the blocks are written into the file at the appropriate locaiton in the same way as they were read from the evidence Disk in the first place. Multiple copies were made of the recovered disk for use by all sides in the matter at hand.

Based on the process described here, it seems highly likely that a successful extraction of the data from the Evidence Disk would yield an accurate depiction of the bit sequences that were on each sector of the Evidence Disk when each of those sectors was last written.

Comments on the method and the example used

The method used tends to support the contention that the disk failures were caused by weak bits. Specifically, if another mechanism was in effect, such as alignment errors or mechanical defects in the original writer, then the realignment process would have yielded better or worse data instead of nearly identical error behaviors. If bits were not written at all or if a typical contemporaneous weak bit writing mechanism were used, the levels would not likely vary across such a wide range of rereads. The fact that different numbers of rereads were needed at different locations on the disk tends to indicate that the failure mechanism produced errors distributed over a range of loss of electromagnetic field such as what is seen in overheating from poor storage, infection of the media with fungi or similar biological effects, or loss of data with time as is seen in many floppy disks, all of which take place over time as opposed to from instantaneous phenomena, These are precisely the sorts of errors that the CRC codes were designed to detect. No further examination of the media in this matter has been done at this time to identify the specific mechanism of failure.

There is a lingering question that is worthy of addressing yet again, that being the potential that the repeated reads could produce result after result that would eventually lead to a result that would produce a valid CRC code and no MFM errors, leading to false sector data accepted as legitimate. This particular scenario, because it involves weak bits, is somewhat less complicated to analyze than a scenario in which random changes are made because the changes associated with weak bits tend to be all in one direction, eliminating transitions and thus turning '1's into '0's. The likelihood of lost transitions causing detections is at least 17/22 for each transition based on the analysis of the table above. Because of the nature of the CRC code, errors that can go undetected also must be in quantities larger than 16 bits and distributed across the sector data area, or as combinations of the sector data area and the CRC area with probability no higher than 1 in 2^{16}. Since the CRC and MFM methods are not correlated in any way we can ascertain, a reasonable assumption is that the likelihood of both failing to detect a change from reduced electromagnetic density is no greater than 1 in $2^{16}*(5/22)^{16}$, or no greater than

152587890625 / 197349575662580634165903360

which is less than 1 in 10^{15}. The odds of coming across such an erroneous recovery is clearly low enough that for retries on the order of hundreds, there is almost no chance that false recovery could take place.

Summary and conclusions

It appears that this technique is effective in that it produces meaningful results in a reasonable amount of time. It appears to be accurate in retrieving data that is otherwise unreadable because of reduction in magnetic field to levels relatively close to original levels such as occur in natural disk degradation with time. Because this technique is based on normal floppy disks reads by standard unmodified equipment, it is less likely to be challenged and easier to implement than more complex and expensive techniques involving some sort of electromagnetic examination of the media or modified electronics.

This method also has some disadvantages in that the repeated reads can cause added wear and tear on the original evidence which may be fragile, it is likely to suffer from increased numbers of reads over time if the failure mechanisms are worsened by the repeated uses, and it does not reveal the specific mechanism of failure even if it produces reasonable results. It is also possible that large numbers of reads will not produce a valid result and that the process will have to be manually terminated and restarted at the following sector, leading to less complete recovery, and of course involving human intervention. If very similar numbers of reads are required for repeated attempts or across multiple locations on the disk, it appears to be potentially indicative of a copy protection or similar scheme, but this has not been tested or validated at this time though our efforts.

Finally, there is the highly unlikely possibility that the repeated reads could produce invalid data that happens to match the CRC code on the sector without creating invalid MFM codes in the controller, leading to false results. This increases as the number of reads increases but the number of reads required to create a serious potential for such an error appears to be far beyond any achievable number of such reads.